Learning by Doing: 
CISCO Certified Network Administrator 
(CCNA) 
Lab Manual version 5.0

Matthew J. Basham 
Basham.Matt@spcollege.edu 
Business Technologies Department, Clearwater Campus 
St. Petersburg College 
St. Petersburg, Florida
About the Author

Matthew Basham is the Program Director, Assistant Professor, Main Contact and Lead Instructor for the St. Petersburg College-CISCO Regional Networking Academy in Clearwater, Florida. He also oversees the daily operations of the Microsoft, Oracle, Linux, and programming degrees. Mr. Basham has been employed (adjunct and full-time) with SPC since 1996. He has completed his CCAI (CISCO instructor), CCNA 2.0, CCNP academy instructor and CCDA certification. He is currently working on a Ph.D. from the University of Florida in curriculum development (computer security), administration and leadership/educational entrepreneurialism, and IT policy development in higher education. He holds an M.A.M.C. from the University of Florida and a B.A. from Oakland University (Rochester, Michigan). Mr. Basham has been working professionally in networking as a network administrator, technical writer, teacher, and consultant since 1993. He has been using computers since the 1970's and still owns his original Commodore 64. He has held certifications or trained students in CCNA, CCDA, CCAI, Novell 3 and 4, NT 4.0 workstation, TCP/IP, NT 4.0 server, A+, Network+, COBOL, Fortran, Pascal, DOS, BASIC, and HTML. His hobbies include computer security, cyber-law, and anti-hacking techniques for network administrators and educators. He has lectured extensively across the United States over the past six years on curriculum developments in security and cyberlaw-related topics for higher education. He also has other publications and textbooks available through Lulu Press.

Looking for college courses? We deliver.

Choose from over 400 online courses in dozens of majors, including our new bachelor's degree program in Technology Management. St. Petersburg College's eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education.

Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

St. Petersburg College’s eCampus

http://e.spcollege.edu

Your time. Your place. Your future.
A note to students and teachers

This manual was developed to prepare students for hands-on training to accompany classroom lectures on CISCO networking theory for the CISCO CCNA examination. These labs are intended to supplement and enhance the Cisco Networking Academy Program with additional information, explanations, and laboratory materials, not to replace them. Think of this like a “cliff’s notes” to accompany the curriculum. If you are looking for a lot of theory, then you have got the wrong book.

This book uses a bunch of educational theory and the book should be used from the start to the finish. For example, in early labs IP addresses and related information is given. The later labs assume you have a good grasp of addressing and can choose your own. Yeah, I know it can stink but understanding subnetting is a critical part of an entry-level technician in networking and the “standard” Cisco curriculum does not address this very well. Also I put an emphasis on troubleshooting and critical thinking to a much deeper extent than the official curriculum. All labs and exercises contain four basic parts:

1. An objective section giving a brief topic for exploration in the lab or exercise
2. A tools and materials list
3. Steps needed to complete the lab or exercise
4. Supplemental lab or exercise challenge activities

Some sections include background information if needed.

Please keep in mind that equipment and IOS variations can cause differences between what is within this book and what you may actually see.

Some of the labs contain “guest router names” that are borrowed from computer security history. I do this to spice up the labs a bit and give you a cross-reference and some history of computer security. I even got an email from a past member of the LOD which was really cool. Keep them emails coming!

Keep track of updates and changes at http://www.spcollege.edu/star/cisco Scroll to the bottom of the page and click on the “Lab Manual Edits.”

Do you have a lab that you want in this book? Send it in and we’ll give you written credit for developing the lab if we pick it for use in our next book. We do, however, reserve the right to reformat the “look” of the lab to be similar to the style in this book and to do any minor edits. Before it appears in the newest version of the book you will still have the final say-so on any changes. After all this is about an open source effort and I feel giving the book away embodies the spirit of open source. The labs must be done using MS Word or compatible format. Sorry to all the Mac/OS users.
Acknowledgements

I would like to thank many people: my wife Michelle (for putting up with me and my “eccentricities,” long nights working on the book, and all the traveling I do), my kids Matthew and Madison, my family, grandparents (rest in peace), Worrell family, Wolfe family, Jeanette LaBelle-Wieske and family, Krysta, Rachel, Autumn, (no particular order), Beth & the Bindle family (you shouldn’t have said “no”), Ronda Tranter and family (legal inspiration), Frank Carlton Serafino Feranna (you’re right…we aren’t all dumb kids), J.C. and Mickey Converse, Uncle Bill, Aunt Mary, cousin David, Dave Ellis, Kent Plate and family, Rich Curtis, Flo Jacobsen, Ms. Minton (HS English Teacher), F*$Kmaster Mike, Melissandre Hilliker, Jaime and Jaime, Jen, Angela, Jessica, Julie Morrow and Lisa Wilson (for the ambition or revenge to do even better), and to all my students everywhere!

I guess I should also thank the makers of Mountain Dew© and Pizza Hut©, without them I would not have had my fuel for this…what ever happened to “Jolt Cola©” anyway?

I want to give a special thanks to all of my students, colleagues, friends, etc. who have pointed out the errors in this book (and sometimes the ones not in the book)...In the future this list will also include anyone else who finds an error and brings it to my attention: Rick Whelan, John Madison, Rich Curtis, Jessie Brown, Chad Olsen, John Collins, Scott Johnson, Brian Borowski, King Wong, JR Deng, John Ferillo, Michael Angel, Jack Fisher, and Francis Robitaille.

Special congratulations to my students who have gone on to pass the CCIE (written and /or practical portions)...those that I know of include Victor Floresca and Michael Brooks.
About the author 3
A note to students and teachers 4
Acknowledgements 5
Table of Contents 7

Part 1 Foundations of Cisco Networking 9

Part 1a: Cisco Foundations
- Searching CISCO for CCNA test information 11
- Registering for your CCNA exam 15
- An overview of Cisco routers and switches 17
- Paper Lab: Cisco Three-Layer Hierarchical Model 22
- Paper Lab: ICONS for computer diagrams 24

- DOS Lab 2K 85
- Windows 2000 Utilities Lab 99
- Dynamic DHCP Lab 101
- Changing TCP/IP Settings on Your Computer (Static DHCP) 105
- Intermediate DOS Lab: Troubleshooting Utilities 110
- FTP/TFTP Lab 117
- Telnet Lab 120
- Hyperterminal Lab 124
- Remote Access Lab 126
- Your Modem and You 135

Part 1b: Workstation Foundations: Knoppix STD
- Just what the heck is Knoppix STD anyway? 140
- DOS-like stuff in Knoppix Konsole 143
- Fun with Knoppix (FTP, Telnet) 154
- Communication Tools with Knoppix STD 162
- Sniffers in Knoppix STD (Using a protocol inspector) 166
- Utilities in Knoppix STD 171

Part 1c: Networking Foundations
- Installing a NIC: Hardware 179
- Paper Lab: Proper Cable for the Proper Job 180
- Peer-to-Peer Networking/File and Print Sharing 188
- Small Single Hub Networks 192
- Small Multiple Hub Networks 194
- Paper Lab: Binary Numbering 196
- Paper Lab: Hexadecimal Numbering 207
- Paper Lab: OSI Model and Encapsulation 211
- Protocols and the OSI Model 218
- Paper Lab: LAN Topologies 225
- Paper Lab: Broadcast and Collision Domains 230
Part 2 Switching
Switch Maintenance 287
Basic STP 292
Basic STP with one router 296
Intermediate STP 297
Basic VLAN 302
Basic VLAN with one router 304
Intermediate VLAN’s 308
Mixing it up: VLAN’s, STP, and RIP 309
Subnetting Example: ABC Packaging Company 311
Basic VTP 312
VLAN Trunking
Using a 2950 switch 315
Using a 4000/5000/6000 switch 323
ACL’s on switches
Part 3 Command Review 333
Whole Enchilada/Crazy Insano Lab #1 (WECIL): Switching 335
Whole Enchilada/Crazy Insano Lab #2 (WECIL): Switching 336
Whole Enchilada/Crazy Insano Lab #3 (WECIL): Switching 337
Whole Enchilada/Crazy Insano Lab #4 (WECIL): Switching 338
Whole Enchilada/Crazy Insano Lab #5 (WECIL): Switching 339

Part 3 LAN Routing Fundamentals
An Overview of CISCO Routers and Switches 155
Basic Router Commands 160
Router Boot Sequence 168
Basic Router Configuration 177
Routing Protocols: RIP 181
Basic Troubleshooting: Router-to-Router 188
Loopback Interfaces 197
Basic RIP with Protocol Inspector 203
Router Telnet Lab 207
Route Summarization with RIP 210
Intermediate RIP with 3 Routers 216
Using Syslog for network monitoring
RIP Metrics and the Limitations of RIP 219
Dynamic Host Configuration Protocol (DHCP) Lab 222
Subnetting with DHCP 225
Paper Lab: Variable Length Subnet Masking (VLSM) 227
Static and Dynamic Routes with Discontiguous RIP Networks 230
Overcoming Problems with Routing Loops 235
Routing Protocols: RIP Version 2 and Redistribution with RIP 237
Protocol Deathmatch! Rip versus Ripv2 283
Routing Protocols: IGRP 284
Packet Structure of IGRP with Protocol Inspectors 288
Intermediate IGRP: Metrics 292
Redistribution of RIP and IGRP 299
Routing Protocols: EIGRP 306
Routing Protocols: OSPF 312
Routing Protocols: BGP 317
Paper Lab: Routing Protocol Categories 322
Part 3 Command Review 240
Whole Enchilada/Crazy Insano Lab #1 (WECIL): Routing 242
Whole Enchilada/Crazy Insano Lab #2 (WECIL): Routing 244
Troubleshooting scenarios for part 2 245

Part 4 Access Control Lists 280
Paper Lab: Wildcard Masks 335
Paper Lab: Access Control Lists 337
Basic Access Control Lists 342
Extended Access Control Lists 347
Named Access Control Lists 351
Making a Protocol Inspector with ACL’s 354
Firewall Basics using Reflexive ACL’s 359
ACL’s on switches 362
Part 4 Command Review 362
Whole Enchilada/Crazy Insano Lab #1 (WECIL): IGRP/RIP 364
Whole Enchilada/Crazy Insano Lab #2 (WECIL): IP/IPX 365

Part 5 Wide Area Network Fundamentals 366
Registering for your CCNA exam 367

Auxiliary Port Configuration 368
Remote Access Lab: Using a Modem with a Router (AUX)
Using a T-1/DSL line with a Router

Serial Line Configuration 371
Point-to-Point Protocol 371
PPP with Authentication 376
Remote Access DUN with PPP Encapsulation 387
Setting up a Router to be a Frame Relay Switch 391
Frame Relay with 2 Routers 393
Frame Relay: Hub and Spoke with 3 routers 396
Fully-Meshed Frame Relay with 3 Routers and Sub-interfaces 399
Frame Relay Operation and Troubleshooting
WIC’s and Network Module Plug-in Configuration
  Basic ISDN Configuration with BRI interface (MERGE)
  Basic ISDN Configuration with BRI interface (ADTRAN)
  ISDN Operation and Troubleshooting
  ISDN Configuration with Multiple Routers (ADTRAN)
  Frame Relay with ISDN Backup
Homestretch
  Part 5 Command Review
  Whole Enchilada/Crazy Insano Lab #1 (WECIL)
Part 1: Foundations of Cisco Networking

Here in this section I break it up into three big chunks. In the first part I wanted to give you a good overview of Cisco, Cisco certifications, testing and searching for stuff on Cisco’s website. The next part I cover some foundational information about workstations that are particularly relevant to our labs here. I left the ones with Windows 98 because I figured there still would be some schools out there somewhere that may need them. I also did some of the stuff for Windows 2000 that should also be pretty close for ME and XP. Now here is the real deal: I put some labs in for Knoppix STD, a Linux-like free operating system (Security Tools Distribution). Long live open source! Do you want Cisco’s operating system? I heard you could find it in China somewhere! Just kidding. The last section covers a whole bunch of networking topics that should bring you up to speed for the Cisco labs. It would really do you some good if you want to make a living doing this stuff to go out and take a couple of PC repair classes, a couple of Microsoft Networking classes, and a couple of Linux classes along with the CCNA.

a. Cisco Foundations
b. Workstation Foundations
c. Networking Foundations
They said “I was crazy!”

How do you like the idea of free and low-cost textbooks? Well here is how you can help me in the effort to bring more of these types of books to you. They say there is safety in numbers, well I can promote this type of textbook production way better if I hear from you personally. Send me an email at Basham.Matt@spcollege.edu just to tell me you are using the book, you like the idea of cheaper high-quality books, or to just say “hello.” Together we will change education for the better!
Searching CISCO for CCNA Test information

Objective:
To learn how to find out the latest CCNA test information from the CISCO website.

Step-By-Step Instructions:
1. Open a browser window.
2. Navigate to www.cisco.com. You should see something like this (remember web pages are frequently updated so you may have to “wing it” a bit…never rely on the web to stay the same):

Feel free to take some time and just enjoy the scenery. There are actually some freebies you can sign up for like Packet magazine and some white papers. You just got to love the free stuff. What’s that? You are a bit confused…don’t worry we’ll hit all the important stuff as it pertains to this book.

Hungry for more???

You are almost there for the CompTIA Network+ and Security+ Certifications too! For more information:

http://www.comptia.org/certification/default.aspx

Building successful careers for students everywhere…
3. Next, on the left hand side you should see a link under the “Learning and Events” link. After clicking on it then you should see:

4. Then (as shown in the above picture) click on the link for “exam information.” The page you should see next is:
5. Click on the link for “Certification Exams.” It will take you to the page for current exams and outlines (isn’t that nice?). You should see:

![Certification Exams Page](image1)

6. Click on the link for the current CCNA exam (probably the one at the top) when this book went to print it was “640-801” and another window should open. You should see:

![CCNA 640-801 Exam Page](image2)
7. Again, scroll down a bit and you should see some available options (hyperlinks). Let’s “dissect” the page a bit…some helpful links and information:

8. The “Preview Course Simulation Lab” link will open another page. To learn more about the simulation tool, use the graphic tutorial links. You may want to spend some time going through the instructions. Figure out if short-cut keystrokes are allowed or not. Your actual CCNA exam may contain some of these simulations.

9. Also look at the description of exam topics. Yeah, I know…they stink. It is kind of getting a recipe with no name and just some of the ingredients without any sort of instructions or amounts to use. Just make sure you feel comfortable with the subjects. The typical Cisco test over parts 1 through 3 will also require you to know parts 4, 5, and 6. Take that sentence for what you want. Use this to guide your studies as you progress through your CCNA training. Not every one of those topics is covered here in this book because this book was not designed to replace the Cisco curriculum, but to be used to enhance and supplement it.

So what have I learned here?
In this lab you have learned how to find the CCNA test objectives. Consider this sort of a “table of contents” for your studies, even though CISCO is extremely vague with their test information. It really doesn’t help all that much. Remember that people are always updating their websites so you may have to do a little winging it. In any event, even though you are not ready for the CCNA test, you should keep those objectives in mind while studying and you should start spending more time at the Cisco website. Later, during your employment as a Cisco technician the more skilled you are at navigating their website, the more successful you should be as a technician.
Registering for Your CCNA Exam

Objective:
To learn how to register for the current CCNA test.

Question and Answers about the CCNA:

Where can I register? With any prometrics center. You can also call 1-800-204-EXAM for more information. Or, you can also go to a VUE testing center.

How much does it cost? $125 per attempt for each test. (Don’t flame me if it changes...blame it on printed stuff)

What is a passing score? For CCNA 849 of 1000 is a passing score. There are about 45-55 questions to complete in 75 minutes. At least on the newer test questions are weighted. Some of those pick three of six questions give you partial credit for being close.

What is it like? The new test has simulations and drag and drop questions. It is Cisco’s attempt at a practical exam for CCNA. Supposedly if you cannot work on the equipment then you should not be able to pass the test. This works well for you because you are “learning by doing.” The rest of the test is mostly multiple-choice questions. Some are command line entries, matching, and fill in the blanks. There are four sections: Planning and design, Implementation and operation, Troubleshooting, and Technology. I had heard from some of my students there are four or five troubleshooting simulations and a bunch of stuff on access control lists, frame relay, and subnetting. Believe it or not, even though OSPF is predominantly a CCNP-level topic, you need to know it very well for the CCNA. Get used to it...for anything in Cisco if you want to pass #4 you must first know 5 and 6. I know it makes absolutely no sense but what else should you think about from such a large conglomeration? Also, unlike other tests you are NOT allowed to mark a question to return to later. You get one look at a question. You will be given a computer workstation, a dry wipe marker, and a two-sided laminated card for notes AND NOTHING ELSE! You are not allowed any food, drinks, notes, NO CALCULATORS, etc. You will need two picture ID's.

I want to put a personal shout out in here to my buddies over at Gambitt Communications. They truly have some snappy equipment simulators. If you or your school cannot afford stacks of racks in equipment then the stuff from Gambitt is the next best thing to being there. Of course I prefer hands on when ever possible...the real world has tons of physical-layer (cabling) problems that you just can’t “simulate.” Also my other concern with simulators is this: THEY WORK PERFECTLY. You almost never see that with hands-on equipment. We’ve got new Cisco routers with ISDN cards that seem to work, only when they feel like it. I like the hands on for discovering the differences between the IOS operating systems too. Don’t let me dissuade you from simulators...in fact, even
if you have stacks of racks I would still buy a simulator just to practice for the simulator questions on the test. So let me put a plug in here for my buds:

**MIMIC® Virtual Lab CCNA**

MIMIC® Virtual Lab creates a real world lab environment with a network of Cisco Routers and switches. It gives you hands-on experience on the devices without buying expensive equipment. It provides a safe environment for practice. No need to worry about bringing down the equipment or network and affecting other users.

**What if I fail?** Study a bit more, practice some more on the equipment and re-take it soon. If you miss by only one or two questions, then most people re-take the exam right then and there and usually pass. Don't feel bad. Most people need a time or two through the first one.

**When should I take it?** You should take it as soon as you finish Semester 4 while the information is still fresh in your mind. Don't wait too long. I had a bunch of students who took the tests at different times and we generally found that taking it on Wednesday morning tended to have the easiest pool of questions. I am really not sure why that seemed to be except that maybe they think people who cram all weekend take tests on Mondays and those who cram all week take tests on Fridays. Probably by the time this comes out it will change because we are on to their little secret. Anyways there is supposedly a pool of about 3,500 questions that are drawn from for the test and your test “locks” a portion of that database. They wouldn’t dare do an adaptive test. That's been tried before and failed. The way those tests worked is each question needed to be answered in so many seconds…get it right and the computer assumed you knew that topic and it moved on to another one. But, get it wrong or take too long and get it right and it may have stumbled upon an area you did not know very well. So, it kept asking you questions about that topic until you barely passed or barely failed the test. Smarty-pants like me would find a question we absolutely knew front and back and just take 5 minutes to answer the question. Then we had effectively rigged the test for questions we knew very well. Neat huh? The best thing I can suggest for practicing is to purchase a Cisco test simulator. Yeah, sure I tried the ones
from Boson, Transcender and the other companies but, strangely enough, the Cisco ones was closest to the “real thing.” Just be careful not to over-think any questions on the test. There is a big difference between what is in the textbooks and what you can do in the real world. If the book says you cannot use the first and last subnet (even though I know we can) then I would mimic that answer on the test. Thankfully, Cisco now will tell you if they are assuming the ip-subnet zero command is enable or not. This command will allow you to use the first and last subnet, but you will learn more about that later.

Keep track of updates and changes at http://www.spcollege.edu/star/cisco. Scroll to the bottom of the page and click on the “Lab Manual Edits.”

A place I love to go…Network Computing Magazine…every month they have a featured “centerfold” showing the layout of a diagram. The part that stinks is because it is called a “centerfold” many school firewalls or IIS programs will not allow those pages to be displayed. Hopefully you will have a progressive network administrator that will be agreeable and allowing this site through:

http://www.nwc.com/departments/ctr.jhtml
An Overview of CISCO Routers and Switches

Objectives:
To become familiar with CISCO networking categories which, in turn, will enable you to more easily find technical information about networking devices on the CISCO website: http://www.cisco.com.

Background:
During the course of your studies you may encounter many different models of CISCO routers and switches. This lab is designed to give you a general overview of how CISCO routers and switches fit into their “3-layer hierarchical model” which, will allow you to more easily find technical information about specific models. This lab will also give you an overview of some of the features of the 2500 and 2600 routers and 1900 and 2900 switches that you may encounter during your CCNA studies.

3-layer Hierarchical model
As you may recall from CISCO textbooks, CISCO strongly suggests using a 3-layer styled model for designing networks. The “core” of any network design should be implemented for high-speed switching. This layer just wants to move the information around as quickly as possible. The distribution layer helps to re-distribute those fast moving information packets, but may be slowed down by some decision-making from a router. Finally the access layer is where users connect to the network. This is considered to be the “slowest” layer because of the extensive decision-making that may be taking place here.

![3-layer Hierarchical model diagram]

The core layer (high-speed switching) is where you would find the most redundancy between devices. The distribution layer is where you would find network policy implementations, some security, and routing between VLAN’s. The access layer is where you would find your users connected to the network, workgroups, servers, and some security. As you progress through your studies you will learn more about the functions of each layer and how they play an important role in network design.
More importantly to you right now if you wanted to find information about a CISCO 2500 router at CISCO’s website you would almost need a miracle to find it unless you knew a 2500 router is classified as an “Access” router. Now, you could go to the CISCO website, access the technical document section, then select the “access” or “modular access” routers heading, and then select 2500’s to get your information. This is much easier. I guess the old phrase “easy when you know how” really fits here. Table 1 shows a general overview of the CISCO routers and switches and which layer they are typically attributed.

<table>
<thead>
<tr>
<th>CORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6500 switches</td>
</tr>
<tr>
<td>8500 switches</td>
</tr>
<tr>
<td>7000 routers</td>
</tr>
<tr>
<td>10000 routers</td>
</tr>
<tr>
<td>12000 routers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DISTRIBUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>4000 switches</td>
</tr>
<tr>
<td>5000 switches</td>
</tr>
<tr>
<td>6000 switches</td>
</tr>
<tr>
<td>3600 routers</td>
</tr>
<tr>
<td>4000 routers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>700 routers</td>
</tr>
<tr>
<td>800 routers</td>
</tr>
<tr>
<td>1700 routers</td>
</tr>
<tr>
<td>2500 routers</td>
</tr>
<tr>
<td>2600 routers</td>
</tr>
<tr>
<td>1900 switches</td>
</tr>
<tr>
<td>2820 switches</td>
</tr>
<tr>
<td>2900 switches</td>
</tr>
<tr>
<td>3500 switches</td>
</tr>
</tbody>
</table>

Table 1—CISCO routers and switches as they correlate to the 3-layer hierarchical design model.

The 2500 router seems to be the staple of many CCNA Academies worldwide. Too bad for them, because CISCO has recently declared these products to be “End of Life” and will not be supporting them, or doing software upgrades on them very shortly. There certainly will be a lot of schools scrambling to find money to replace them. Let’s look at what some people call the “front” of a 2500 router in figures 1, 2, and 3. The 2500’s are, for the most part, “fixed” units. There is very little we can do to change them. If we need three Ethernet ports, then we will have to add another router. At best we can have two Ethernet ports (using transceivers on the AUI ports).
Figure 1—CISCO 2501 router “front” view.

Nothing fancy here…personally I consider this to be the “rear” of the router since I do all of my work on the other side. So let’s take a look at the CISCO-termed “rear” of the 2500 router.

Figure 2—CISCO 2501 router “rear” view, dual serial, single AUI.

Figure 3—CISCO 2514 router “rear” view, dual serial, dual AUI.

The 2600’s, on the other hand, are more “modular” in style. From figures 4 and 5 we can see some removable plates/covers. This is where a variety of modules can be inserted. The two smaller plates can have WAN Interface Cards (WIC’s) inserted. These are things like dual serial interfaces, ISDN modules and T-1 modules. The larger removable plate/cover is for, well, larger modules with many Ethernet, serial interfaces or even multiple ISDN interfaces. We are talking up to 24 or so lines. A far cry from those 2500’s huh? Different routers can use different modules so check your documentation carefully.
Figure 4—CISCO 2610 router “rear” view, single Ethernet, no serial.

Figure 5—CISCO 2611 router “rear” view, dual Ethernet, no serial.

Figure 6—CISCO 1924 switch “front” view, 24-port switch (10Base T ports with 2 uplinks).

Figure 7—CISCO 1924 switch “rear” view, 24-port switch (10Base T ports with 2 uplinks)—same on 2924.
Figure 8—CISCO 2924 switch “front” view, 24-port switch (100 Base T ports—all ports capable of being uplinks).

Figures 6 and 7 show the switches common to most students in these labs. These switches have 24-10BaseT ports and two ports at 100BaseT that serve as uplink/downlink ports. Heck, they are even called ports “26” and “27.” Now there is a task…try to figure out where port “25” is located! In figure 8 we see the 2924 switch common to CCNP labs. The only difference between the two is every port is 100BaseT and capable up uplink/downlink. That is why no “extra” ports 26 and 27 are out to the right side.

Supplemental Lab or Challenge Activity:
Go to www.cisco.com and look up:

1. Release Notes for CISCO 2500 Series Routers
2. Hardware Installation Notes for 2600 Series Routers
4. Catalyst 2900 User Guide

Print out the first page of each as evidence of completion for your instructor.

So What Have I Learned Here?
In this lab you have been introduced to the CISCO hierarchical model. We won’t be doing too much with this here in the CCNA course but if you want to learn about the design stuff (CCDA) plan on seeing it in your sleep. We also have a lab on it again in Part 3. This is a nifty overview of the routers and switches that you may encounter during your CCNA studies.
Why do we need to do this? Simple, it will help with navigating Cisco’s website. We don’t go out looking for a 2620 router help; we first look for access routers then pick the 2620 from there. Crazy, I know, I know.

Match the function with the layer.
1. Provides workgroup and user access to the network. core
2. Provides policy-based connectivity. distribution
3. Provides optimal transport between sites. access

For the following please answer (1) for core-layer function, (2) for distribution-layer function, or (3) for access-layer function.

1. _____ Usually a LAN or group of LAN’s.
2. _____ Gives network services to multiple LAN’s within a WAN.
3. _____ Provides users with network access.
4. _____ Provides fast wide-area connections between geographically remote sites.
5. _____ Where ACL’s are found.
6. _____ Where security policies are implemented.
7. _____ Used to tie together a number of campus networks in a WAN.
8. _____ Where servers are connected.
9. _____ Where the campus backbone is found.
10. _____ Usually point-to-point links.
11. _____ Broadcast/multicast domain definition.
12. _____ Where filters are found.
13. _____ T1/T3 lines are usually used here.
14. _____ Where servers that will be access by different workgroups would be placed.
15. _____ Used to connect together buildings on a single campus.
16. _____ Shared bandwidth.
17. _____ Provides boundary definition.
18. _____ Frame Relay lines are usually used here.
19. _____ Fast Ethernet is usually used here.
20. _____ Switched bandwidth.
21. _____ SMDS lines are usually used here.
22. _____ Provides a fast path between remote sites.
23. _____ MAC-layer filtering.
24. _____ Departmental or workgroup access to the next layer.
25. _____ Load Sharing, redundancy, and rapid convergence are essential.
26. _____ Microsegmentation.
27. _____ The layer where packet manipulation occurs.
28. _____ Address or area aggregation.
29. _____ Connects LAN’s into WAN’s.
30. _____ Efficient use of bandwidth is a key concern here.
31. _____ VLAN routing.
32. _____ Where any media transitions occur.
33. _____ Isolation of broadcast traffic.

Match the CISCO networking device with its associated layer. Use a (1) for core-layer device, (2) for a distribution-layer device, or a (3) for an access-layer device.

<table>
<thead>
<tr>
<th>Routers:</th>
<th>Layer:</th>
<th>Features:</th>
</tr>
</thead>
<tbody>
<tr>
<td>700</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
<tr>
<td>800</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
<tr>
<td>1600</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
<tr>
<td>1720</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
<tr>
<td>2500</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
<tr>
<td>2600</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
<tr>
<td>3600</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
<tr>
<td>4000</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
<tr>
<td>7000</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Switches:</th>
<th>Layer:</th>
<th>Features:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1548</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
<tr>
<td>1900</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
<tr>
<td>2900</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
<tr>
<td>4000</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
<tr>
<td>5000</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
<tr>
<td>6000</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
<tr>
<td>8000</td>
<td>_____</td>
<td>_____________________________</td>
</tr>
</tbody>
</table>

There are some rumblings and grumblings about a fourth layer called “the edge” but I really don’t see much difference at the CCNA-level. Just know it exists and it will be changing this a bit in a later version.

---

Take college courses on a new schedule--yours.

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education. Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

[http://e.spcollege.edu](http://e.spcollege.edu)

St. Petersburg College’s eCampus. Your time. Your place. Your future.
Paper Lab: ICONS for Computer Diagrams

Objective:
To learn about ICONS used in CISCO drawings and for what each represents.

Tools and Materials:
None.

Step-By-Step Instructions:
Let’s just go through all of them one by one:

![Router Icon]
Router—Layer 3 device. Models include 2500 and 2600 series for access layer.

![Communication Server Icon]
Communication Server—This provide access to networking devices over a LAN or WAN using Serial Line Internet Protocol (SLIP). You won’t probably use this too much since other technologies are getting cheaper and easier to use.

![Gateway Icon]
Gateway—Device that acts as a “gateway” to the network or Internet.

![Bridge Icon]
Bridge—Old school layer 2 device not used too much anymore.

![Workgroup switch Icon]
Workgroup switch—Layer 2 device that you will use plenty. A CCIE-guy told me “one good future in networking is in switching” (the other is in security).

![100BaseT Hub Icon]
100BaseT hub—Not used too much anymore since switches cost about the same.
10BaseT Hub—Not used too much anymore since switches cost about the same.

CISCO CAT5000/5500—Older switching technology that uses “set” based commands. Newer 4000’s and 6500’s replace these.

Router switch processor (RSP)—The brain of a switch router that handles routing functions on a switch.

Putting those two together…CISCO Big-Cat’s 4000/5000 with route switch processors (RSP).

ATM switch—Not hard…a switch for ATM networks.

ISDN switch—ditto for ISDN networks.

TAG router switch—uses TAG’s to forward packets. Does routing functions too.

Broadband router—Router for broadband connections.
CISCO Net Ranger—CISCO security device. Think of it as a spy on your computer.

ATM Router—Router for ATM. 8500 series routers.

CISCO 7505 Router—distribution/core layer router.

CISCO 7507 Router—distribution/core layer router.

CISCO 7500 (7513) Router—distribution/core layer router.

ATM TAG switch/router—higher level switch routing. Typically 7000 series related.

MAIN Frame—oh…that’s the old school stuff.

IBM A/S 400—ditto, although these are still found in accounting departments.
CSU/DSU—Channel Service Unit/Data Service Unit…from the “WAN cloud” into this and then into your router. A TSU is a CSU/DSU for a T-line.

PIX Firewall—Security device. Only works with IP. All other protocols must be tunneled through it…so what’s the point of having it?

Small PBX—mini telephone company service that goes in your company. If you dial a “9” to get an outside line, then you have a PBX-type system.

The “Cloud”—This is where all WAN starts and ends. We use this in many instances…to represent the Internet, a frame relay cloud, an ISDN cloud, a POTS cloud, etc.

PC/Workstation—I really should not have to explain this one.

Dumb terminal—Like a regular PC, but no hard disk. It was mainly used to connect to mainframe who did the storage and processing for them. Yeah, they are still used. One of the newspapers here in town uses them with a mainframe.

Printer—I really should not have to explain this one either. So there.
Laptop—ditto.

File server—Used on networks to hold files and share processing requests from workstations. Some here, some on the PC. It’s called client-server networking.

Supercomputer—See Nasa, Berkely, MIT, etc. Kind of like the W.O.P.R. in Wargames.

Web cluster—A special cloud indicating several web devices are contained within the cloud.

Web server—Holds the Internet pages of a company. Microsoft IIS and Apache are common software packages on these.

Repeater—Layer 1 device that performs no intelligent processing, only cleaning up, amplifying, and re-timing the signals. Not used too much anymore.

Token Ring—ICON to represent a layer 2 token ring topology. Not used too much anymore.
FDDI—Icon to represent a layer 2 FDDI topology. Similar to token ring stuff.

Ethernet—Icon to represent a layer 1 or 2 Ethernet cable.

Serial—Icon to represent a layer 1 or 2 cable. V.35 and V.24 are common examples.

Circuit Switched Serial—ditto.

Modem—Modulator/Demodulator. Translates analog into digital signals.

Phone—I should not have to explain this.

PC Camera—Itty bitty camera for your computer.

PolyComm phone—Speaker phone commonly used for conference calls.
Firewall—Network Address Translation device. Great when they work properly. There is a big future in computer security…especially if you can get these things to work right. A Cisco PIX firewall is an example…the symbol for a PIX firewall and this little brick wall are sometimes used interchangeably.

Router with firewall—Just what it sounds like…a router with the addition of firewall commands.

Satellite—If you have the bucks you can set up a network with this…sometimes you have no choice…think about a cruise ship company and how they communicate.

Satellite dish—used with satellites.

CISCO Call manager—Works with Voice over IP equipment. Starting to be a “hot” item for resumes and career development.

IP telephone—yes you really can read your email over this phone…gets its own IP address and everything.

You will see some of these used in the drawings in this book. I put the other ones in here because I see them being used in articles and books about networking. You can also download this file and copy and paste the icons too.

More Icons on the web! (amazingly they didn’t change since the first print!)

So what have I learned here?
You have been given a brief introduction to icons used in network drawings. Let’s test your knowledge here. *Without looking back* at the pages can you identify what these icons represent?

So what have I learned here?
You have been given a brief introduction to icons used in network drawings. Let’s test your knowledge here. *Without looking back* at the pages can you identify what these icons represent?
They said “I was crazy!”

How do you like the idea of free and low-cost textbooks? Well here is how you can help me in the effort to bring more of these types of books to you. They say there is safety in numbers, well I can promote this type of textbook production way better if I hear from you personally. Send me an email at Basham.Matt@spcollege.edu just to tell me you are using the book, you like the idea of cheaper high-quality books, or to just say “hello.” Together we will change education for the better!
Objective:
This lab is designed to become familiar with basic DOS commands and utilities on Windows Operating Systems version 2000.

Tools and Materials:
(1) Computer with Windows 2000
paper and pencil

Background:
In this lab you will learn about DOS…no, DOS is not dead! Being able to master simple DOS commands and utilities will enhance your networking skills considerably, especially in troubleshooting network problems. You may even wish to purchase a DOS tutorial at some point in your networking career. Many operating systems (windows-based too) use DOS commands for updates, patches, and maintenance. I know the Novell system frequently makes use of changing file attributes before applying new patches to the operating system. These are done with DOS-like commands. UNIX/LINUX is heavily DOS-command style oriented. If you want to get into computer security then you will have to live, eat, and breath DOS and UNIX/LINUX (or as you will find I like to use Knoppix).

Step-By-Step Instructions:
1. Opening DOS. Open the MS-DOS prompt into a full-window. If you are not sure, then follow these steps.
   a. Click on the “start” button on your task bar.
   b. Click on “programs.”
   c. Search for and click on MS-DOS prompt A black screen or a window with a black screen should appear.
d. Or, if you want to be a show-off then click on “Start” then “Run.”

```
e. Type in “cmd” (without quote marks) and the black screen DOS window should appear. (right picture above)  
f. If you really have some time to kill then go to “Start” then “Programs” then (but don’t click on it) “MS-DOS Prompt.” Once you are there right-click on it and select properties. (left picture below)  

g. Ok…now you can really start showing off…click on the “options” tab. (right above)
```
Here you can change which shortcut keys are allowed, sensitivity, etc. There are some neat settings under the screen tab also. Lots of things to play with and lots of things to do with DOS. Try changing background colors, fonts, etc. Aha! Your first script kiddie assignment…hearing the “Oooo’s” and “Aaaaahh’s” when your DOS prompt comes up with different colors. Yeah, it only takes a little to impress.

2. **DOS prompt and directory file structure.** The DOS prompt and DOS system can be thought of similar to a filing cabinet. If you have three drives (C, D, and E) then each one can be thought of as separate filing cabinets C, D, and E. Each of those cabinets are then called the “root” directory of each cabinet. Each root directory can contain many different “directories.” These directories can be thought of as drawers in the cabinets. From there each directory can contain many different “sub-directories” similar to folders. Each “sub-directory” can contain other subdirectories and so on…at any point (root, directory, sub-directory, etc) can contain computer files (thought of similar to documents…they can be placed in a folder, drawer, etc). So lets take a peak and put this all into perspective…

```
C:\ Root prompt
C:\Windows directory called “windows” of root “C”
C:\Windows\System sub-directory called “system” in directory “windows” of root “C”
```

Let’s look at an example of navigation with Windows 2000 DOS. Using the directory “tree” structure shown on the next page we could write down the paths for certain files. For example the complete path to the album.zip file would become:

```
C:\ Documents and Settings\Basham.Matt.admin\MY_Documents\My_Pictures\album.zip
```
See if you can give the complete path for the following files (This is not what your computer will look like...just a make-believe one for this exercise):

<table>
<thead>
<tr>
<th>File Name</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>lulu.url</td>
<td>C:\Documents and Settings\Basham.Matt.admin\2600.url</td>
</tr>
<tr>
<td>letter.doc</td>
<td>C:\Documents and Settings\Basham.Matt.admin\My Documents\My Pictures\album.zip</td>
</tr>
<tr>
<td>disk cleanup.lnk</td>
<td>C:\Documents and Settings\Basham.Matt.admin\My Documents\My Files\addresses.doc</td>
</tr>
<tr>
<td>Favorites</td>
<td>C:\Documents and Settings\Basham.Matt.admin\My Documents\My Webs</td>
</tr>
<tr>
<td>Accessories</td>
<td>C:\Documents and Settings\Basham.Matt.admin\Start Menu\Programs\Accessories Communications Hyperterminal</td>
</tr>
</tbody>
</table>

Make a map of the structure of the C: drive on your computer. Be sure to include all sub-directories and folders if you have time. A quick way to do this is to type `tree` and the folder name at the DOS prompt and you will see “the tree” for that folder. If you just type tree you will see everything from the C drive on down. The problem is you will not be able to scroll up and see everything.

**Navigation.** The next thing to learn is navigating and finding files in DOS. We have several commands and techniques for doing this. Sometimes this is called navigating the “tree” or walking up and down the tree. The first command you will learn allows you to change directories. You do this by typing “CD” at any prompt and the root/directory/subdirectory you wish to change to. For example, when we first open our DOS window we see the prompt: “C:\Documents and Settings\Basham.Matt.admin\>” If we wanted to navigate to the “My Documents” file directory (C:\Documents and Settings \Basham.Matt.admin\windows\my documents) we could switch to it in one of several ways... (1) type “CD C:\Documents and Settings\Basham.Matt.admin\mydocuments” or (2) type “CD My Documents” (capitalization is not important...this is also known as case sensistivity) this will
change you from the directory “C:\Documents and Settings \Basham.Matt.admin\” prompt to the “C: Documents and Settings\Basham.Matt.admin\ My Documents” prompt. Please note that you can use the dot-dot to go back one level with the CD command. To get back to the C:\ Documents and Settings \Basham.Matt.admin\ prompt just type “CD..”.

So using the tree above as a guide what would you type at the following prompts (don’t actually do it…your computer file structure will be way different)?

From c:\ Documents and Settings\Basham.Matt.admin\ to get to the root prompt

__________________________________________________________________

From letter.doc back up two levels  ____________________________________

Finding Files in DOS.  Sometimes we do not always know or cannot remember the exact file name.  For those times we can use a wildcard character.  Say for example we knew it was an autoexec file but couldn’t remember the extension.  We can just do a directory for all files named autoexec by typing “dir autoexec.*”  The asterisk will replace any one or any number of characters as in “dir *utoexec.*”  If files named buttoexec.com, cutoexec.zip, and futoexec.wiz existed on the directory being searched, then they all would be listed.  As Emeril says, “let’s kick it up a notch!”  If we wanted to see all files in a directory then we would type “dir *.*” but, be careful, too many files might whiz by…in that case we could append /p to the end of the command to only list one page at a time…then we would have to hit any key to see the next page(s) one at a time “dir *.* /p”  Getting tired of too many pages?  Just press control+C to cancel the action.  You can get a “widescreen” view using the /w option…“dir *.* /w” or combine them: “dir *.* /w /p” or, in Windows 2000 you can simply just scroll up or down.

3. Getting help.  To find out any subcommand or options available with a command just append /? to the command.  For example, if we wanted to find out the subcommands available with ping type “ping /?” and read away!

What do these commands do?  (Hint: some will not have anything listed for help)

Internal commands: Built into the operating system file (command.com) and loaded into memory whenever your computer is turned on.

break  ______________________________________________________
call  ______________________________________________________
clp  ______________________________________________________
chcp  ______________________________________________________
cls  ______________________________________________________
copy  ______________________________________________________
ctty  ______________________________________________________
date  ______________________________________________________
del  ______________________________________________________
echo  ______________________________________________________
41

4. Make some files. Open up your notepad and create some files in the c:\temp folder:

<table>
<thead>
<tr>
<th>File name</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dave.txt</td>
<td>This is Dave’s text file…so keep out!</td>
</tr>
</tbody>
</table>
Matt.txt This is Matt’s text file…so keep out!

Scott.txt This is Scott’s text file…so keep out!

Tim.txt This is Tim’s text file…so keep out!

5. **RENAME.** One of those tools you might require when loading patches or something is the ability to rename a file. It’s usually a good idea to make a back up of a file before doing something drastically with it. For example if we had an executable called matt.exe that we were going to upgrade we should copy it to another directory and make a backup of it first.

```
Copy c:\windows\matt.exe c:\temp
Ren c:\temp\matt.exe c:\temp\matt.bak
```

On the second line we see our rename command. First we indicate the rename, the file to be renamed, and then what the new file name will be.

6. **DOS utilities.** Let’s find out about some really neat dos utilities on your computer. Try each file and getting help for each file. These are some from the same sub-directory as my command.com file. Most of these can be found in C:\WINNT\SYSTEM32. The ones in **bold** will be used a lot in up-coming labs.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCWIZ.EXE</td>
<td></td>
</tr>
<tr>
<td>ARP.EXE</td>
<td></td>
</tr>
<tr>
<td>ATMADM.EXE</td>
<td></td>
</tr>
<tr>
<td>CALCS.EXE</td>
<td></td>
</tr>
<tr>
<td>CALC.EXE</td>
<td></td>
</tr>
<tr>
<td>CDPLAYER.EXE</td>
<td></td>
</tr>
<tr>
<td>CLIPBRD.EXE</td>
<td></td>
</tr>
<tr>
<td>CLSPACK.EXE</td>
<td></td>
</tr>
<tr>
<td>CLEANMGR.EXE</td>
<td></td>
</tr>
<tr>
<td>CLICONFG.EXE</td>
<td></td>
</tr>
<tr>
<td>COMP.EXE</td>
<td></td>
</tr>
<tr>
<td>CONTROL.EXE</td>
<td></td>
</tr>
<tr>
<td>DDESHARE.EXE</td>
<td></td>
</tr>
<tr>
<td>DOSX.EXE</td>
<td></td>
</tr>
<tr>
<td>DOSSKEY.EXE</td>
<td></td>
</tr>
<tr>
<td>DRWTSN32.EXE</td>
<td></td>
</tr>
<tr>
<td>EVENTVWR.EXE</td>
<td></td>
</tr>
<tr>
<td>EDIT.EXE</td>
<td></td>
</tr>
<tr>
<td>EXPLORER.EXE</td>
<td></td>
</tr>
<tr>
<td>FAXCOVER.EXE</td>
<td></td>
</tr>
<tr>
<td>FAXSEND.EXE</td>
<td></td>
</tr>
<tr>
<td>FREECELL.EXE</td>
<td></td>
</tr>
<tr>
<td>FTP.EXE</td>
<td></td>
</tr>
<tr>
<td>GPRESULT.EXE</td>
<td></td>
</tr>
<tr>
<td>HOSTNAME.EXE</td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IEHWIZ.EXE</td>
<td></td>
</tr>
<tr>
<td>IEXPRESS.EXE</td>
<td></td>
</tr>
<tr>
<td>IMMC.EXE</td>
<td></td>
</tr>
<tr>
<td>IPCONFIG.EXE</td>
<td></td>
</tr>
<tr>
<td>IPSECMD.EXE</td>
<td></td>
</tr>
<tr>
<td>IRFTP.EXE</td>
<td></td>
</tr>
<tr>
<td>JVIEW.EXE</td>
<td></td>
</tr>
<tr>
<td>LPR.EXE</td>
<td></td>
</tr>
<tr>
<td>MAGNIFY.EXE</td>
<td></td>
</tr>
<tr>
<td>MEM.EXE</td>
<td></td>
</tr>
<tr>
<td>MOBSYNC.EXE</td>
<td></td>
</tr>
<tr>
<td>MPLAY32.EXE</td>
<td></td>
</tr>
<tr>
<td>MSPAINT.EXE</td>
<td></td>
</tr>
<tr>
<td>NARRATOR.EXE</td>
<td></td>
</tr>
<tr>
<td>NBTSTAT.EXE</td>
<td></td>
</tr>
<tr>
<td>NET.EXE</td>
<td></td>
</tr>
<tr>
<td>NETSH.EXE</td>
<td></td>
</tr>
<tr>
<td>NETSTAT.EXE</td>
<td></td>
</tr>
<tr>
<td>NOTEPAD.EXE</td>
<td></td>
</tr>
<tr>
<td>NSLOOKUP.EXE</td>
<td></td>
</tr>
<tr>
<td>NTBACKUP.EXE</td>
<td></td>
</tr>
<tr>
<td>NTDSUTIL.EXE</td>
<td></td>
</tr>
<tr>
<td>ODBCAD32.EXE</td>
<td></td>
</tr>
<tr>
<td>OSK.EXE</td>
<td></td>
</tr>
<tr>
<td>PACKAGER.EXE</td>
<td></td>
</tr>
<tr>
<td>PATHPING.EXE</td>
<td></td>
</tr>
<tr>
<td>PING.EXE</td>
<td></td>
</tr>
<tr>
<td>PERFMON.EXE</td>
<td></td>
</tr>
<tr>
<td>PROGMAN.EXE</td>
<td></td>
</tr>
<tr>
<td>RASADMIN.EXE</td>
<td></td>
</tr>
<tr>
<td>RCP.EXE</td>
<td></td>
</tr>
<tr>
<td>REGEDIT32.EXE</td>
<td></td>
</tr>
<tr>
<td>ROUTE.EXE</td>
<td></td>
</tr>
<tr>
<td>RUNAS.EXE</td>
<td></td>
</tr>
<tr>
<td>SECEdit.EXE</td>
<td></td>
</tr>
<tr>
<td>SETVER.EXE</td>
<td></td>
</tr>
<tr>
<td>SHRPUBW.EXE</td>
<td></td>
</tr>
<tr>
<td>SIGVERIF.EXE</td>
<td></td>
</tr>
<tr>
<td>SNDREC32.EXE</td>
<td></td>
</tr>
<tr>
<td>SNDVOL32.EXE</td>
<td></td>
</tr>
<tr>
<td>SOL.EXE</td>
<td></td>
</tr>
<tr>
<td>SYSEDIT.EXE</td>
<td></td>
</tr>
<tr>
<td>SYSKEY.EXE</td>
<td></td>
</tr>
<tr>
<td>TASKMGR.EXE</td>
<td></td>
</tr>
<tr>
<td>TELNET.EXE</td>
<td></td>
</tr>
<tr>
<td>TFTP.EXE</td>
<td></td>
</tr>
</tbody>
</table>
7. Let’s look at those in bold a little closer…type the command and /? or ? to find out the available options for the command.

   ARP.EXE
   FTP.EXE
   GPRESULT.EXE
   HOSTNAME.EXE
   IPCONFIG.EXE
   NBTSTAT.EXE
   NET.EXE
   NETSTAT.EXE
   NSLOOKUP
   PATHPING.EXE
   PING.EXE
   PERFMON.EXE
   ROUTE.EXE
   TELNET.EXE
   TFTP.EXE
   TRACERT.EXE

8. **DOSKEY.** One very nice command for use with DOS is the DOSKEY command. If you enable this during a DOS session you will be able to use the up and down arrows to recall any previously typed commands. This is very nice when you are trying to ping different computers on the same network. Try it, you’ll like it! (Hint: you can also use F3). This is turned on by default in Windows 2000.

9. **EDIT.** The DOS editor is used to match basic DOS files like batch files. Here you can read the contents of some files. Go through and select all options from each pull-down
menu to see what they do…don’t forget to read the help too! Save this file as rename.txt in a notepad or word document.

ECHO
ECHO    Let’s start those little buggers up!
ECHO
copy c:\temp\dave.txt c:\temp\dave.bak
copy c:\temp\matt.txt c:\temp\matt.bak
copy c:\temp\scott.txt c:\temp\scott.bak
copy c:\temp\tim.txt c:\temp\tim.bak
ECHO    ALL DONE!

Now copy that file and go into your DOS window. All you have to do to copy that into the DOS window is right click with your mouse. You will see something like this:

C:\Temp>ECHO
ECHO is on.
C:\Temp>ECHO    Let's start those little buggers up!
    Let's start those little buggers up!
C:\Temp>ECHO
ECHO is on.
C:\Temp>copy c:\temp\dave.txt c:\temp\dave.bak
    1 file(s) copied.
C:\Temp>copy c:\temp\matt.txt c:\temp\matt.bak
    1 file(s) copied.
C:\Temp>copy c:\temp\scott.txt c:\temp\scott.bak
    1 file(s) copied.
C:\Temp>copy c:\temp\tim.txt c:\temp\tim.bak
The system cannot find the file specified.
C:\Temp>ECHO    ALL DONE!
    ALL DONE!
C:\Temp>
So now go back and look at your temp directory and see if they were created:

C:\Temp>dir
Volume in drive C has no label.
Volume Serial Number is 1C6D-B558

Directory of C:\Temp
07/13/2004  11:45a      <DIR>          .
07/13/2004  11:45a      <DIR>          ..
07/13/2004  11:44a                  27 dave.bak
07/13/2004  11:44a                  27 dave.txt
07/13/2004  11:44a                  27 matt.bak
07/13/2004  11:44a                  27 matt.txt
07/13/2004  11:44a                  28 scott.bak
07/13/2004  11:44a                  28 scott.txt
6 File(s)     164 bytes
2 Dir(s)     69,598,878,720 bytes free
C:\Temp>

See if you can make a script to rename those and/or to delete those back up files now.
10. Ok, every now and then you may have to change the attributes of a file. Let’s start by looking at the attributes of those three txt files.

C:\Temp> attrib
A C:\Temp\dave.txt
A C:\Temp\matt.txt
A C:\Temp\scott.txt

C:\Temp>
What exactly does that mean? Well silly us we can find out with attrib /?

C:\Temp> attrib /?
Displays or changes file attributes.
ATTRIB [+R | -R] [+A | -A ] [+S | -S] [+H | -H] [[drive:] [path]
filename]
   [/S [/D]]
   + Sets an attribute.
   - Clears an attribute.
   R Read-only file attribute.
   A Archive file attribute.
   S System file attribute.
   H Hidden file attribute.
   /S Processes matching files in the current folder and all subfolders.
   /D Processes folders as well.

Sometimes we need to make some changes. Let’s say for example we do not want anyone to “see” the scott.txt file. So, let’s change its attribute to hidden:

C:\Temp> attrib +h scott.txt

Now, let’s go ahead and see the contents, or supposed contents, of our directory:

C:\Temp> dir
07/13/2004 11:45a      <DIR>          .
07/13/2004 11:45a      <DIR>          ..
07/13/2004 11:44a                  27 dave.txt
07/13/2004 11:44a                  27 matt.txt
2 File(s)       54 bytes
2 Dir(s) 69,597,230,592 bytes free

BUT! When we do a search for attributes on a directory we can “see” the hidden file:

C:\Temp> attrib
A C:\Temp\dave.txt
A C:\Temp\matt.txt
A H C:\Temp\scott.txt

C:\Temp>

Aha! Looks like good computer security stuff too! I will cover that in another book. Can you see the hidden folder in Windows Explorer? Aha! Users are
generally naïve about windows, let alone optional settings in Windows Explorer to find hidden directories. In Windows Explorer select Tools>Folder Options. Now the folder will not be seen in explorer. A good way to hide things, yes?

Supplemental Lab or Challenge Activity:
1. Go out to the web and find out what 8.3 means in regards to DOS (especially file names).
2. Write a batch file to install a \temp folder on the root drive of a computer and make it a hidden folder.

So What Have I Learned Here?
In this lab you have learned the basics of DOS. I find that many students do not have the experience with DOS that I had as I was brought up through the Commodore 64’s, IBM’s, 386’s, 486’s, etc. To me it is old-hat…to many newcomers though it is totally foreign. You will be using some DOS while you are working on many of the labs in this book so I thought it best to put it right up front. Keep referring back to this lab as often as you need to. Later in this section I have put another lab on “intermediate DOS.” Here you will learn about some DOS troubleshooting tools that you will probably use quite frequently. DOS is not dead. If you continue your studies you may even end up purchasing my computer security fundamentals book called the “Script Kiddie Cookbook.” In that book one of the labs is about stopping pop-up ads. Sometimes you need to use DOS to help determine how to best stop them.
Windows 2000 Utilities Lab

Objective:
To become better aware of utilities included with Windows 2000 Operating systems.

Tools and Materials:
(1) computer with Win 2000
paper and pencil

Background:
In this lab you will learn the answer to “Why didn’t anyone tell me these programs were here?” Well, quite simply, you have no one to blame but yourself. No one gives you anything for free (except for me), you have to go out and get it for yourself. As such, this lab is designed to help you explore little-publicized Windows utilities, some of which are pretty nifty. If you are not familiar with basic DOS commands you should do the DOS commands lab first. As a network administrator you will need to know basic DOS commands including: searching for files, wild-card characters, changing directories, and manipulating file names with DOS.

Step-By-Step Instructions:
1. Open the MS-DOS prompt into a full window.
2. Enable DOSKEY.
3. Start hunting for any executable, command, and batch files from the following prompts: root, windows subdirectory and windows/system subdirectory. Write down all files on your paper.
4. Go back and execute each file one at a time noting what happens. Some will do absolutely nothing noticeable. Be sure to check for any available subcommands and options using the DOS help feature.
5. Pare the list down to just the interesting programs.

Supplemental Lab or Challenge Activity:
6. Which programs did you find that may be useful to you as a network administrator?
7. If you had two different computers, one with 2000 and one with XP, what are the differences between the available programs?
8. Try a Windows ME or XP using the same techniques.
9. Make a chart comparing the “evolution” of programs in each operating system over time.
10. What has changed for the better, stayed the same, or changed for the worse?

So What Have I Learned Here?
This is actually almost a repeat of the DOS lab…I just wanted to make sure everyone realized the difference in the two and that no one skipped over either of these labs.
## Cool Windows 2000/XP/ME Utilities

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Dynamic DHCP Lab

Objective:
To learn about DHCP and how it works with a workstation.

Materials and Tools:
(1) Workstation on network with DHCP server

Background:
Most workstations connected to networks use a DHCP server from which to obtain their IP address automatically. As you found out in the multiple hub networks using static addresses can cause problems very quickly. In this lab you will learn how to release and renew the IP address and mask from your workstation using DOS commands and windows utilities. Later, you will learn how to set up your router to be a dhcp server.

Step-By-Step Instructions:
1. Open up a DOS window.
2. Then type “ipconfig” to see your IP settings using DOS. From DOS you should see something like this:

```
C:\Documents and Settings\basham.matt.ADMIN>ipconfig
Windows 2000 IP Configuration
Ethernet adapter Local Area Connection:
    Connection-specific DNS Suffix . : spcollege.edu
    IP Address . . . . . . . . . . . . : 192.168.151.60
    Subnet Mask . . . . . . . . . . . . : 255.255.255.0
    Default Gateway . . . . . . . . . : 192.168.151.1
C:\Documents and Settings\basham.matt.ADMIN>
```

3. It’s always a good idea to get a snapshot of the settings before we start changing them in case we need to put them back in later. Do not rely on your memory, write them down or print them out! Before we start changing these settings from DOS let’s explore the options available with the ipconfig command. I have highlighted the commands we are more likely to use as networking administrators. On the next page I took a quick snapshot and look at my options with ipconfig as well.
C:\Documents and Settings\basham.matt.ADMIN>**ipconfig /?**

Windows 2000 IP Configuration

**USAGE:**

```
ipconfig [/? | /all | /release [adapter] | /renew [adapter]
   | /flushdns | /registerdns
   | /showclassid adapter
   | /setclassid adapter [classidtoset] ]
```

**Options**

- `/?` Display this help message.
- `/all` Display full configuration information.
- `/release` Release the IP address for the specified adapter.
- `/renew` Renew the IP address for the specified adapter.
- `/flushdns` Purges the DNS Resolver cache.
- `/registerdns` Refreshes all DHCP leases and re-registers DNS names.
- `/displaydns` Display the contents of the DNS Resolver Cache.
- `/showclassid` Displays all the dhcp class IDs allowed for adapter.
- `/setclassid` Modifies the dhcp class id.

The default is to display only the IP address, subnet mask and default gateway for each adapter bound to TCP/IP.

For Release and Renew, if no adapter name is specified, then the IP address leases for all adapters bound to TCP/IP will be released or renewed.

For SetClassID, if no class id is specified, then the classid is removed.

**Examples:**

```
> ipconfig ... Show information.
> ipconfig /all ... Show detailed information
> ipconfig /renew ... renew all adapters
> ipconfig /renew EL* ... renew adapters named EL....
> ipconfig /release *ELINK?21* ... release all matching adapters, eg.ELINK-21,
   myELELINKi21adapter.
```

4. From DOS we can now type `ipconfig /release_all` to “let go” of our IP address. After doing that you should see:

C:\Documents and Settings\basham.matt.ADMIN>**ipconfig /release**

Windows 2000 IP Configuration

IP address successfully released for adapter "Local Area Connection"
Then we can use `ipconfig /renew_all` or `ipconfig /renew` to “get a new one” from the DHCP server. You should see:

```
C:\Documents and Settings\basham.matt.ADMIN>ipconfig /renew
Windows 2000 IP Configuration
Ethernet adapter Local Area Connection:
  Connection-specific DNS Suffix . : spcollege.edu
  IP Address . . . . . . . . . . . . : 192.168.151.60
  Subnet Mask . . . . . . . . . . . : 255.255.255.0
  Default Gateway . . . . . . . . . : 192.168.151.1
```

5. Notice how our address may differ slightly. When we give up our IP address it usually will go to one of the next devices requesting an IP…sometimes we get the same one back and sometimes we do not. Sometimes we encounter an error like this (and then do an `ipconfig`):

```
C:\Documents and Settings\basham.matt.ADMIN>ipconfig /renew
Windows 2000 IP Configuration
The following error occurred when renewing adapter Local Area Connection: DHCP Server unreachable
```

```
C:\Documents and Settings\basham.matt.ADMIN>ipconfig
Windows 2000 IP Configuration
Ethernet adapter Local Area Connection:
  Connection-specific DNS Suffix . : 
  IP Address . . . . . . . . . . . . : 169.254.55.102
  Subnet Mask . . . . . . . . . . . : 255.255.0.0
  Default Gateway . . . . . . . . . : 
```

Notice how our IP address is within the 169 network. Does this mean it worked? Not at all. Microsoft uses the “169 address” as a “place holder” in case something goes wrong with DHCP.

---

**So What Have I Learned Here?**

You have learned how to release and renew the DHCP address from a workstation using DOS. In later labs you will work more with DHCP and need to know how to do what we learned in this lab when setting up your routers to be DHCP servers.
Recent events at home and abroad have mandated that Americans can accept nothing short of the best possible training for those working in the fields of security and criminal justice. The Journal of Security Education: New Directions in Education, Training, and Accreditation is a comprehensive, one-stop resource on security education and training programs that will help educators, practitioners, and students meet the increasing need for security in the United States. Affiliated with the Academy of Security Educators and Trainers, the journal presents the latest developments in theory, practice, research, and assessment with an emphasis on up-to-date methods, techniques, and technology.
Objective:
In this lab you will complete the installation of the NIC by performing the software installation and changing TCP/IP settings. You will be changing TCP/IP settings in many of the labs in this book.

Tools and Materials:
(1) Workstation (2000)

Lab Diagram:

Step-by-Step Instructions:
In this lab you will be configuring only the workstation portion of the above lab diagram. It is just shown as an overall reference perspective.

1. Open the “My Network Places” icon on the desktop. You should see the network and dial up connections window (pictured below on left):
2. Then right click on the icon “local area connection” and select “properties.” You should see (pictured above on right):

3. Double-click on Internet Protocol (TCP/IP) or highlight Internet Protocol (TCP/IP) and select “properties.” In either case you should see another pop up window like this (picture below on left).

4. Now, say we are told to put in an IP address of 192.168.1.3 with a subnet mask of 255.255.255.0 and a gateway of 192.168.1.1. Here is how we would do it. First we would select “specify an IP address” and then put in IP address and mask on this window. After doing that the window should look like this (picture above on right).

5. Sometimes you can add in more than one gateway. For example if you have two routers connected to one switch and a workstation coming from that switch, as long as everyone is on the same subnet you have two possible “gateways” to route your information (see figure on next page). So, if you prefer one way over the other you can put the more preferred one in last and the least preferred one first (it moves it down when new ones are entered). Note: because there is more than one path (or way to get to the Internet or “home”) this type of network is often referred to as a “multi-homed network.”
To add another gateway click on the advanced tab. You should see:

![Advanced TCP/IP Settings](image)

Just click on the “add” tab and add in your second gateway. You can also change the metrics too…its almost like making a routing table on your PC.

6. Almost done. To finish it up we click on “ok” three times. You should then be prompted to reboot your computer to make the settings take effect. If you do not reboot then they will not work properly.
7. You can double-check your settings using those DOS or windows commands “\IPCONFIG.EXE.”

**Supplemental Lab or Challenge Activity:**
8. Try to find out about all of those other tabs and settings in the network and TCP/IP Properties windows.
9. What is a gateway?

**So What Have I Learned Here?**
Now you are talking about the “meat and potatoes” of things to come. In almost every lab you will be installing workstation TCP/IP settings. Heck, some of your troubleshooting will involve this later…I have seen it too many times before…“Mr. Basham, my computer doesn’t get any Internet!” The answer: “Did you see if the last student reset their TCP/IP settings back to obtaining them automatically?” Better learn it good now and **never** assume anything was put back properly.

---

**Earn your Bachelor’s Degree from St. Petersburg College in Technology Management**

Transferring and having access to a 4-year degree has never been easier, more convenient, or cost effective.

**St. Petersburg College** allows students who have received an Associate degree (Associate in Arts or Associate in Science, depending on the program) to transfer into the **Bachelor of Applied Science in Technology Management**.

Opportunities Include:
- On-line and in traditional classroom courses
- Day and evening classes
- Full and part-time course schedules

Explore the opportunities waiting for you at St. Petersburg College!

For more information call: (727) 341-4772  
E-mail: 4yeardegrees@spcollege.edu
Intermediate DOS Lab: Troubleshooting Utilities

Objective:
To learn about DOS utilities to use for troubleshooting in networks.

Tools and Materials:
(2) workstations
(1) cross-over cable (xo)

Lab Diagram:

Step-By-Step Instructions:
1. Cable the lab as shown.
2. Ask your instructor or buddy for help if necessary if you have problems with peer-to-peer networking. You may have to use the ip address of the other workstation as a gateway address. Sometimes yes, sometimes no...you just got to love Microsoft.
3. In this lab we will be using ping and trace route commands for troubleshooting (layer 3 commands). Let’s start by opening a DOS window and finding out what options are available with ping.

   C:\>ping /?
   [-r count] [-s count] [][-j host-list] [][-k host-list]
   [-w timeout] destination-list

   Options:
   -t           Ping the specified host until stopped.
   -a           To see statistics and continue - type Control-Break;
   -n count     To stop - type Control-C.
   -l size      Resolve addresses to hostnames.
   -f           Number of echo requests to send.
   -i TTL       Send buffer size.
   -f           Set Don't Fragment flag in packet.
   -v TOS       Time To Live.
   -r count     Type Of Service.
   -s count     Record route for count hops.
   -j host-list Timestamp for count hops.
   -k host-list Loose source route along host-list.
   -w timeout   Strict source route along host-list.
   -w timeout   Timeout in milliseconds to wait for each reply.

4. The first step in troubleshooting is testing layer 1 and working our way up the OSI model. Check the cabling. Be certain the LED on the NIC’s is lit up. You can also do a visual verification on the cable to be certain you are using the correct one. Just
because the light is lit does not mean the cable is working or is the proper cable. Be careful!

5. First we can test the functionality of the NIC (layers 1-2) and the computer for its ability to communicate with networking. We can do this by using ping to any address on the 127.0.0.1-127.255.255.254 network. This is called the “loopback adapter network.” So I pick an IP address from the 127 network and ping it. You should see something like this if everything is fine:

C:\ > ping 127.127.127.127

Pinging 127.127.127.127 with 32 bytes of data:

Reply from 127.127.127.127: bytes=32 time<10ms TTL=128
Reply from 127.127.127.127: bytes=32 time=1ms TTL=128
Reply from 127.127.127.127: bytes=32 time=1ms TTL=128
Reply from 127.127.127.127: bytes=32 time=1ms TTL=128

Ping statistics for 127.127.127.127:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\ >

6. Next we can test our basic network connection between the two computers using ping (layer 3). If my workstation used 192.168.1.1 and the other one used 192.168.1.2 then I would ping 192.168.1.2 to test connectivity. If you cannot ping the other workstation then check the IP addresses and masks on each workstation. When all else fails reboot the workstations too.

C:\ > ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time<10ms TTL=128
Reply from 192.168.1.2: bytes=32 time=1ms TTL=128
Reply from 192.168.1.2: bytes=32 time=1ms TTL=128
Reply from 192.168.1.2: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.1.2:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 0ms, Maximum = 1ms, Average = 0ms

Ok…time to play with our new found ping friend. Let’s see what some options are for ping and what they do. First adding the –t option will cause multiple pings UNTIL YOU STOP IT by using the break sequence in DOS (control+C)...this is technically illegal because it creates a very, very small denial of service attack:
C:\ > ping 192.168.1.2 -t
Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time<10ms TTL=128
Reply from 192.168.1.2: bytes=32 time=1ms TTL=128
Reply from 192.168.1.2: bytes=32 time=1ms TTL=128
Reply from 192.168.1.2: bytes=32 time=1ms TTL=128
Reply from 192.168.1.2: bytes=32 time=1ms TTL=128
Reply from 192.168.1.2: bytes=32 time=1ms TTL=128
Reply from 192.168.1.2: bytes=32 time=1ms TTL=128

(control+C stops it)

Ping statistics for 192.168.1.2:
Packets: Sent = 7, Received = 7, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms

Why do this? Let’s just say we start it up on one machine and it is telling us that it is not replying…by using the constant ping we can “see” the instant the other computer or interface comes on-line. This is very handy later when you will be doing access control list labs. Ok…lets try another one. Adding the –n will let us specify how many packets to send. Sometimes waiting for four packets can be problematic, so we just want to send one.

C:\ > ping 192.168.1.2 –n 1
Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time<10ms TTL=128
Ping statistics for 192.168.1.2:
Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms

Now, the “mother of them all”…adding the –l will let us change the size of our packet from 32 bytes to whatever we want it to…sometimes during labs you may want to see how much it would take to “choke” out the performance of an interface or to test some traffic balancing and this would work for it. Actually a Linux box would work way better for actually choking something out but you should get the point with this:

C:\ > ping 192.168.1.2 –l 50000
Pinging 192.168.1.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms

Huh? What happened to our one ping and why didn’t it get “received?” Yeah, you can only have so big of a size go round trip through DOS on a Windows-based
workstation. I even set it down to 5000 bytes and got the same thing. From a Linux box it worked no problem. We can combine these too:

C:\ > ping 192.168.1.2 -l 5000 -n 2

Pinging 192.168.1.2 with 5000 bytes of data:

Reply from 192.168.1.2: bytes=5000 time=10ms TTL=30
Reply from 192.168.1.2: bytes=5000 time=10ms TTL=30

Ping statistics for 192.168.1.2:
   Packets: Sent = 2, Received = 2, Lost = 0 (0% loss)
Approximate round trip times in milli-seconds:
   Minimum = 10ms, Maximum = 10ms, Average = 10ms

One last thing here. You can open MULTIPLE DOS windows too. Try it. Go to the “run” panel and type in “cmd” and then repeat it several times. Try taking the ip address out of one of your workstations. Then put a continuous ping from the workstation (with the good ip address) to the one without. Watch it for a couple of seconds and then put the address back in. You should see the ping packet replies almost instantly. Here’s another fun one…ping the broadcast address (192.168.1.255). Why does it work? You will find out later or ask your instructor if you really need to know right now, right now.

7. We know we have good connections between the two. When you have more than two computers in a network you can also use another layer 3 tool: trace route. Let’s start by looking at our options with tracert in DOS:

C:\ > tracert
Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout] target_name
Options:
   -d         Do not resolve addresses to hostnames.
   -h maximum_hops Maximum number of hops to search for target.
   -j host-list Loose source route along host-list.
   -w timeout Wait timeout milliseconds for each reply.

If you are having difficulty connecting to another device several hops away trace route will show you exactly which device “looses” your communication. For example, if I had a network with several routers and was trying to get to www.spjc.edu I could find the faulty device. First, since it helps to have a baseline before something goes bad let’s look at a good trace route to our destination:

C:\ > tracert www.spjc.edu
Tracing route to www.spjc.edu [172.16.1.68] over a maximum of 30 hops:
   1 1 ms 1 ms 1 ms 192.168.151.1
   2 4 ms 5 ms 5 ms 192.168.154.1
   3 5 ms 7 ms 4 ms do-esr5000 [172.23.1.1]
   4 6 ms 6 ms 6 ms 192.168.100.27
   5 6 ms 6 ms 6 ms www.spjc.edu [172.16.1.68]
Trace complete.
Now, when troubleshooting if we ran a trace route and got this:

```
C:\ > tracert www.spjc.edu
Tracing route to www.spjc.edu [172.16.1.68] over a maximum of 30 hops:
  1  1 ms  1 ms  1 ms  192.168.151.1
  2  4 ms  5 ms  5 ms  192.168.154.1
  3  5 ms  7 ms  4 ms  do-esr5000 [172.23.1.1]
  4 *  *  *  Request timed out
  5 *  *  *  Request timed out
Trace complete.
```

Then we would have a good idea there is a problem with the do-esr5000 device with IP address 172.23.1.1. In this case it’s a 5000 series router at district office. If it does not work at all have your instructor check with your school’s network administrator, then some of them have been denying icmp traffic within the school. Let’s do another tracert, this time to www.yahoo.com

```
C:\ > tracert www.yahoo.com
Tracing route to www.yahoo.akadns.net [216.109.117.110] over a maximum of 30 hops:
  1 <10 ms <10 ms <10 ms 192.168.151.1
  2 <10 ms <10 ms  10 ms 192.168.154.1
  3 <10 ms  10 ms  10 ms do-esr5000 [172.23.1.1]
  4  10 ms <10 ms  10 ms 192.168.100.27
  5  10 ms  10 ms <10 ms 192.168.255.3
  6 *  *  *  Request timed out.
  ...  
 18 *  *  *  Request timed out.
 19 40 ms 40 ms 30 ms p25.www.dcn.yahoo.com [216.109.117.110]
Trace complete.
```

You can see we had a lot of time outs here and not a whole lot of information. Tracert is limited in DOS but can occassionaly yield some good information.

Keep track of updates and changes at [http://www.spcollege.edu/star/cisco](http://www.spcollege.edu/star/cisco) Scroll to the bottom of the page and click on the “Lab Manual Edits.”
Basic Troubleshooting

Check cabling and lights  
Layer 1

Ping the loopback adapter  
Layer 1-2

Ping, trace route  
Layer 3

Still stuck?  Ask your instructor or a buddy for help.

Supplemental Lab or Challenge Activity:
1. Is there an upper limit in DOS to the size of packet that you send?
2. Open up multiple DOS windows and send pings to each workstation in your classroom only at the same time.
3. Go find out what a traffic generator is…how could you use your knowledge of ping to make a traffic generator?
4. Make a traffic generator using ping commands that will choke out your network. You will know it is working when they start timing out. Figure out the optimal ping size that starts choking the network and the maximum size just before the network chokes. This will be cool to use later to test your networks.
5. Sometimes in your reading you are hearing about “network broadcasts.” How can you make a network broadcast using the “ping” command?
6. What are the similarities and differences between ping, tracert, and pingpath?

So What Have I Learned Here?
In this lab you learned the basics of troubleshooting workstation network problems. You will be using this knowledge as you “Learn by Doing” and practicing for your CCNA Exam.
Objective:
To learn the basics about file transfer programs.

Background:
The File Transfer Program (FTP) has probably been used by nearly everyone who uses
the web, whether they know it or not. This program is used to transfer files from one
computer to another. The Trivial File Transfer Program (TFTP) is a similar program but
is used for more specific applications like downloading software to a router (like a
CISCO router...aha!). Here you will learn how to use FTP and its basic commands to
upload and download a file. In a later lab you will use the similar TFTP program to
download an operating system to a router.

Step-by-Step Instructions:
1. Open the MS-DOS prompt.
2. Type “ftp ftp1.ipswitch.com”
3. When prompted use “anonymous” and joe@hotmail.com for password (use your
e-mail address). If you log in correctly you will see:

   C:\WINDOWS\Desktop>ftp ftp1.ipswitch.com
   Connected to ftp1.ipswitch.com.
   220-ftp1.ipswitch.com X2 WS_FTP Server 5.0.4 (1155495796)
   User (ftp1.ipswitch.com:(none)): anonymous
   331 Password required
   Password:
   230 user logged in
   ftp>

4. Type “dir” to see what files and directories are available. List those here:

   ________________________________________________________________
   ________________________________________________________________

   Be sure to be polite and not cause any problems...we don’t want them to not allow us
to use this site for education.
5. Type cd ipswitch to change to the “ipswitch” sub-directory.
6. Type dir to see what files and directories are available.
7. Type cd manuals to change to the “manuals” sub-directory.
8. Type dir to see what files and directories are available.
9. Type cd msdos to change to the msdos sub-directory.
10. Type dir to see what files and directories are available.
11. Type “get ftpserv.pdf” to download the file to your computer.
12. Type “lcd” to find out where it put it on your computer. Where did it go?
13. Now you can go out an open the program. You should have a pdf file of a user’s
guide for using WS_FTP server.
14. Type ? to see what commands are available. Write them down.
15. Type help ____ for each command for a more detailed explanation of each
    command...for example the first one listed is “!” so type “help !” and write down
    what it says.
16. I would tell you how to leave the session but you will be able to figure out many ways to do it after you explore those commands a bit.

Supplemental Labs or Challenge Activities:
1. Go out and find a program called “CuteFTP” and compare it to FTP.
2. Your instructor will have the TFTP program (or you can download it from CISCO). How do these programs differ?
3. Go out and search the web for ftp sites. Go on and play a bit…have fun!

So What Have I Learned Here?
You have learned about basic FTP commands and how FTP works. I have seen some CCNA test review software that ask about the FTP commands (get and put specifically) so I wrote this lab for all of you. Ain’t that nice?

Hungry for more???
You are almost there for the CompTIA Network+ and Security+ Certifications too!
For more information:

http://www.comptia.org/certification/default.aspx

Building successful careers for students everywhere…
Objective:
To learn how to use terminal emulation (TELNET) software for Internet connectivity.

Background:
During your studies you will use many different software packages: FTP, TFTP, DOS, Protocol Inspector, and now you will learn TELNET. We saw it briefly back in the DOS lab but now we will use it to visit government sites, gopher sites, and other types of sites. We will also look briefly at “port-surfing.”

Step-By-Step Instructions:
1. Open the telnet application. A quick way to do this is to click on Start>Run then type in telnet and press “ok.” You should see the DOS prompt come up like this:

Microsoft Telnet>

2. Start by reviewing everything in the help files. This will acquaint you more with what telnet can and cannot do.

3. Let’s start with an easy one. Let’s telnet to the Library of Congress. Start with this:

Microsoft Telnet>open locis.loc.gov

The screen should go blank and then come up like this:

L O C I S: LIBRARY OF CONGRESS INFORMATION SYSTEM
To make a choice: type a number, then press ENTER
1 Copyright Information -- files available and up-to-date
2 Braille and Audio -- files frozen mid-August 1999
3 Federal Legislation -- files frozen December 1998
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
The LC Catalog Files are available at:
http://lcweb.loc.gov/catalog/
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

8 Searching Hours and Basic Search Commands
9 Library of Congress General Information
10 Library of Congress Fast Facts

12 Comments and Log Off
Choice:

I would like to add a brief note here that this site seems to be up and down a lot. If you just get a DOS window with nothing on it...just assume the library of congress telnet site is not functioning or that telnet may be disabled on your network.

4. Let’s try to telnet to a “MUD” site (multiple user dungeon)...it’s a gaming site.

Microsoft Telnet>open entconn.com
5. If there is an available line you will see (this is my old Windows 98 picture but the information will be very similar):

![Telnet Connection](image)

6. Fun…isn’t it? Here’s one for the “Hard Drive Café” at St. Petersburg College:

   Microsoft Telnet> `open hdcbbs.net`

   Synchronet BBS for Win32 Version 3.12 Copyright Rob Swindell
   Client CONN: Telnet
   ADRR: seffw.spcollege.edu [66.194.104.5]
   SERVER NAME: The Hard Drive Café
   ADRR: hdcbbs.net
   NODE: 2 (of 10)
   Date: 2/26/05
   TIME: 09:17 am
   ADMN: Jason

7. You can also telnet to specific ports on the computer. We could also telnet in to port 23 on the same machine (the telnet port). Like this:

   C:\Documents and Settings> `open hdcbbs.net 23`

8. We can telnet to all kinds of sites. This is not used as much anymore because everyone pretty much uses http on port 80. If you know how to use it you can really zip around and you can find much more information (although some of it is older). Think about it…the web sites will tell you where to buy the book, but telnet/BBS/FTP sites may have the full text documents…they have been around a lot longer than the “commercial Internet.” On the next page you will find some “fun ports to surf.” Let’s try one that “mixes” using Telnet, Hyperterminal and BBS all in one. Go to [http://www.homershut.net/telnet_sites/telnet_sites.html](http://www.homershut.net/telnet_sites/telnet_sites.html) and then click on Homer’s Hut. You will see the web browser open up a hypterminal session to “telnet” into the Homer’s Hut BBS. Do you want to see something interesting? Click on the link for “Thorny’s.” See anything familiar? Cool huh?
Supplemental Labs or Challenge Activities:
1. Go out and perform the tutorials on how to use telnet and its associated websites: http://www.cs.indiana.edu/docproject/zen/zen-1.0_7.html
2. Find some more BBS and telnet sites at http://www.telnet.org/. It’s fun for the whole family. One cool site I saw was an “asciimation” of the star wars movie. Telnet to towel.blinkenlights.nl Also you can telnet to a game site at shadowlands.com on port 4000.
3. Go out and find all port numbers and their associations.

So What Have I Learned Here?
You have learned about more utilities that can be used, but are not used as much anymore. Let’s face it…it’s the old school stuff…unforgiving, DOS-like, tough to use programs. The Internet is easier, but this will help “round you out.”

Fun Ports to Surf with Telnet

To open Telnet, go to START, then RUN, and type “TELNET” then press enter.

***Be careful when surfing telnet ports. If you are not authorized on anyone’s computer then you will be guilty of a felony!****

<table>
<thead>
<tr>
<th>Port</th>
<th>Service</th>
<th>What it is…</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Echo</td>
<td>Whatever you type in is repeated</td>
</tr>
<tr>
<td>9</td>
<td>Discard/null</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Systat</td>
<td>Lots of info on users in network</td>
</tr>
<tr>
<td>13</td>
<td>Daytime</td>
<td>Time and date at computer’s location</td>
</tr>
<tr>
<td>15</td>
<td>Netstat</td>
<td>Lots of info on network—a must see!</td>
</tr>
<tr>
<td>19</td>
<td>Chargen</td>
<td>ASCII character stream</td>
</tr>
<tr>
<td>20</td>
<td>ftp</td>
<td>ftp data</td>
</tr>
<tr>
<td>21</td>
<td>ftp</td>
<td>Transfer files (control)</td>
</tr>
<tr>
<td>23</td>
<td>telnet</td>
<td>Terminal emulation program</td>
</tr>
<tr>
<td>25</td>
<td>Smtp</td>
<td>Mail program</td>
</tr>
<tr>
<td>37</td>
<td>Time</td>
<td>Time</td>
</tr>
<tr>
<td>39</td>
<td>Rlp</td>
<td>Resource location</td>
</tr>
<tr>
<td>43</td>
<td>Whois</td>
<td>info on hosts and networks</td>
</tr>
<tr>
<td>53</td>
<td>Domain</td>
<td>Name server</td>
</tr>
<tr>
<td>70</td>
<td>Gopher</td>
<td>Out-of-date information tool</td>
</tr>
<tr>
<td>79</td>
<td>Finger</td>
<td>UNIX information finder</td>
</tr>
<tr>
<td>80</td>
<td>http</td>
<td>Web server</td>
</tr>
<tr>
<td>107</td>
<td>rTelnet</td>
<td>Remote telnet</td>
</tr>
<tr>
<td>110</td>
<td>Pop</td>
<td>Email post box server</td>
</tr>
<tr>
<td>113</td>
<td>Ident/auth</td>
<td>Identification/authorization</td>
</tr>
<tr>
<td>119</td>
<td>nntp</td>
<td>News group servers</td>
</tr>
<tr>
<td>135</td>
<td>Epmap</td>
<td>DCE endpoint resolution</td>
</tr>
<tr>
<td>139</td>
<td>Netbios</td>
<td>Netbios session service</td>
</tr>
</tbody>
</table>
Hyperterminal Lab

Objectives:
Learn how to set up a router and login through a router console port from a workstation using the Hyperterminal program.

Tools and Materials:
Workstation with Hyperterminal Program
CISCO router
(1) rollover cable (ro)

Background:
“Easy when you know how…” is very applicable when accessing a router through a workstation. This lab is designed to show you how to set up the hyperterminal program, to connect cabling and how to access the router.

Lab Diagram:

Step-By-Step Instructions:
1. Verify the existence of the hyperterminal program on your Windows workstation. Check this path: Start>Programs>Accessories>Hyperterminal or Start>Programs>Communications>Hyperterminal. If you do not have it installed on your workstation, then follow these steps (you will probably need your Windows CD):
   1. go to Start>Settings>Control Panel>Add/Remove Programs
   2. select the middle tab “Windows Setup”
   3. select “Communications”
   4. select the “Hyperterminal” pick box
   5. follow the prompts to finish the installation
2. Open the Hyperterminal folder/program using the path you just found.
3. Open the “hypertrm” icon.
4. Type in a name for the session and select an icon.
5. Pick “Connect using direct to COM1”
6. Make sure you have the following settings:
   - 9601 bits per second
   - 8 data bits
   - None parity
   - 1 stop bit
   - Hardware flow control

Later on you may have to change these settings. Some switches (like Cabletron) like to use flow control set to “none” instead of “hardware.”

7. Connect the router from the console port to COM1 on your workstation using a rollover cable. You may need to add in a DB-9 to RJ-45 adapter to your COM1 port.

8. Now you can turn the power “on” to the router. After a couple of seconds you should start seeing some information on the Hyperterminal window.

Troubleshooting:
Are you connected to COM1? Do you have a rollover cable? Is your rollover cable good? Do you have your Hyperterminal settings correct? Is COM1 correctly set up in your BIOS?

Supplemental Lab or Challenge Activity:
1. Go search the Internet for instructions on COM ports, their settings, and what they do. Why do we set to 9600 bps, 8 databits, no parity, and 1 stop bit? What is parity?
2. Look up a program called “Kermit” on the web. How does it differ from Hyperterminal? What about “Xmodem?”
3. Go to downloads.com and see if there are other communications software packages available.
4. Go to www.sigmanet.com and download the utilities for the Adtran Atlas 550. They have a communication tool package their too. See if you can use their communication package to hyperterminal into a router too.
5. Is hyperterminal only for routers? Try it by connecting to lynx.cc.ukans.edu
6. It is possible to capture text from a hyperterminal session and save it to a text file WHILE you are working. In this manner you can see everything you did during an active session. Click on the “transfer” pull-down menu, then enter a path and file name to save it too. It’s just that easy!

So What Have I Learned Here:
Another day, another utility to use. Gosh! Will they ever stop? Oh who cares…more knowledge, more tricks in our arsenal, more lines on the resume. We learned about some more communication software. Hyperterminal is going to be used quite a lot through out the rest of this book. Who know? Be different and use another communications tool to access the router and impress your friends or just show off smugly.
Remote Access Lab

Objective:
To learn how to set up windows dial-up networking (DUN) and connect to another computer to share files.

Materials:
(3) PC workstations
(3) External Hayes modems (or internals if you must)
(3) RS-232 to DB-9 adapters
(3) RJ-11 (phone cords)
(1) Adtran

Lab Diagram:

Background:
Setting up DUN is easy. There are three steps: (1) configure a connection on the PC, (2) configure the communication rules, and (3) set up to receive calls.
(Step 1) Configure a connection on the PC

1. Check to see if your computer has dial-up networking capabilities first. If not, then you will have to install dial-up networking software from your Windows installation CD.
2. Double-click on the “my computer” icon on your desktop.
3. If you have a folder called “dial-up networking,” then you have DUN installed and are ready to go!
4. If not, then you will have to install DUN.
   1. Click on Start>Settings>Control Panel>Add/Remove Programs
   2. Click on the tab for “Windows Setup”
   3. The computer will search for settings. Then select “Communications.”
   4. Select “Dial-up Networking.”
5. Select “Dial-up Server.” This will allow you to receive calls.

Figure 1—Select Dial-up Networking and Dial-up Server.

6. Click on “ok.” You may be prompted for the Windows installation CD rom.

If you are doing this at school, then chances are your school network administrator may have put the installation files (*.cab files) on the computer (so you won’t need the cd). These are files that contain compressed images of the Windows operating system. A long time ago, before CD-roms, we had to install operating systems from floppy diskettes. These *.cab files are an off-shoot from those days. Currently your operating system may need as many as 30-35 floppy diskettes to make a back-up copy from the CD-rom. In the “old-days” we could make back-up copies with seven floppy diskettes (Windows 3.x) or even three (DOS).

ii. Click on “ok.”

iii. You may have to re-boot your computer.

7. Check to see if your computer has a modem and software installed.
8. Click on Start>Settings>Control Panel>Modems. If you have one installed, then you should see one here. It may look like this:

![Modems Properties window](image)

Figure 2—Laptop with PCMCIA modem card installed.

9. If you do not have one, then you will have to add one. We will walk through adding an external modem to your computer here.
   1. Click on “start>settings>control panel>add new hardware
   2. When the “add new hardware wizard” opens click on “next” twice.
   3. Click on “no, I want to select the hardware from a list.”
   4. Then click on “next.”
   5. Select “modem.”
   6. Click on “next.”
   7. Make sure your modem is connected. I used an RS-232 adapter on the DTE of the modem to a DB-9 connector on my COM1 port. The RS-232-DB-9 were on the ends of one cord.
   8. Click on “next.” The computer should find your modem on COM1.
   9. If not, select “next.”
   10. Select “have disk” and change to the CD-drive.
   11. Select the modem. I selected Hayes V.90 PCI modem for my external one.
   12. Select “next.”
   13. Select “finish.”

9 Make yourself a new connection. You can actually make many different connections with each one set up to dial a different number. In our lab diagram above we could make three different connections, one for each different user, and put icons on the desktop to make it easier to dial. To make a dial-up connection:
   1. Double-click on “my computer”
   2. Double-click on “dial-up networking”
   3. Click on “make a new connection”
   4. Give the connection a name (matt, scott, dave, etc)
   5. Select a modem to use
6. Click on “next”
7. Put in the phone number to call…In our example if I was configuring “matt” to call “dave” then I would use 555-6003.
8. Select a country or region code (US)

(Step 2) **Configure the communication rules**

10 Configure the communication rules (“protocols”):
   1. On that connection we just made in the dial-up networking folder, right-click it
   2. Select “properties”
   3. Make optional selections in the next steps.

11 Along the top you will see some tabs to configure various communications rules for this connection (step 6-10 explain these settings in more detail):
   1. Server types—will allow you to select the type of dial-up server to be called along with some optional settings, will allow you to select the “allowable network protocols,” and will allow you to see or change your TCP/IP settings.
   2. Scripting—allows us the option to use a modem script or another type of script for the dial-up access.
   3. Multilink—allows us the option of using multi-link for connections.

**Server Types Tab:**

12 For most connections you will probably just use a connection to a “PPP: Internet, Windows NT Server, Windows 98” dial-up server. This is usually used at home to dial into an ISP like AOL, MSN, or Netcom. Your ISP should be able to walk you through these steps via technical support or will have “self-installing” software to do this for you.

13 You can select any “advanced options.”
   1. Log on to network—Used only when using DUN to have access to a Microsoft NT controlled network. Most ISP’s run on UNIX so you probably will not need this.
   2. Enable software compression—if your ISP requires use of compression technologies (most do not) then select this.
   3. Require encrypted password—Almost all dial-in connections require a password. Select this only if your password must be encrypted. Since the encryption settings must be identical on each end, changes are, at this point in your networking career, you won’t need this to be enabled.
   4. Require data encryption—Ditto..this just encrypts the data.
   5. Record a log file for this connection—with this enabled a record of all activities during the connection will be made. This is similar to keyboard recorders except more information is included.

14 You can select any “allowable network protocols.” This helps to establish the routed protocols to be used during your connection. Ok, ok, so netbeui is non-routable…don’t sue me for Microsoft putting it here…actually netbeui is encapsulated within another protocol to allow it to be routed. Select TCP/IP
for your networking connection. Most of the time you will be using this
protocol suite. Heck, even Macintoshes and Novell use TCP/IP. If you want
to check all three to feel safer, then go ahead. Just be aware that IPX sends
out its own little broadcasts every 60 seconds which can affect the
performance of your connection.

Scripting Options Tab:
15 Here you can select a file with script settings to establish the DUN. You can
also select if you want the script lines to be “stepped” through which means
you will be prompted (asked) before each line if you wish that line to be
processed. Finally you can select if you want the terminal screen to be
minimized when you start.

Multilink Tab:
16 Multilink will allow you to use additional devices for establishing and
maintaining connections. Think of this as something like a “conference call.”

(Step 3) Set up to receive calls
17 In the dial-up networking window select “connections” from the pull-down
menu and then “Dial-up server.”
18 Select “allow caller access.”

![Figure 3—Setting up Dial-up Server.](image)

19 Put a password in if you want.
20 Click on “server type.”
21 Select “Type of Dial-up Server.”
22 Select “PPP: Internet, Windows NT Server, Windows 98.”
23 Disable “Require encrypted password” if none will be used.
24 Click on “OK” twice. You are now set up to send and receive calls.

Step-By-Step Instructions:
1. You are to establish, maintain, and tear-down DUN’s on Matt’s, Scott’s and
Dave’s workstations to each other. You will then share files between each of
the workstations. To begin you need to make some files and folders for sharing.
2. On each computer make a folder for each user.
   i. On Matt’s computer make a folder called c:\matt
   ii. On Scott’s computer make a folder called c:\scott
   iii. On Dave’s computer make a folder called c:\dave

3. On each computer put an IP address in the TCP/IP setting for each dial-up adapter. Use 192.168.1.1/24 for Matt, 192.168.1.2/24 for Scott, and 192.168.1.3/24 for Dave. This is not the same TCP/IP setting you have been using. See figure 4. How you set them will look identical. Just make sure you pick the right one.

![Figure 4—Selecting the TCP/IP for the Dial-up Adapter](image)

4. On each computer make a text document for each user.
   i. On Matt’s computer
      1. in c:\matt make a document called c:\matt\matt.txt
      2. In that document write “This is Matt’s file”
   ii. On Scott’s computer
      1. in c:\scott make a document called c:\scott\scott.txt
      2. In that document write “This is Scott’s file”
   iii. On Dave’s computer
      1. in c:\dave make a document called c:\dave\dave.txt
      2. In that document write “This is Dave’s file”

2. Make DUN’s for each computer to contact each other. Here are instructions for making a DUN to Scott on Matt’s computer:
   1. Open “my computer.”
   2. Double-click on “dial up networking” folder
   3. Double-click on “make a new connection.”
4. Give a name to the connection
5. Select modem to use.
6. Click on “next.”
7. Put in the phone number.
8. Click on “next.”
9. Click on “finish.”
10. If you need to change any properties then go back and right-click the DUN and make the changes.

3. Have Matt establish a DUN to Scott. You will see a window similar to Figure 5 when you are connected. Go ahead and select “more information” to see what is available to you.

Figure 5a—Dialing to connect to Dave from Matt via dial-up networking.

Figure 5b—Verifying user name and password (none) to connect to Dave from Matt.

Figure 5c—Logging on to the network to connect to Dave from Matt.
Figure 5d—Isn’t that nice?

4. Copy c:\scott\scott.txt into c:\matt\. Can’t find the other computer in “networking neighborhood?” In the DOS window try to ping it. If it returns a ping, then it is there and windows is being difficult. In windows explorer search for the computer using the “find” utility under the tools menu. Search by IP address and it should be found. If not, then re-check your IP settings.

5. On Matt’s computer open explorer and verify there are now two files in c:\matt. If not, then double-check your file and print sharing. You may see a window similar to figure 6 during the connection. If not, then go back into the dial-up networking window, click on “connect to Dave” and then “details.”

Figure 6—Active connection with DUN.

6. Close the connection.
7. Have Matt establish a DUN to Dave.
8. Copy c:\dave\dave.txt into c:\matt."
9. On Matt’s computer open explorer and verify there are now three files in c:\matt. If not, then double-check your file and print sharing.
10. Close the connection.
11. Have Scott establish a DUN to Matt.
12. Copy c:\matt\matt.txt into c:\scott\.
13. On Scott’s computer open explorer and verify there are now two files in c:\scott.
   If not, then double-check your file and print sharing.
14. Close the connection.
15. Have Scott establish a DUN to Dave.
16. Copy c:\dave\dave.txt into c:\scott\.
17. On Scott’s computer open explorer and verify there are now three files in c:\scott.
   If not, then double-check your file and print sharing.
18. Close the connection.
20. Copy c:\matt\matt.txt into c:\dave\.
21. On Dave’s computer open explorer and verify there are now two files in c:\dave.
   If not, then double-check your file and print sharing.
22. Close the connection.
23. Have Dave establish a DUN to Scott.
24. Copy c:\scott\scott.txt into c:\dave\.
25. On Dave’s computer open explorer and verify there are now three files in c:\dave.
   If not, then double-check your file and print sharing.
26. Close the connection.

Ok…so it was a bit of over-kill doing connections to everyone else but you know they all work now and can share any files between them.

Supplemental Lab or Challenge Activities:
1. Turn on logging. Find the log file and view the contents after a connection is closed.
2. Share only certain files.
3. Use a protocol inspector to view session establishments.
4. Set up three computers to simulate ISP’s.
5. Instead of using the dial-up networking try using Hyperterminal. Go ahead get crazy and type stuff in too!

Take college courses on a new schedule--yours.
Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education. Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

http://e.spcollege.edu

St. Petersburg College’s eCampus. Your time. Your place. Your future.
Your Modem and You

Objective:
This lab will familiarize you with the features of modems, the AT command set, and modem scripts. This lab is more information-based than hands-on oriented.

Tools and Materials:
(3) PC workstations
(3) External Hayes modems (or internals if you must)
(3) RS-232 to DB-9 adapters
(3) RJ-11 (phone cords)
(1) Adtran

Lab Diagram:

Background:
Modem configurations vary by manufacturer. Fortunately some vendors have attempted to follow a “AT command set” standard (non-formalized). It is not really a standard, or protocol, just an attempt to be consistent (how nice for us!). When you buy a modem you should receive a modem configuration book, disk or CD (or at least instructions on where to download them). Fear not! On the CISCO website there is a comprehensive AT command set book (76 pages!). You should go download that if you want thorough knowledge of AT command sets.
Modems use their own little language. Every language has its own alphabet and modem-speak is no different. Here is the common “alphabet” of modem-speak:

- a-z “alphabet”
- ^ “carat”
- $ “dollar sign”
- % “percent sign”
- & “ampersand”
- ) “parenthesis”
- * “asterisk”
- - “hyphen”
- : “colon”
- @ “character command set”
- \ “backslash”
- # “character command set”

Each one is unique and each one can be command with other “alphabet letters” to make scripts in modem-speak. I have filled in a chart with some common commands for my Hayes modem and what they do. Complete the chart with commands for your modem.

<table>
<thead>
<tr>
<th>Your modem</th>
<th>My Hayes V.90</th>
<th>Description of command</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>Attention</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Dial</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>Hang up</td>
<td></td>
</tr>
<tr>
<td>^V</td>
<td>Display bootstrap revision</td>
<td></td>
</tr>
<tr>
<td>$B57600</td>
<td>Set serial port to 57600 bps</td>
<td></td>
</tr>
<tr>
<td>$D</td>
<td>Run power-up diagnostics</td>
<td></td>
</tr>
<tr>
<td>%M</td>
<td>Set modulation</td>
<td></td>
</tr>
<tr>
<td>&amp;C1</td>
<td>Set up modem for carrier detect</td>
<td></td>
</tr>
<tr>
<td>&amp;D3</td>
<td>Set up modem for when the data terminal ready (dtr) transitions to “off”</td>
<td></td>
</tr>
<tr>
<td>&amp;F</td>
<td>Load factory defaults and settings</td>
<td></td>
</tr>
<tr>
<td>&amp;K3</td>
<td>Set hardware flow control</td>
<td></td>
</tr>
<tr>
<td>&amp;Q9</td>
<td>Set compression</td>
<td></td>
</tr>
<tr>
<td>&amp;T</td>
<td>Diagnostic test mode</td>
<td></td>
</tr>
<tr>
<td>&amp;W</td>
<td>Save configuration to modem</td>
<td></td>
</tr>
<tr>
<td>-D</td>
<td>Repeat dial</td>
<td></td>
</tr>
<tr>
<td>@E</td>
<td>Detailed modem call status</td>
<td></td>
</tr>
<tr>
<td>\E</td>
<td>Echo</td>
<td></td>
</tr>
<tr>
<td>\S</td>
<td>Read on-line status</td>
<td></td>
</tr>
</tbody>
</table>

Writing scripts:
You can combine several modem-speak commands to write scripts. The one I frequently use is:

```
AT&FS0=1&C1&D3&K3&Q9&W
```

Let’s break it down and see what it really does…
AT&F  load factory defaults and settings
S0=1  set modem to answer on first ring
&C1&D3 set modem up for “action” (cd/dtr)
&K3  set hardware flow control
&Q9  set compression
&W  save configuration to modem

During the course of using modems there are several other “abbreviations” you should also be familiar with. You will see these when using modems with routers and using the “debug” commands:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TxD</td>
<td>transmit data</td>
</tr>
<tr>
<td>RxD</td>
<td>receive data</td>
</tr>
<tr>
<td>RTS</td>
<td>request to send</td>
</tr>
<tr>
<td>CTS</td>
<td>clear to send</td>
</tr>
<tr>
<td>DSR</td>
<td>Data Set Ready</td>
</tr>
<tr>
<td>GRD</td>
<td>Signal ground</td>
</tr>
<tr>
<td>CD</td>
<td>Carrier detect</td>
</tr>
<tr>
<td>DTR</td>
<td>Data terminal ready</td>
</tr>
<tr>
<td>HS</td>
<td>High-speed</td>
</tr>
<tr>
<td>RI</td>
<td>Ring Indicate</td>
</tr>
<tr>
<td>CD</td>
<td>Carrier Detect</td>
</tr>
<tr>
<td>OH</td>
<td>Off Hook</td>
</tr>
<tr>
<td>RD</td>
<td>Receive Data</td>
</tr>
<tr>
<td>SD</td>
<td>Send Data</td>
</tr>
<tr>
<td>TR</td>
<td>Terminal Ready</td>
</tr>
<tr>
<td>MR</td>
<td>Modem Ready</td>
</tr>
</tbody>
</table>

Lights when communicating at more than 4800kbps
Blinks on and off when detecting incoming ring
Lights when the DCD signal from the fax modem to the computer is on
Lights when the fax modem is off hook
Light flashes when data is sent from the fax modem to your computer or other serial device. At high speeds the light may appear to be always “on.”
Flashes whenever data or commands are transmitted from the serial port of your computer or other device to the fax modem.
Lights when the computer is ready to send or receive data. Indicates the status of the DTR signal from the terminal or computer.
Lights when the fax modem is turned on. Flashes during self-test.

Above information from “Hayes Installation Guide” (2000).

Step-By-Step Instructions:
1. Set up the lab and cable it as shown.
2. Have each computer, one at a time, establish DUN between each other. Be sure to watch the indicator lights on the modem. Try to record the order during a call establishment and termination.
3. Try calling from one phone to another.
4. Try calling from one phone into another computer. As it tries to go you will hear negotiation taking place (Screech! Squak! Scratch!)
Supplemental or Challenge Activities:
1. Go out to CISCO and download the AT command set.
2. Try writing different scripts for your modem.
   a. Write one to limit line speed to 9600 bps.
   b. Write another to answer on the second ring.
   c. Write one to show default settings during the boot.
3. Try using a protocol inspector to “see” the negotiation between two PC’s using DUN. Change the settings for protocols and stuff.

So What Have I Learned Here?
Ok…so this is a bit more in-depth than you really need to get with modems. If you continue on with your CISCO training then you will have to be very familiar with these commands. We use these to set up the ability to dial into a router and make changes. Again, this is one of those labs where I grew up doing this but newer people to the field have not had the need for anything like this…call it a catch-up if you want.

This space for rent.

Send an email today to Basham.Matt@spcollege.edu to find out how your company can advertise in this space!

Hitting a million targeted readers provides good value for your advertising dollars.
1b. Workstation Foundations: Knoppix STD

They said “I was crazy!”

How do you like the idea of free and low-cost textbooks? Well here is how you can help me in the effort to bring more of these types of books to you. They say there is safety in numbers, well I can promote this type of textbook production way better if I hear from you personally. Send me an email at Basham.Matt@spcollege.edu just to tell me you are using the book, you like the idea of cheaper high-quality books, or to just say “hello.” Together we will change education for the better!
Just what the heck is Knoppix STD anyway?

Knoppix is a slimmed-down version of Linux that has incorporated a bunch of security tools (hence the comedic name STD: Security Tools Distribution). Linux started about 15 years or so ago although it doesn’t seem like that long ago. Linus Benedict Torvalds was not too keen on the free operating systems available in the early 1990’s so he wrote one himself: Linux.

This operating system was designed with a much better stock “kernel” (think of it like the brain of the operating system) to place other applications and programs on top of (think of those like the senses and extremities) than the existing ones of the day. But, unlike other operating systems he decided to only put into the operating system the bare minimum requirements. Users could then later customize their systems to contain exactly what they needed. This made for a more stable operating system and less, or even no, “blue screens of death.” Later on an even better advantage surfaced: Linux was immune to the viruses of the day.

From this kernel spawned many versions, also known as “flavors”, of the Linux operating system: Red Hat, Caldera, Mandrake and others. Each one with different abilities and functions. Recently a new flavor has been added called Knoppix and its more customized version “STD.” A difference here is that STD can be self-contained on a CD-rom and placed into any machine, rebooted, and the Knoppix/STD operating system comes up on the machine. When the machine is rebooted to the hard drive the next time the regular operating system comes back. No harm, no foul.

To use Knoppix STD you need at least a 486 CPU with a minimum of 20 MB of RAM. You can probably find these everywhere in garage sales and thrift shops. I do, however, recommend using 128 MB of RAM to get the full range of tools in graphic mode. To load Knoppix STD first switch your BIOS settings to boot from CD. Don’t know how to do this? Ask your instructor or go look it up in your books or on the web. Get used to not knowing how to do things in the computer world…you just have to take one step at a time and do not get discouraged. There are other ways to boot up Knoppix like using a floppy disk/CD combination. Look out on the web for how to do that. Once you get Knoppix up you should see the desktop (and maybe an internet window):
Yeah, I know, it really, really looks similar to Windows, but believe me it does not work like Windows. Along the bottom you will see a taskbar with a bunch of options:

Now let’s look at each of programs and menus separately.

This represents the “Start Applications.” It is similar to the “Start” button in Windows. Why a K? No, not directly to represent Knoppix but the KDE or “Kool Desktop Environment.” Clicking on this gives you the bulk of the stuff in Knoppix. We’ll come back to that in a bit.

This icon represents “window list.” It does just what it sounds like…it gives you a listing of all open windows in the KDE.

This will show you the desktop…which you can already see. Use it later to toggle back in forth between windows.

This is the “home icon.” Clicking on it will bring up file://home/knoppix Then you will be seeing the home directory, the desktop, GNUstep, a folder for the tmp directory and some other stuff. Not much to worry about right now.

This icon will bring up the KDE control center. From here you can make all your settings and stuff. Think of this as being similar to the control panel in Windows.

This is the Konsole Shell. Think of this as similar to the DOS prompt in Windows. From here you will do several labs with Knoppix STD.

This icon represents the Konqueror web browser. (Explorer-ish)

This icon represents the Mozilla web browser. (Netscape-ish)

This icon is used when you want to block all inbound TCP packets. Don’t worry about using this right now…that comes later.
The first thing you should do is click on the “Start Applications” button and just explore:

Lot’s of good stuff here to explore and play with. Later I will point out some more useful stuff for you to use in Knoppix STD during your Cisco studies. Notice the little “help” icon that looks like a life buoy. Hmmm…you might want to look at that.

Good reading (I hope the links stay good for you)
http://www.knoppix.net (you can download Knoppix here too)
or
http://www.knoppix.net/docs/index.php/KnoppixForNewbies

From the above document:
“With Knoppix you have an ideal introductory package which includes the most of the best known applications (in Linux)...Knoppix is also ideal for other reasons. Trying out Knoppix is a zero investment and zero risk proposition: you do not need any dedicated computers and you cannot no matter how hard you try, crash your computer or lose data by playing around with Knoppix.” (p.1)
**Objective:**
This lab is designed to help you become familiar with various commands in the Knoppix Konsole screen.

**Tools and Materials:**
Windows-based computer
Knoppix STD CD

**Step-by-Step Instructions:**
1. Re-boot the machine into the Knoppix environment from the CD. Boot to a user mode, not the root mode.
2. Open a Konsole session. Along the bottom taskbar you will see a picture of a monitor with a yellow shell superimposed on it.

   ![Konsole Shell Icon]

Click on the Konsole Shell icon.

When the window opens you will see a prompt that says `knoppix@` and the hostname of your workstation with a colon, tilde and a dollar sign:

```
knoppix@star10616121:~$
```

(Your prompt will probably vary and may include the word “root”) On the next page I put a picture of this screen but, for sake of keeping this document small I will only show the text and not the screen whenever necessary.
There is a command called `host` that you can use to find out a bunch of information. Let's start off with looking at the help for host:

```
knoppix@star10616121:~$ host
Usage: host [-v][-a][-t querytype][options] name [server]
Listing:host[-v][-a][-t querytype][options]-l zone [server]
Check soa:  host [-v] [options] -C zone
Addrcheck: host [-v] [options] -A host
Listing options:[-L level][-S][-A][-p][-P prefserver]
[-N skipzone]
Common options:[-d][-f|-F file][-I chars][-i|-n][-q][-Q]
[-T] [-2]
Other options: [-c class] [-e] [-m] [-o] [-r] [-R] [-s]
       secs] [-u] [-w]
Special options: [-O srcaddr] [-j minport] [-J maxport]
Extended usage:  [-x [name ...]] [-X server [name ...]]
```

You can verify your hostname with the command `hostname`.

```
knoppix@star10616121:~$ hostname
star10616121
```

3. Next let's look at the tcp/ip settings on your workstation. As you will see we have considerably more commands and options in Knoppix than we did under Windows and DOS (remember using ipconfig and winipcfg from the command prompt?). First let's just for the heck of it type in `ip`:

```
knoppix@star10616121:~$ ip
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
where   OBJECT := { link | addr | route | rule | neigh |
tunnel | maddr | mroute | monitor }
OPTIONS :={ -V[ersion] | -s[tatistics] | -r[esolve]|
      -f[amily] { inet | inet6 | ipx | dnet | link } | -o[nneline] }
```
So, let’s go ahead and pick the object equal to `addr` for address information:

```
knoppix@star10616121:~$ ip addr
1: lo: [LOOPBACK,UP] mtu 16436 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
2: eth0: [BROADCAST,MULTICAST,UP] mtu 1500 qdisc pfifo_fast
   link/ether 00:c0:4f:14:39:04 brd ff:ff:ff:ff:ff:ff
   inet 192.168.151.60/24 brd 192.168.151.255 scope global
```

Keep in mind I highlighted some stuff to illustrate the step. In the above output we can see the ip address of the workstation I used is 192.168.151.60 with a subnet mask of 255.255.255.0 (aka “/24”). But, unlike in Windows/DOS, we also have the MAC address of `00:c0:4f:14:39:04` and a broadcast address of `ff:ff:ff:ff:ff:ff`. But wait! We are seeing that things may look the same but we have differences. We can use another command that is similar to ipconfig in DOS. In Knoppix we can use the `ifconfig` command.

```
knoppix@star10616121:~$ ifconfig
eth0
```

4. Ok, so now we know who we are let’s find out what is our path to the Internet by using the `traceroute` command. This is similar to the tracert command you used in DOS and the same command you will be using on Cisco routers later. But first, let’s see what our options are for `traceroute`:

```
koppix@star10616121:~$ traceroute
Version 1.4a12
Usage: traceroute [-DFIlonrvx] [-g gateway] [-i iface] [-f first_ttl][-m max_ttl] [ -p port] [-q
nqueries] [-s src_addr] [-t tos] [-w waittime] [-z pausemsecs] host
```
Just a few more options here than we had in DOS. For example, if we had more than one NIC we could specify from which interface (iface) we wanted to start our trace route. Let’s take a guess, based upon our ip address that we already found, that the gateway for ip address 192.168.151.60 is 192.168.151.1 and doublecheck it with traceroute:

```
knoppix@star10616121:~$ traceroute 192.168.151.1
traceroute to 192.168.151.1 (192.168.151.1), 30 hops max, 38 byte packets
1 192.168.151.1 (192.168.151.1) 0.905 ms 0.863 ms 0.841 ms
```

From this we can see our default trace route packet size is 38 bytes and our gateway is only one stop away. We can also see trace route will only work up to 30 hops maximum. We have good evidence that 192.168.151.1 is our gateway but we can really be sure by using traceroute to hit something on the Internet, like yahoo.com (I took out some spaces to make this read better):

```
knoppix@star10616121:~$ traceroute www.yahoo.com
trace route Warning: www.yahoo.com has multiple addresses; using 216.109.118.76
trace route to www.yahoo.akadns.net (216.109.118.76), 30 hops max, 38 byte packets
1 192.168.151.1(192.168.151.1)0.952 ms 0.913 ms 0.918 ms
2 192.168.154.1(192.168.154.1)4.197 ms 3.984 ms 3.982 ms
3 do-ess5000 (172.23.1.1)4.039 ms 3.911 ms 3.892 ms
4 192.168.100.27(192.168.100.27)4.787 ms 4.564 ms4.542 ms
5 192.168.255.4(192.168.255.4)5.644 ms 5.261 ms 5.173 ms
6 66.194.104.14(66.194.104.14)5.797 ms 6.124 ms 6.019 ms
7 64-132-156-189.gen.twtelecom.net(64.132.156.189)8.253 ms 7.829 ms 8.340 ms
8 64-132-156-225.gen.twtelecom.net(64.132.156.225)
23.822 ms 12.032 ms 11.627 ms
9 66.192.243.224(66.192.243.224)8.064 ms 8.107 ms7.885 ms
10 dist-02-ge-2-3-0-0.tamq.twtelecom.net (66.192.243.102)
35.732 ms 36.434 ms 36.207 ms
11 dist-01-so-0-0-0-0.mtld.twtelecom.net (66.192.243.6)
23.822 ms 12.032 ms 11.627 ms
12 dist-02-ge-3-3-0-0.tmt.telecom.net (66.192.243.130)
11.597 ms 11.604 ms 14.049 ms
14 core-01-so-0-0-0-0.asbn.twtelecom.net (66.192.255.27)
35.953 ms 38.449 ms 36.380 ms
15 66.192.255.229(66.192.255.229)110.49ms35.55ms35.515 ms
16 g2-12-bas2.dce.yahoo.com (106.223.115.2) 35.732 ms
36.434 ms 36.207 ms
17 vlan201-msr1.dcn.yahoo.com(216.115.96.163)37.052 ms
36.930 ms 36.045 ms
18 v130.bas1-m.dcn.yahoo.com (216.109.120.142) 36.136 ms
v147.bas1-m.dcn.yahoo.com (216.109.120.218) 39.633 ms
v130.bas1-m.dcn.yahoo.com (216.109.120.142) 37.084 ms
19 p13.www.dcn.yahoo.com(216.109.118.76)37.740ms 36.807 ms
38.488 ms
Here we can see some distinctive differences between DOS tracer and Knoppix traceroute. First and foremost Knoppix is way quicker than DOS. This is because it does not have to wade through tons of stuff on its way out of the machine and ditto on the return trip. Next, if you are comparing this to the DOS output earlier, you will also see that most of the packets did not timeout and we actually got the information back on the actual routes used. With knowledge comes power (Scientia es gravis), but that will be covered a bit more in the Script Kiddie Cookbook and what to do with this information. So, once again, we have found really good evidence that our gateway is 192.168.151.1 but you would think that Knoppix would have an iron-clad way of letting us know this information, right? Right! I was just stalling a bit. We have a command for address resolution that will not only tell us the gateway, but the MAC address of it as well:

```
knoppix@star10616121:~$ arp
Address        HWtype  HWaddress         Flags Mask  Iface
192.168.151.1  ether   08:00:02:1D:FC:B7   C         eth0
```

Ok, so now we have found out a bunch of stuff about our own ip address, MAC address, gateway ip address, and gateway MAC address. Let’s turn next to some other icmp-related commands. One last tool for us is to “see” a routing table for the workstation. Yeah, I said it…a routing table for the workstation. Here we can the same information only this time we see our subnet mask of 24 bits too:

```
knoppix@star10616121:~$ route
Kernel IP routing table
Destination    Gateway Genmask       Flags Metric Ref  Use Iface
192.168.151.0   *      255.255.255.0   U    0   0   0  eth0
default      192.168.151.1 0.0.0.0     UG   0   0   0  eth0
```

5. Let’s first look at the icmp implementation, better known as ping and its options in Knoppix (notice how Knoppix uses 64 bytes instead of 32 in DOS):

```
koppix@star10616121:~$ ping
usage: ping [-LRdfnrqv] [-c count] [-i wait] [-l preload]
          [-p pattern] [-s packetsize] [-t ttl] [-I interface
       address] host

knoppix@star10616121:~$ ping www.spcollege.edu
PING www.spcollege.edu (172.16.1.68): 56 data bytes
64 bytes from 172.16.1.68: icmp_seq=0 ttl=124 time=5.4 ms
64 bytes from 172.16.1.68: icmp_seq=1 ttl=124 time=5.6 ms
64 bytes from 172.16.1.68: icmp_seq=2 ttl=124 time=5.3 ms
--- www.spcollege.edu ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 5.3/5.4/5.6 ms
```

92
You will have to use the break sequence (Control+C) to stop the icmp on-slaughter. By default Knoppix uses a continuous ping (so does Linux). In DOS you had to add the \texttt{-t} option to do that. You can see here I only let it run for three packets. This too is much quicker than in DOS under Windows. Take a second and play with those options if you wish.

6. We have seen many similarities between Knoppix and the DOS commands that you used under Windows. One distinct difference that you will find in Knoppix and not in DOS in the ability to get help or similar commands related to other commands. Let’s burn up a few pages on getting “manuals” in Knoppix for commands. First, in case you cannot remember the specific Knoppix command you can just rely on the old DOS standby and ask for help:

```
knoppix@star10616121:~$ help arp
bash: help: no help topics match `arp'. Try `help help' or `man -k arp' or `info arp'.
```

So I went ahead and did the \texttt{man \textasciitilde-k arp} command just like it said to (aren’t computers great...they don’t do anything you don’t tell them too...see also the “Hacker Manifesto” for more details)

```
knoppix@star10616121:~$ man \textasciitilde-k arp
arp (7) - Linux ARP kernel module.
arp (8) - manipulate the system ARP cache
arp2ethers (8) - convert arpwatch address database to ethers file format
arpfetch (8) - obtain ethernet/ip address pairings via snmp
arpcount (8) - keep track of ethernet/ip address pairings
arpspoof (8) - intercept packets on a switched LAN
arpwatch (8) - keep track of ethernet/ip address pairings
dmassagevendor (8) - convert the ethernet vendor codes master list to arpwatch format
in.telnetd (8) - DARPA telnet protocol server
massagevendor (8) - convert the ethernet vendor codes master list to arpwatch format
nemesis-arp (1) - ARP/RARP Protocol (The Nemesis Project)
portmap (8) - DARPA port to RPC program number mapper
rarp (8) - manipulate the system RARP table
sane-sharp (5) - SANE backend for SHARP scanners
telnetd (8) - DARPA telnet protocol server
XWarpPointer (3x) - move pointer
Carp::Clan (3pm) - Report errors from perspective of caller of a "clan" of modules
```

Ok, so it is not the greatest but it is better than the DOS stuff. We can see some \texttt{arp}-related commands that can be used and a brief description. Unfortunately most of these \texttt{arp}-related commands don’t work with Knoppix.
Let’s move on to looking at another manual for a command:

```
knoppix@star10616121:~$ man -k ifconfig
ifconfig (8)         - configure a network interface
```

Nice. Simple and to the point. Not much confusion there. Another one:

```
knoppix@star10616121:~$ man -k traceroute
paratrace (1)- Parasitic Traceroute via Established TCP Flows & IPID Hopcount
traceroute (8) - print the route packets take to network host
traceroute.lbl(8)-print the route packets take to network host
```

```
knoppix@star10616121:~$ man -k ping
con2fbmap (8)        - hows and set mapping between consoles and framebuffer devices.
devdump (8)          - Utility programs for dumping and verifying iso9660 images.
getkeycodes (8)      - print kernel scancode-to-keycode mapping table
hping (8)            - send (almost) arbitrary TCP/IP packets to network hosts
hping2 (8)           - send (almost) arbitrary TCP/IP packets to network hosts
irdaping (8)         - sends IrDA test frames
isodump (8)          - Utility programs for dumping and verifying iso9660 images.
isinfo (8)           - Utility programs for dumping and verifying iso9660 images.
isovfy (8)           - Utility programs for dumping and verifying iso9660 images.
mysqldump (1)        - text-based client for dumping or backing up mysql databases, tables and data.
ping (8)             - send ICMP ECHO_REQUEST packets to network hosts
setkeycodes (8)      - load kernel scancode-to-keycode mapping table entries
swapoff (2)          - start/stop swapping to file/device
swapoff (8)          - enable/disable devices and files for paging and swapping
swapon (2)           - start/stop swapping to file/device
swapon (8)           - enable/disable devices and files for paging and swapping
TIFFReverseBits (3t) - byte- and bit-swapping routines
TIFFSwab (3t)        - byte- and bit-swapping routines
TIFFSwabArrayOfLong(3t)-byte- and bit-swapping routines
TIFFSwabArrayOfShort(3t)- byte- and bit-swapping routines
TIFFSwabLong (3t)    - byte- and bit-swapping routines
TIFFSwabShort (3t)   - byte- and bit-swapping routines
wcsspn (3)           - advance in a wide-character string, skipping any of a set of wide characters
wctrans (3)          - wide character translation mapping
xfs_bmap (8)         - print block mapping for an XFS file
g1DDepthRange (3x)   - specify mapping of depth values from
```
Let’s finish off this lab with some DHCP with Knoppix. You will need to know how to do this if you keep using Knoppix with the Cisco labs. First, open up a Konsole Shell session and look at your current settings:
Now in DOS we did the \texttt{ipconfig /release} and \texttt{ipconfig /renew} commands to release and renew dynamic ip addresses. In Knoppix we just need to shut down the interface and bring it back up to do the same thing (you may have to change to superuser first before bringing it down):

```
Knoppix@star10616121:~$ ifconfig eth0 down
SIOCSIFFLAGS: Permission denied
Knoppix@star10616121:~$ su
Password:
Knoppix@star10616121:~$ ifconfig eth0 down
Knoppix@star10616121:~$ ifconfig
```

```
lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      UP LOOPBACK RUNNING MTU:16436 Metric:1
      RX packets:123769 errors: dropped:0 overruns:0 frame:0
      TX packets:123769 errors: dropped:0 overruns:0 carrier:0
      collisions:0
      RX bytes:34349148 (32.7 MiB)  TX bytes:34349148  (32.7 MiB)
```

Notice how the \texttt{eth0} interface has disappeared from our \texttt{ifconfig} output. Now that the ip address is gone we need to statically add one in (or if you want to get technical we could revert to dynamic again, but what is the point at this time?). You will also notice that even though we did not put in the broadcast address Knoppix figured it out for us from our netmask and ip address.

```
Knoppix@star10616121:~$ ifconfig eth0 192.168.151.169 netmask 255.255.255.0
Knoppix@star10616121:~$ ifconfig
Eth0    Link encap:Ethernet  Hwaddr  00:C0:4F:14:39:04
       Mask:255.255.255.0
       UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
       RX packets:73817 errors: dropped:0 overruns:0 frame:0
       TX packets:25193 errors: dropped:0 overruns:0 carrier:0
       collisions:0
       RX bytes:9538284 (9.0 MiB)  TX bytes:21162759  (20.1 MiB)
```

```
lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      UP LOOPBACK RUNNING MTU:16436 Metric:1
      RX packets:123769 errors: dropped:0 overruns:0 frame:0
      TX packets:123769 errors: dropped:0 overruns:0 carrier:0
      collisions:0
      RX bytes:34349148 (32.7 MiB)  TX bytes:34349148  (32.7 MiB)
```

Knoppix@star10616121:~$
Now, to get our ip address back to automatic addressing we just need to take the interface down again (which essentially wipes out the static address for the moment), and re-enable dhcp:

    Knoppix@star10616121:~$ ifdown eth0
    Knoppix@star10616121:~$ ifup eth0

I know it looks silly but it makes sense in Knoppix language. This all refers to a file you can look at in /etc/network/interfaces (notice it is plural)...if you want to look at that file the easiest way (and most reliable) is to type this at the Konsole prompt:

    Knoppix@star10616121:~$ kwrite /etc/network/interfaces

This will bring up the default script file for your interface configuration. It is this one that is used for your dhcp and other settings when the machine boots up.

```plaintext
# /etc/network/interfaces -- configuration file for ifup(8),
ifdown(8)

# The loopback interface
# automatically added when upgrading
auto lo eth0
iface lo inet loopback

iface eth0 inet dhcp
```

The pound sign (#) in front of the lines is ignored by the computer and just signifies a “remark” or “comment” that is inserted by the programmer for ease of reading and understanding later. You get a lot more out of this file than you would if this was the file:

```plaintext
auto lo eth0
iface lo inet loopback

iface eth0 inet dhcp
```

So, these little comments are great. We can double check to see if our dhcp is working correctly by going back to the Konsole shell and typing `ifconfig` again and see if it went back to the original ip address of 192.168.151.68. Sometimes that ip address may have been re-administered to another machine that requested a dhcp address during its period of non-use so if it doesn’t come back exactly then you know why. Most of the times you should get it back though.
Knoppix@star10616121:~$ ifconfig
Eth0 Link encap:Ethernet  Hwaddr  00:C0:4F:14:39:04
    Mask:255.255.255.0
    UP  BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
    RX packets:73817 errors: dropped:0 overruns:0 frame:0
    TX packets:25193 errors: dropped:0 overruns:0 carrier:0
    collisions:0
    RX bytes:9538284 (9.0 MiB)  TX bytes:21162759  (20.1 MiB)

lo  Link encap:Local Loopback
    inet addr:127.0.0.1  Mask:255.0.0.0
    UP LOOPBACK RUNNING  MTU:16436  Metric:1
    RX packets:123769 errors: dropped:0 overruns:0 frame:0
    TX packets:123769 errors: dropped:0 overruns:0 carrier:0
    collisions:0
    RX bytes:34349148 (32.7 MiB)  TX bytes:34349148  (32.7 MiB)

Knoppix@star10616121:~$

**Are you enjoying the materials?** Well be on the lookout for some other manuals and textbooks on [http://www.lulu.com/learningbydoing](http://www.lulu.com/learningbydoing) and [http://www.spcollege.edu/star/cisco](http://www.spcollege.edu/star/cisco)


This book title is based upon the Anarchist Cookbook that came out in the 1980’s. In that tremendously popular book it showed you how to build bombs at home. Similarly the Script Kiddie Cookbook will show you step by step hacking and hacking methodology. You will go to jail if you use this information improperly!!

Learn the basics about computer security with this book. This book assumes you know nothing about security and starts there. All tools and software used are freeware. Other books will build upon the materials in this book. Learn about the nuts and bolts of SPAM, port scanning, Knoppix tools, cookies, and bring it all together with exercises in hacking at legal gaming sites.

**GET THE BOOK THE GOVERNMENT DOES NOT WANT YOU TO OWN!!!**
**Objective:**
This lab is designed to help you become familiar with various commands in the Knoppix Konsole screen for using FTP and TELNET.

**Tools and Materials:**
Windows-based computer
Knoppix STD CD

**Step-by-Step Instructions:**
11. Re-boot the machine into the Knoppix environment from the CD. Boot to a user mode, not the root mode.
12. Open a Konsole session. Along the bottom taskbar you will see a picture of a monitor with a yellow shell superimposed on it.

Click on the Konsole Shell icon.

When the window opens you will see a prompt that says knoppix@ and the hostname of your workstation with a colon, tilde and a dollar sign:

knoppix@star10616121:~$
Let’s start off by exploring FTP. To get FTP started just type ftp at the Konsole Shell prompt:

```
koppix@star10616121:~$ ftp
```

```
lftp :~>
get [OPTS] <rfile> [-o <lfile>] help [<cmd>]
```

Ok…now that we have seen an overview of the commands we are ready to go. Type quit to get back to the prompt and then type in `ftp ftp1.ipswitch.com` to open the ftp session and to connect to ipswitch.com. I am going to have you download a .pdf file.

```
koppix@star10616112:~$ ftp ftp1.ipswitch.com
```

Then see what directories are available to you.

```
lftp ftp1.ipswitch.com:~$ dir
```

```
-r-x------  1 anonymou System          322 Mar 10  2000 win3.qif
-r-x------  1 anonymou System          758 Mar 13  2000 win95.qif
-r-x------  1 anonymou System          758 Mar 10  2000 winNT.qif
```

```
```

100
Let's switch to the ipswitch folder first and look at what's there:

```
1ftp ftp1.ipswitch.com:/> cd ipswitch
cd ok, cwd=/ipswitch
1ftp ftp1.ipswitch.com:/ipswitch> dir
dr-x------ 2 anonymou System 0 Sep 3 13:59 .
dr-x------ 2 anonymou System 0 Sep 3 13:59 ..
dr-x------ 2 anonymou System 0 May 13 2003 Acrobat
dr-x------ 2 anonymou System 0 Sep 24 18:52 International_Downloads
dr-x------ 2 anonymou System 0 Sep 8 2001 Manuals
dr-x------ 2 anonymou System 0 Jul 8 15:14 Product_Downloads
```

We are almost there. Now switch to the manuals folder/subdirectory and see if the file wug7.pdf is in there.

```
lftp ftp1.ipswitch.com:/ipswitch> cd manuals
cd ok, cwd=/ipswitch/manuals
1ftp ftp1.ipswitch.com:/ipswitch/manuals> dir
dr-x------ 2 anonymou System 0 Sep 8 2001 .
dr-x------ 2 anonymou System 0 Sep 8 2001 ..
-r-x------ 1 anonymou System 848621 May 25 16:50 ftpserv.pdf
-r-x------ 1 anonymou System 440885 Oct 16 2000 ftpserv1.pdf
-r-x------ 1 anonymou System 793740 Oct 16 2000 ftpserv2.pdf
-r-x------ 1 anonymou System 698318 Nov 29 2001 ftpserv3.pdf
-r-x------ 1 anonymou System 757484 Jun 11 2002 ftpserv31.pdf
-r-x------ 1 anonymou System 720416 Apr 30 2003 ftpserv4.pdf
-r-x------ 1 anonymou System 1241555 Mar 10 2000 Imail1406.pdf
-r-x------ 1 anonymou System 2018046 Mar 10 2000 imail15.pdf
-r-x------ 1 anonymou System 2150082 Mar 10 2000 imail16.pdf
-r-x------ 1 anonymou System 2879228 May 29 2002 imail17.pdf
-r-x------ 1 anonymou System 2879228 May 29 2002 Imail71.pdf
-r-x------ 1 anonymou System 254383 Jun 27 2001 imail17gs.pdf
-r-x------ 1 anonymou System 3568782 Oct 6 15:14 imail18.pdf
-r-x------ 1 anonymou System 3569956 Jun 27 2003 imail1801.pdf
-r-x------ 1 anonymou System 3458130 Mar 5 2003 imail1802.pdf
-r-x------ 1 anonymou System 2604235 Mar 31 07:32 imail1810.pdf
-r-x------ 1 anonymou System 305388 Mar 31 07:32 Imail1810gs.pdf
-r-x------ 1 anonymou System 168375 Nov 27 2001 imailav1.pdf
-r-x------ 1 anonymou System 386643 May 5 2003 imailexpressgs8.pdf
-r-x------ 1 anonymou System 288178 Mar 5 2003 imai1gs8.pdf
-r-x------ 1 anonymou System 358217 Mar 10 2000 imopts.pdf
-r-x------ 1 anonymou System 174413 Mar 10 2000 os2faq.zip
-r-x------ 1 anonymou System 184164 Mar 10 2000 Tnhost.pdf
-r-x------ 1 anonymou System 220177 Mar 10 2000 vt320.pdf
-r-x------ 1 anonymou System 423638 Mar 10 2000 vt320.ps
-r-x------ 1 anonymou System 472372 Mar 10 2000 Vt320w32.pdf
-r-x------ 1 anonymou System 400966 Mar 10 2000 whatsup.pdf
-r-x------ 1 anonymou System 2607864 Dec 20 2002 whatsupg.pdf
-r-x------ 1 anonymou System 262820 Mar 10 2000 wsd1150.pdf
-r-x------ 1 anonymou System 1107941 Sep 26 2002 wsftp.pdf
-r-x------ 1 anonymou System 456257 Mar 10 2000 wsftp50.pdf
-r-x------ 1 anonymou System 855225 Mar 10 2000 wsftp60.pdf
```
Trar she blows! Now all that is left is to go out and get the file.

```
lftp ftp1.ipswitch.com:/ipswitch/manuals> get wug7.pdf
```

During the transfer you will see percentages flash across the screen.

```
knoppix@star10616112:$ ftp ftp1.ipswitch.com
'wug7.pdf' at 1078940 (48%) 130.3K/s eta:9s [Receiving data]
```

When the file is done transferring you just have to go out and find it using the locate current directory command:

```
2215303 bytes transferred in 25 seconds (86.5K/s)
lftp ftp1.ipswitch.com:/ipswitch/manuals> lcd
lcd ok, local cwd=/home/knoppix
```

Now you can go and see the file just by using dir at the prompt. You should be seeing that it is not really that much different than you had with DOS under Windows, except that the Knoppix STD works better, quicker, and has more options.

---

Some of the best books are brought to you by Lulu!

[http://www.lulu.com/about/](http://www.lulu.com/about/)
We have seen the basics of ftp with Knoppix STD so let’s turn to using telnet with Knoppix STD. Let’s start like we have been doing all along, looking for our help commands.

```
telnet> help
Commands may be abbreviated. Commands are:

  close     close current connection
  logout    forcibly logout remote user and close the connection
  display   display operating parameters
  mode      try to enter line or character mode ('mode ?' for more)
  open      connect to a site
  quit      exit telnet
  send      transmit special characters ('send ?' for more)
  set       set operating parameters ('set ?' for more)
  unset     unset operating parameters ('unset ?' for more)
  status    print status information
  toggle    toggle operating parameters ('toggle ?' for more)
  slc       set treatment of special characters
  auth      turn on (off) authentication ('auth ?' for more)
  z         suspend telnet
  environ   change environment variables ('environ ?' for more)
telnet>
```

Let’s jump in and telnet to the library of congress site like we did in the DOS labs.

```bash
knoppix@star10616121:-$ telnet
telnet> open locis.loc.gov
Trying 140.147.254.3...  
Connected to locis.loc.gov.  
Escape character is '^]'.

L O C I S:  LIBRARY OF CONGRESS INFORMATION SYSTEM

  To make a choice: type a number, then press ENTER

1   Copyright Information    -- files available and up-to-date
2   Braille and Audio        -- files frozen mid-August 1999
3   Federal Legislation      -- files frozen December 1998
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *

The LC Catalog Files are available at:
  http://lcweb.loc.gov/catalog/
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *

8   Searching Hours and Basic Search Commands
9   Library of Congress General Information
10  Library of Congress Fast Facts
12  Comments and Logoff
      Choice:
```
LOCIS: LIBRARY OF CONGRESS INFORMATION SYSTEM

To make a choice: type a number, then press ENTER

1 Copyright Information -- files available and up-to-date
2 Braille and Audio -- files frozen mid-August 1999
3 Federal Legislation -- files frozen December 1998

The LC Catalog Files are available at:
http://lcweb.loc.gov/catalog/

8 Searching Hours and Basic Search Commands
9 Library of Congress General Information
10 Library of Congress Fast Facts
12 Comments and Logoff

Choice:

After looking around I decided to go out to our gaming site at entcom.com:

knoppix@star10616121:~$ telnet
telnet> open entconn.com
Trying 66.74.22.52…
Escape character is '^]'
Auto-sensing…
?L

Welcome

Welcome, newcomer! You have logged on to the world's most advanced multi-user online system: Worldgroup.

Before going into that, though, let's get acquainted. If you'll tell us a little bit about yourself, we'll create an account for you that you can use anytime for "free samples" of what we have to offer. There will be certain things you won't be able to do (like posting messages, etc.) until you elect to become a full Member of the Board… but you will have a chance to see if you like us first.

The following word may or may not be blinking: ANSI
Is it blinking (Y/N)? y

Good! Your answer has been used to control the ANSI features of this system. Now if you'll tell us a little about yourself, we'll get underway.

Please enter your first and last name:
joshua brackens
Enter the first line of your address (your street address or P. O. Box):
123 anystreet

Enter the second line of your address (city, state, and ZIP):
Dumpsville, FL

Enter the last line of your address (Country or press ENTER for U.S.):

Now enter the telephone number where you can be reached during the day:
123-4567

Please enter your date of birth in "mm/dd/yyyy" format:
01/01/1960

Are you male, or female? m

Now, you need to choose a "User-ID" for yourself. Your User-ID will be your "code name" on this system. You will use it to identify yourself to the system when you log on, and other users will know you by this name. Your User-ID can be 3 to 29 letters long (including letters, spaces, commas, periods, hyphens, and digits). If you want it capitalized a certain way (for example, "John McLean"), type it in just that way. If you use all lower case, or ALL UPPER CASE, the system will apply its own capitalization standards.

Enter the User-ID you want to be known as: joshua

Here is a simulated message, showing how your User-ID will appear to other users:

  From Joshua: How does this run so many users at once?

Are you satisfied with your choice of User-ID (y/n)? y

Ok, Joshua, now you'll also need to select a password, so that you can keep other people from using your account without your permission. Make it short and memorable, but not obvious. The security of your account depends on nobody else knowing what your password is.

Enter the password you plan to use: *******

Keep track of updates and changes at http://www.spcollege.edu/star/cisco Scroll to the bottom of the page and click on the “Lab Manual Edits.”
Ok...so now you want to go out and play? Try finding some help and learning how to use TFTP. The help function is not the same...weird.

```
      knoppix@star10616121:~$ tftp
      tftp> help
      ?Invalid command
      tftp> ?
      Commands may be abbreviated. Commands are:

      connect         connect to remote tftp
      mode            set file transfer mode
      put             send file
      get             receive file
      quit            exit tftp
      verbose         toggle verbose mode
      trace           toggle packet tracing
      status          show current status
      binary          set mode to octet
      ascii           set mode to netascii
      rexmt           set per-packet retransmission timeout
      timeout         set total retransmission timeout
      ?               print help information
      tftp>
```

---

**St. Petersburg College**

**Earn your Bachelor’s Degree from St. Petersburg College**

In Technology Management

Transferring and having access to a 4-year degree has never been easier, more convenient, or cost effective.

**St. Petersburg College** allows students who have received an Associate degree (Associate in Arts or Associate in Science, depending on the program) to transfer into the Bachelor of Applied Science in Technology Management.

Opportunities Include:

- On-line and in traditional classroom courses
- Day and evening classes
- Full and part-time course schedules

Explore the opportunities waiting for you at St. Petersburg College!

For more information call: (727) 341-4772
E-mail: 4yeardegrees@spcollege.edu
Objectives:
Learn how to set up a router and login through a router console port from a workstation using the Knoppix STD Minicom program.

Tools and Materials:
Knoppix STD-booted workstation with Minicom program
CISCO router
(1) rollover cable (ro)

Background:
“Easy when you know how…” is very applicable when accessing a router through a workstation. This lab is designed to show you how to set up the Minicom communications program, to connect cabling and how to access the router. Another cool feature with Knoppix is there are some communication tools that will allow you to communicate outside the workstation through serial ports, modems, dsl lines, or other telephony equipment. Just be warned they can be pretty darned temperamental though. Here’s where we start getting you out of GUI’s and back into the command line. I tried at least 10 times to get Minicom to start using the GUI and it just kept not starting. Here is a shot showing how to get to Minicom by the menus:
Step-By-Step Instructions:

1. Setup and cable the workstation to the router as shown using a rollover cable. If you are not sure what a rollover cable is, then ask your instructor, a buddy, or skip ahead to the networking foundations section and look it up there.

2. Log into Knoppix as the root. You will not be able to make any changes to the Minicom program default settings unless you are at the root. If your instructor has already set up the defaults then you can skip ahead. If you are not logged in as the root then open a Konsole Shell and type `su` to switch to “super user” and put in the root password.

   Knoppix@star10616112:~$ su
   Password:
   root@star10616112:/home/knoppix#

3. Now you should be ready to get into Minicom. I would skip that whole GUI menu system. You could try using the “Run Command” on the “Start Applications” menu, but let’s just get you used to shell stuff:

   root@star10616112:/home/knoppix#minicom

4. You will see a warning flash up about not running Minicom from the root, but we don’t necessarily want to run it from root, we just want to change the settings. When its done you will see:

   Welcome to minicom 1.83.1

   OPTIONS: History Buffer, F-Key Macros, Search History Buffer, I18n

   Press CTRL-A Z for help on special keys
5. How nice! By pressing control and A at the same time, releasing them, and then pressing Z we can get some help on special keys.

6. Press O (not zero) to get into the configure Minicom menu. (Hint: instead of doing the CTRL+A, Z you could just use CTRL+A, O instead). You should see this popup in the Konsole window:

   ![Konsole Window Showing Configure Minicom Menu]

   Using your up and down arrows highlight the “Serial port setup” and hit enter.

7. Now we need to change our serial line settings. Make your settings look like the ones below: (use dev/ttyS0 for com 1 and dev/ttyS1 for com 2)

   ![Serial Port Setup in Minicom]

   If you are not logged in at the root you will not be allowed to change the serical device which, by default is set to com 3.
8. When you are finished hit enter and the previous menu should be still there (these things open on top of each other…windows, if you will):

![Image of configuration menu]

Change your option to “Save setup as df1” (default file 1) and press the enter key. You will see a pop-up message that the configuration has been saved. Then select exit and press enter.

9. We are now ready to go, right? Wrong! We have to first exit Minicom by closing the Konsole Shell window, opening a new Konsole Shell window, and then typing minicom to start it up again. When you restart it you should see your router waiting for your every command! It’s good to be the king!
**Objective:**
To learn how to use a protocol inspector in a simple network setting.

**Tools and Materials:**
Knoppix STD workstation with connections to the classroom and Internet

**Step-By-Step Instructions:**
1. Ok, so the picture is a bit much, but it does depict an accurate representation of how your classroom computer, more or less, gains access to the Internet. You computer will connect to a switch, or series of switches. These switches may or may not have an ip address associated with them. Next the switch(es) will connect to an interface on a router. This interface will have an ip address and a MAC address associated with it. Usually each router interface will be its own subnet. Therefore everything that connects to that subnet will be able to be seen by Ethernet, including the traffic coming and going to the subnet from outside the subnet (ie., internet traffic, DHCP requests and replies, broadcasts, etc)...you will learn more about the nuts and bolts of that stuff later. From the router packets go across a DTE serial cable into some sort of modem, CSU/DSU, TSU, or other type of telephony device. That device is the connection to the ISP. Remember all that icmp stuff with gateways, trace routes, and stuff? Yeah, starting to get the picture? Right now this lab will show you how to get started using the Ethereal protocol inspector included in the Knoppix STD operating system.

2. The easiest way to start Ethereal is not using the GUI (“Start Applications” > “Sniffers” > “Ethereal”), but to open a Konsole Shell and typing `ethereal` at the prompt.

Knoppix@star10616112:~$ `ethereal`
3. After about a second or two you will see the program come up in a separate window:

![Program Window]

4. To get this thing going you can click on Control+K or using the menus click on “Capture” and then “Start.”

![Additional Program Window]
5. Now after you click that you should see a pop up window like this:

![Ethereal Capture Options Window](image)

6. We just need to tell ethereal our interface. To double check you can use the Konsole Shell and type `ifconfig` to see how Knoppix STD refers to your interfaces.

```
Interface name
```

![Shell No. 2 - Konsole](image)
7. Now type in eth0 (e-t-h-zero) and click “ok.” You will see another popup window start counting packets.

8. You can have some fun by shooting some ping packets across your network or ping the whole darned subnet by using the broadcast address and you will generate a bunch of icmp packets. Remember for every request there usually is a reply so sending 4 pings results in 8 captured. Click on stop to stop capturing packets. After a few seconds the main window will pop up with all of those packets you just captured:
9. Notice how we have three frames within the window. The top one shows us basic over-all information about the packets captured. When we highlight on a packet we are asking Ethereal to show us the contents of that packet. The middle frame is more user friendly. It shows us block by block what we are looking at. The bottom frame shows us the hexadecimal composition of the actual packet. You can actually do some sorting too.

Clicking on one of the headers will sort it alphabetically or numerically.

**Hungry for more???

You are almost there for the *CompTIA Network+* and *Security+* Certifications too!

For more information:

[http://www.comptia.org/certification/default.aspx](http://www.comptia.org/certification/default.aspx)

*Building successful careers for students everywhere...*
Utilities in Knoppix STD

Etherape
Finally I wanted to through in some of the utilities you may want to play with in Knoppix STD. First let’s look at Etherape, a pseudo-network monitoring utility. It is another way to “see” packets of a sort over the network. Again, you have to be a super user or root to do this.

Knoppix@star10616112:~$ su
Password:
root@star10616112:/home/knoppix#
etherape

There really is not much to do to the defaults. Just fire it up and watch it go. Oh sure, you can go in and change the colors, protocols and stuff but that is just gravy. Right now, just know it is there and you can really make your workstation look impressive as a monitoring station with this little utility.
Khexedit
This utility is an actual hexadecimal editing utility. Let’s look at an example from an Ethereal captured packet. This packet was used to take down a DNS server. How do I know? It’s a secret that I explain in the Script Kiddie Cookbook. For now, know this packet has caused evil.

Now, let’s see what happens when I load a captured packet into KhexEdit:
You can see all kinds of similar stuff that we saw in Ethereal. The only difference is now we can change the packet. I took part of the damaging payload out by changing the repeating sequence \texttt{c0 c0} and changing it to \texttt{00 00}.

In any case you have another tool to begin exploring with as you begin your career in networking.

\begin{boxed_quote}
\textbf{Take college courses on a new schedule--yours.}

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education. Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

\url{http://e.spcollege.edu}

St. Petersburg College’s eCampus. Your time. Your place. Your future.
\end{boxed_quote}
Konqueror
The last utility I wanted to point out here is the Konqueror browser. Not just because we have a cool looking spinning gear in the place of a world being surrounded by the Microsoft emblem (in Internet Explorer) but because they have a toolbar set up with some quick hyperlinks to more useful security websites. When you first pull up the browser you will see this:

Be prepared though...this is not your ordinary browser system on an unstable Windows platform...you have all sorts of new stuff like being asked if you want to transmit cookies while visiting websites...another good topic for the security book. Click on the link for Linux Security you will first see a cookie message then the page:
I just really like the penguins...they taste like chicken you know. Have some fun and look around. The next hyperlink I want to point out is to the Internet Storm Center. Here you can see if there is any unusually high activity on any given port(s).
Below you can see the port history looks fairly consistent...no major problems.

But, if there was a spike in there it may be symptomatic of a virus problem or hacking incident on a global basis. Do some searching out here. Look for history charts during the I Love You, Nimda, Code Red, or other worm attacks. Boy, I am dangerously treading on those computer security book topics. I had better stop know and get you into the networking foundations stuff.
Electronic technology continues to infiltrate industry, electrostatic discharge threatens operations causing costly damage, injury and downtime. Pro-Cote Inc. can protect your work environment against destructive static shock with Valspar ESD Epoxy Flooring Systems.

Certified Conductive and Static Dissipative applications for every need:

- Facilities with highly sensitive electronic equipment
- Computer facilities
- Electronic manufacturing & assembly
- Facilities containing AGV lines
- Clean rooms
- Manufacturing/processing flammable or combustible materials
- Munitions plants

“On Site Certification”

Specialty Applications:

- Non-skid coatings
- Chemical resistant urethanes
- Epoxy coatings
- Moisture cured urethane
- Seamless epoxy floor systems
- Waterborne systems
- VOC compliant systems
- Concrete leveling
- Concrete resurfacing
- Control joint repair

The Management at Pro-Cote Inc. wants to put its expertise to work for you. With over 55 years of combined experience, our goal is to be the best specialty coatings applicator company, period.

Application Experience

- Industrial floor coatings
- Manufacturing production areas
- Distribution and warehouse floors
- Loading docks and ramps
- Safety zones and work areas
- Battery charging areas
- UPS rooms
- Parking garages

Let us help with your specification needs.

“Safety Solutions”

Pro-Cote Inc. is a proud installer of Valspar Flooring Systems

www.valsparflooring.com

Pro-Cote Inc.
P. O. Box 1766
Largo, FL 33779
Toll Free (877) 588-4300
They said “I was crazy!”

How do you like the idea of free and low-cost textbooks? Well here is how you can help me in the effort to bring more of these types of books to you. They say there is safety in numbers, well I can promote this type of textbook production way better if I hear from you personally. Send me an email at Basham.Matt@spcollege.edu just to tell me you are using the book, you like the idea of cheaper high-quality books, or to just say “hello.” Together we will change education for the better!
Installing a NIC: Hardware

Objectives:
To be able to install a network interface card (NIC) into a personal computer (PC).

Tools and Materials:
(1) PC, a variety of NIC’s, screwdrivers and nutdrivers

Step-by-Step Instructions:
I guess the old phrase “you get what you pay for” really applies to NIC’s. The more inexpensive the NIC, usually the more problems you will have installing it. It usually applies more to the software side but I have seen alignment problems with the hardware side. Do not go cheap on NIC’s unless you want to experiment or have had good experiences with a certain brand of NIC’s before.

1. Unplug the PC power cord from the wall or outlet.

   ***Warning***
   Do not attempt to install a NIC into an energized PC. Electrocution could occur.

   ***Warning***
   Some computer towers have extremely sharp edges within them. In the field we call these “ginsu” cases.

2. Remove the cover from the PC using screwdrivers or nutdrivers. Every PC is different so go slowly, don’t force anything, and ask questions whenever needed.
3. Remove a cover plate from an available slot (usually a PCI or EISA slot) using a screwdriver.
4. Gently slide the NIC into the appropriate slot. You may have to rock it slightly.
5. Attach the NIC with a screw to the case foundation.
6. Replace the cover.
7. Plug in the PC again (it works better that way).
8. You are now ready for the software portion of the installation, although this can vary. Sometimes the workstation will autosense a new card is installed and will take care of it automatically. Other times you will have to manually add the software portion.

Supplemental Lab or Challenge Activity:
1. Try to see how a Token Ring NIC differs from an Ethernet NIC.
2. Go and find out the differences in motherboard slots: MCA, ISA, EISA, etc.
3. When you add more than one NIC to a computer, are you turning it into a router? Why or why not?

So What Have I Learned Here?
You have learned how to physically install a NIC.
Paper Lab: Proper Cable for the Proper Job

Objective:
To learn which type of networking cable to use in which instance.

Tools and Materials:
Paper and pencils
Different colored pencils or markers would be nice.

Background:
You will be putting together lots of equipment with plenty of cables during your career. Knowing which cable to use and when will save you plenty of time, trouble, and potential embarrassment if you get it right from the start. Heck, you can even help someone else later…most network administrators do not know a straight through from a rollover.

Telephones have been around since the late 1800’s and our wiring patterns have evolved from the telephone industry. The two most common wiring patterns are EIA/TIA 568A and EIA/TIA 568B (Electronics Industry Association/Telecommunications Industry Association). There are four pairs of wires in a Category 5-type cable. Pair 1 is the blue pair, pair 2 is the orange pair, pair 3 is the green pair, and pair 4 is the brown pair. For you football fans…“The Blue and Orange Gators play on the Green Grass with the Brown Football.” (Yeah, I went to UF) In fact, 66 and 110 punch down blocks are wired in this fashion:

Blue Pr  \{ White/blue  \\
        Blue  \\
        White/orange  \\
Or. Pr  \{ White/orange  \\
           Orange  \\
Gr. Pr  \{ White/green  \\
          Green  \\
Br. Pr  \{ White/brown  \\
        Brown

Figure 1—punch down block.

Unfortunately our wiring patterns for our cables could not align easily with this pattern (figure 2). They had to go and come up with some other ones (see figure 3).

White/blue—blue—white/orange—orange—white/green—green—white/brown—brown

Figure 2—Matt’s “nice” pattern.
Figure 3—EIA/TIA 568A and B wiring patterns.

Straight Through (ST): Used for connecting **dis-similar devices** (workstations to hubs, switches to routers, hubs to switches, etc.). The cables are wired with the same wiring pattern on each end.

Crossover (xo): Used for connecting **similar devices** (workstations to workstations, switches to switches, hubs to hubs, etc). The cables are wired with pairs 2 and 3 “crossing over” from one end to the other (see also figure 3).
Rollover (ro): Used for connecting communication ports to other communication ports (workstation com ports to router console ports, etc). It does not matter which colors are used here as long as the pattern “rolls over” from one side to the other.

In the following diagrams indicate which type of cable is used, label each cable, apply the appropriate pattern in the drawing, and indicate which port or connection would be used at the each end of the cable.

Peer-to-Peer Cabling

Two workstations and a hub
Three workstations and a hub

Six workstations (3 to a hub) and two hubs
Change hubs to switches:
Add in a router:
Add in a web access:
Are you enjoying the materials? Well be on the lookout for some other manuals and textbooks on
http://www.lulu.com/learningbydoing and http://www.spcollege.edu/star/cisco

The “Script Kiddie Cookbook” Available Now at http://www.lulu.com/learningbydoing

This book title is based upon the Anarchist Cookbook that came out in the 1980’s. In that tremendously popular book it showed you how to build bombs at home. Similarly the Script Kiddie Cookbook will show you step by step hacking and hacking methodology. You will go to jail if you use this information improperly!!!

Learn the basics about computer security with this book. This book assumes you know nothing about security and starts there. All tools and software used are freeware. Other books will build upon the materials in this book. Learn about the nuts and bolts of SPAM, port scanning, Knoppix tools, cookies, and bring it all together with exercises in hacking at legal gaming sites.

GET THE BOOK THE GOVERNMENT DOES NOT WANT YOU TO OWN!!!
Peer-to-Peer Networking/File and Print Sharing

Objective:
To learn how to set up two computers to communicate and share files in Windows 95/98. Windows 2000/XP/ME would have too many variants for file sharing options for me to write about here, so I will only cover P2P for 95/98.

Tools and Materials:
(2) Workstations
(1) Cross-connect cable (a.k.a cross-over cable)

Lab Diagram:

![Lab Diagram](image)

IP address: 192.168.1.1     192.168.1.2
Subnet Mask: 255.255.255.0     255.255.255.0
Gateway: 192.168.1.2     192.168.1.1

Step-By-Steps Instructions: For Windows 95/98

1. Cable the lab as shown. Put one end of the crossover cable in the NIC on one computer and the other end in the NIC of the other computer. Make certain the LED lights up on the NIC when the cable is plugged into BOTH ends. If the lights do not turn on, then check to make sure you have a good crossover cable. Ask your instructor for help if necessary.

2. Change the TCP/IP settings on each computer.

3. In Windows 95/98 to enable file and print sharing right click on “network neighborhood” (just like you did for changing the TCP/IP settings. You should see:

![Network Settings Control Panel](image)

Figure 1—Network settings control panel.
4. Click on the file and print sharing box. You will see:

![Figure 2—file and print sharing control panel](image)

5. Then select the “pick box” for file sharing.” You can pick the one for print sharing if you have printers that need to be shared also. Now you can re-boot (it’s a Canadian term) your computer. It should look like this when you are finished:

![Figure 3—file and print sharing control panel](image)

6. When your computer is rebooting you will still have to put in user names and passwords otherwise you will not have your full networking capabilities. I know it doesn’t sound right but it is “Microsoft” after all. Once your computer reboots we have to actually share some files. Otherwise you wouldn’t see anything when you access the other computer. One easy way to enable file sharing is with the “my computer icon” on your desktop. Double-click on it and you will see something like:

![Figure 4—My computer control panel](image)
7. Then right click on the “C” drive and select sharing. On the other folder you should only see the “C” drive (which in our case is everything).

Figure 5—Now file sharing can be accomplished.

8. If you only want to share a specific folder or document double click on the C drive to open it and then select the folder or document and pick sharing. On the other computer you should only see that folder or document. You should see something like this (pay no attention to that casino folder…its only an example for another lab 😊)

Figure 6—Selecting a specific folder to be shared.

9. In either case you will be presented with a window for setting the parameters for the share. You can create a name for the drive, folder, or document. You can allow full access, read only, or password-protected access to the drive, folder or document.
10. Once you are finished select “apply”, then “OK,” and you should be able to see the drive, folder, or document on the other computer.

Step-By-Steps Instructions: For Windows 2000/XP/ME
1. Cable the lab as shown. Put one end of the crossover cable in the NIC on one computer and the other end in the NIC of the other computer. Make certain the LED lights up on the NIC when the cable is plugged into BOTH ends. If the lights do not turn on, then check to make sure you have a good crossover cable. Ask your instructor for help if necessary.
2. Change the TCP/IP settings on each computer.
3. Test ping from one computer to the other. If successful then you are done!

Supplemental Lab or Challenge Activity:
1. Pick one computer to be the computer for your boss. The other will be the employee. Have only certain folders and documents sharable on the boss’s computer. Have all drives shared on the employee’s computer. Can your boss find out where you have been on the Internet?
2. Why do we use a crossover cable? Why wouldn’t a straight through cable work?
3. Put a dollar sign ($) on the end of a shared file name and see what happens.

So What Have I Learned Here?
In this lab you have learned how to hook up two computers using peer-to-peer networking and file and print sharing. For this you needed to use your knowledge of TCP/IP software settings you learned in an earlier lab. In later labs you will be expanding upon this knowledge to build more complicated networks and more in-depth file and print sharing exercises.

Another Windows 2000 note:
Ok, so they changed the bejeebers out of file and print sharing. I did have a lab on it for Windows 2000 but found out it may be a bit different in each school. Instead, just have your instructor set up any file and print sharing for you specific to your room and school.
Small Single-Hub Networks

Objective:
To learn how to hook up several computers with a hub and share files between them.

Tools and Materials:
(3) Workstations (I used Windows 98 ones for this lab)
(1) Hub (No hubs? A switch set to its default configuration will work fine)
(3) Straight-through cables

Lab Design:

Step-By-Step Instructions:
1. Cable the lab as shown. Each straight-through cable should be connected from the NIC on the workstation to the respective port on the hub.
2. Set up the IP addresses and masks on each workstation. No gateway number is needed because no single device acts as a gateway.
3. Ping from A to B. Ping from A to C. Ping from B to A. Ping from B to C. It should work just fine.
4. Enable file sharing on each computer. Pick something different on each computer to share…a drive, a folder, or several folders.
5. You should be able to access the files from computer to computer now using network neighborhood. If you cannot “see” the icon for the other computer then go out to DOS and try to ping them. If you can ping them then use the “Find computer option in Windows Explorer” to manually bring them up in Network Neighborhood (gotta love that quirky Microsoft in small networks).
You should see something like this:

![Figure 1](image1.png)

**Figure 1**—Using windows explorer to “find” computers on the network.

![Figure 2](image2.png)

**Figure 2**—The “find computer” option pop up window.

If it doesn’t work then check everything you have done so far and reboot everything.

**Supplemental Lab or Additional Activities:**

1. Try to add in more computers. You will have to pick addresses that will work.
2. Try to add in another computer with an IP address of 172.16.1.2 and a mask of 255.255.255.0. Do you think it will work? What happens when you try to find it on the network? Ping it? Share files with it?
3. Is it possible to hide or secretly share a file? How would it work?
4. How would you change the identity of your computer on the network?
5. Try it again but change ip addresses to the 169-net and see if it still works.
6. Open up multiple DOS windows and use continuous pings and watch when workstations are coming up and going down.

**So What Have I Learned Here?**

You have learned how to hook up several workstations to share files using a hub. You learned that the IP addresses had to be within the same “subnet” in order to communicate with each other. Also you were acquainted with the quirks of Microsoft networking for small networks. Microsoft really likes having that hub out there to work.
Small Multiple-Hub Networks

Objective:
To learn how to hook up several computers with a hub and share files between them.

Tools and Materials:
(6) Workstations
(2) Hubs (No hubs? A switch set to its default configuration will work fine)
(6) Straight-through cables (ST)
(1) Cross-over cable (XO)

Lab Design:

Step-By-Step Instructions:
1. Cable the lab as shown. Each straight-through cable should be connected from the NIC on the workstation to the respective port on the hub. Use a crossover cable between the two hubs. It should not matter which port you use depending upon your type of hub. Some have uplink ports that must be used for this
purpose. Check your documentation. Don’t have any documentation? Go out to the web and download it. Think “outside the box”…or “out of the book” in this case.

2. Set up the IP addresses and masks on each workstation. No gateway number is needed because no single device acts as a gateway.

3. Ping from each workstation to each other.

4. Enable file sharing on each computer. Pick something different on each computer to share…a drive, a folder, or several folders.

5. You should be able to access the files from computer to computer now using network neighborhood. If you cannot “see” the icon for the other computer then go out to DOS and try to ping them. If you can ping them then use the “Find computer option in Windows Explorer” to manually bring them up in Network Neighborhood (gotta love that quirky Microsoft in small networks). If it doesn’t work then check everything you have done so far and reboot everything.

Supplemental Lab or Additional Activities:

1. Try to add in another computer with an IP address of 172.16.1.2 and a mask of 255.255.255.0. Do you think it will work? What happens when you try to find it on the network? Ping it? Share files with it?

2. Put in two computers with the same IP address. What kind of message do you see? Does it appear on one workstation or multiple ones?

3. Open up multiple DOS windows and use continuous pings and watch when workstations are coming up and going down.

So What Have I Learned Here?

You have learned how to hook up several workstations to share files using multiple hubs. You learned that the IP addresses had to be within the same subnet in order to communicate with each other. As you build larger and larger networks you can see where planning for IP addresses is important. Errors make the network act weird. Also you were acquainted with the quirks of Microsoft networking for small networks. Microsoft really likes having that hub out there to work

Keep track of updates and changes at http://www.spcollege.edu/star/cisco Scroll to the bottom of the page and click on the “Lab Manual Edits.”
Paper Lab: Binary Numbering

Objective:
To learn how to convert binary numbers into decimal numbers and vice versa.

Tools and Materials:
Paper and pencil
“Bit Bashing” worksheet

Background: Converting Binary to Decimal
If I asked you to count from zero to nine I would expect everyone would have no problem with it. You would respond with “zero-one-two-three-four-five-six-seven-eight-nine.” This is what is known as the decimal (or base 10) system. There are ten possible combinations available for each column. Each column represents a progressively higher power of ten. For example the number 532:

\[
\begin{array}{c|c|c}
10^2 & 10^1 & 10^0 \\
100 & 10 & 1 \\
\hline
532 = & 5 & 3 & 2 \\
\end{array}
\]

This represents 5 units of \(10^2\) (10x10=100) which is 5 hundreds, 3 units of \(10^1\) (10x1=10) which is 3 tens or 30, and 2 units of \(10^0\) (1) which is 2. Put them all together and you get five hundred and thirty-two. Ok. I know you know this stuff already it will just make the transition to learning stuff on binary easier.

Binary is a base 2 system. Instead of ten numbers we only have two numbers: zero and one (0 or 1). Like our decimal system our columns each represents a progressively higher power of 2.

\[
\begin{array}{c|c|c|c|c|c|c|c}
2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
\end{array}
\]

Each column heading represents a decimal number with a binary power. To convert between binary and decimal the rule is simple: Any place you have a “1” you just add the column heading to get the decimal total. For example, if we were given a binary number of 01101101 to convert into decimal we would write it under our “bit-bashing” chart. Then, in any column where a 1 appeared, we would add the column headings together. That would be our binary to decimal equivalent.

\[
\begin{array}{c|c|c|c|c|c|c|c|c}
2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
\hline
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
\end{array}
\]

64+32+8+4+1=109
Now along the column headings we see a 1 in the columns for 64, 32, 8, 4, and 1. So we add these numbers together 64+32+8+4+1=109. Therefore the binary number 01101101 is equivalent to the decimal number 109. Let’s do another one…convert 10010101 to decimal.

\[
\begin{array}{cccccccc}
2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
\end{array}
\]

\[128 + 16 + 4 + 1 = 149.\]

It’s another one of those things: easy when you know how. Let’s take a quick time out and let you try some binary to decimal conversions:

1. 10101010
2. 01010101
3. 11001100
4. 11000101
5. 11111111

Now let’s check your answers with the answer section. Did you get the right ones? I certainly hope so. Try not to use a calculator. You will not be allowed to use one on the CCNA test so get practice without it now.

Converting from Decimal to Binary:
This is just the opposite of what we just did except we use subtraction. If we are given the decimal number 141 to convert to binary we just subtract our number (141) from each column heading in succession until we have a remainder of zero. If we encounter a negative number then we put a zero in our bit bashing column. This is tough to explain without working it through…so let’s learn by doing. Starting out with our 128 column heading: 141 - 128 = 13. So we put a “1” under the 128 heading and move to the next column heading.

\[
\begin{array}{cccccccc}
2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
1 & | & | & | & | & | & | \\
\end{array}
\]

Our next one: 13 - 64 = -51. Since this is negative we put a zero in the column heading for 64 and move on to the next one.

\[
\begin{array}{cccccccc}
2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
1 & 0 & | & | & | & | & | \\
\end{array}
\]
Our next one: 13 - 32 = -19. Since this is negative we put a zero in the column heading for 32 and move on to the next one.

\[
\begin{array}{cccccccc}
2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
1 & 0 & 0 & 0 \\
\end{array}
\]

Our next one: 13 - 16 = -3. Since this is negative we put a zero in the column heading for 16 and move on to the next one.

\[
\begin{array}{cccccccc}
2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
1 & 0 & 0 & 0 & 0 \\
\end{array}
\]

Our next one: 13 - 8 = 5. So we put a “1” under the 8 heading and move to the next column heading.

\[
\begin{array}{cccccccc}
2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
1 & 0 & 0 & 0 & 1 \\
\end{array}
\]

Our next one: 5 - 4 = 1. So we put a “1” under the 4 heading and move to the next column heading.

\[
\begin{array}{cccccccc}
2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 \\
\end{array}
\]

Our next one: 1 - 2 = -1. Since this is negative we put a zero in the column heading for 2 and move on to the next one.

\[
\begin{array}{cccccccc}
2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\end{array}
\]

Our next one: 1 - 1 = 0. So we put a “1” under the 1 heading.

\[
\begin{array}{cccccccc}
2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
\end{array}
\]

And we are done…right? Wrong! We should always double-check our work. To do this we convert from binary back to decimal. By adding the column headings: 128+8+4+1=141. It worked!
Let’s try another one: 223. Starting out with our 128 column heading: $223 - 128 = 95$. So we put a “1” under the 128 heading and move to the next column heading.

<table>
<thead>
<tr>
<th>2^7</th>
<th>2^6</th>
<th>2^5</th>
<th>2^4</th>
<th>2^3</th>
<th>2^2</th>
<th>2^1</th>
<th>2^0</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Our next one: $95 - 64 = 31$. So we put a “1” under the 64 heading and move to the next column heading.

<table>
<thead>
<tr>
<th>2^7</th>
<th>2^6</th>
<th>2^5</th>
<th>2^4</th>
<th>2^3</th>
<th>2^2</th>
<th>2^1</th>
<th>2^0</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Our next one: $31 - 32 = -1$. Since this is negative we put a zero in the column heading for 32 and move on to the next one.

<table>
<thead>
<tr>
<th>2^7</th>
<th>2^6</th>
<th>2^5</th>
<th>2^4</th>
<th>2^3</th>
<th>2^2</th>
<th>2^1</th>
<th>2^0</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Our next one: $31 - 16 = 15$. So we put a “1” under the 16 heading and move to the next column heading.

<table>
<thead>
<tr>
<th>2^7</th>
<th>2^6</th>
<th>2^5</th>
<th>2^4</th>
<th>2^3</th>
<th>2^2</th>
<th>2^1</th>
<th>2^0</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Our next one: $15 - 8 = 7$. So we put a “1” under the 8 heading and move to the next column heading.

<table>
<thead>
<tr>
<th>2^7</th>
<th>2^6</th>
<th>2^5</th>
<th>2^4</th>
<th>2^3</th>
<th>2^2</th>
<th>2^1</th>
<th>2^0</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Our next one: $7 - 4 = 3$. So we put a “1” under the 4 heading and move to the next column heading.

<table>
<thead>
<tr>
<th>2^7</th>
<th>2^6</th>
<th>2^5</th>
<th>2^4</th>
<th>2^3</th>
<th>2^2</th>
<th>2^1</th>
<th>2^0</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Our next one: $3 - 2 = 1$. So we put a “1” under the 2 heading and move to the next column heading.

<table>
<thead>
<tr>
<th>$2^7$</th>
<th>$2^6$</th>
<th>$2^5$</th>
<th>$2^4$</th>
<th>$2^3$</th>
<th>$2^2$</th>
<th>$2^1$</th>
<th>$2^0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Our next one: $1 - 1 = 0$. So we put a “1” under the 1 heading.

<table>
<thead>
<tr>
<th>$2^7$</th>
<th>$2^6$</th>
<th>$2^5$</th>
<th>$2^4$</th>
<th>$2^3$</th>
<th>$2^2$</th>
<th>$2^1$</th>
<th>$2^0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

And we are done…right? Wrong! We should always double-check our work. To do this we convert from binary back to decimal. By adding the column headings: $128+64+16+8+4+2+1=223$. It worked!

Let’s take a quick time out and let you try some decimal to binary conversions:
1. 84
2. 243
3. 24
4. 254
5. 179

Now let’s check your answers with the answer section. Did you get the right ones? I certainly hope so. Try not to use a calculator. You will not be allowed to use one on the CCNA test so get practice without it now. Notice in this lab we have been using 8 binary numbers for our conversions. Each one of those binary numbers is called a “bit” and 8 of them together (which is extremely common in computers) is called an “octet” or “byte.” We can do conversions for more or less bits, but it is just a matter of adding more or less columns to our bit-bashing table.

**Supplemental Lab or Challenge Activity:**
1. Make a binary to decimal conversion chart for all decimal numbers between 0 and 255.
2. Try to calculate the binary numbers for these decimal numbers:
   a. 1024
   b. 4096
   c. 3333
   d. 4309
   e. 64768
3. See if you can find out what are hexadecimal, octal, gray code, and binary coded decimal conversions.
4. You can make a “binary to decimal” self-tutoring aid using standard index cards. On one side of the index card write a big “0” on it. On the other side write a big “1” on it. Then arrange the index cards so all zeroes or all ones are facing up.
Then, using a different color marker write one of the column headings in small numbers along the bottom. Then flip them and do the same on the other side. They should look like this on one side:

```
0 0 0 0 0 0 0 0
128 64 32 16 8 4 2 1
```

And like this on the other side:

```
1 1 1 1 1 1 1 1
128 64 32 16 8 4 2 1
```

Now, instead of adding column headings you can just flip the index cards as needed. Let’s work through one with the index flip cards. Let’s convert 234 from decimal to binary. Start with your cards like this:

```
0 0 0 0 0 0 0 0
128 64 32 16 8 4 2 1
```

Then just subtract the column headings (in this case the little numbers on the bottom of the card)...234-128=106. Since it is a positive number flip the card and move on to the next one.

```
1 0 0 0 0 0 0 0
128 64 32 16 8 4 2 1
```

106 - 64 = 42. Since it is a positive number flip the card and move on to the next one.

```
1 1 0 0 0 0 0 0
128 64 32 16 8 4 2 1
```
$42 - 32 = 10$. Since it is a positive number flip the card and move on to the next one.

$$
\begin{array}{cccccccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
\end{array}
$$

$10 - 16 = -6$. Since it is a negative number leave the card on zero and move on to the next one.

$$
\begin{array}{cccccccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
\end{array}
$$

$10 - 8 = 2$. Since it is a positive number flip the card and move on to the next one.

$$
\begin{array}{cccccccc}
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
\end{array}
$$

$2 - 4 = -2$. Since it is a negative number leave the card on zero and move on to the next one.

$$
\begin{array}{cccccccc}
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
\end{array}
$$

$2 - 2 = 0$. Since it is a positive number flip the card and move on to the next one.
Since our remainder is zero then all other numbers to the right are also zero (only one card in this case).

$$
\begin{array}{cccccccc}
1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
\end{array}
$$
Let me just walk through one more…you can do the math yourself. Let’s convert 158 to binary.

<table>
<thead>
<tr>
<th></th>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

So what have I learned here?
In this lab you learned how to do binary to decimal and decimal to binary conversions. You will be using these later in subnetting labs using IP addresses in decimal and being able to convert them to binary. You will find bit-bashing sheets on the next page.
<table>
<thead>
<tr>
<th>$2^0$</th>
<th>$2^1$</th>
<th>$2^2$</th>
<th>$2^3$</th>
<th>$2^4$</th>
<th>$2^5$</th>
<th>$2^6$</th>
<th>$2^7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Recent events at home and abroad have mandated that Americans can accept nothing short of the best possible training for those working in the fields of security and criminal justice. The *Journal of Security Education: New Directions in Education, Training, and Accreditation* is a comprehensive, one-stop resource on security education and training programs that will help educators, practitioners, and students meet the increasing need for security in the United States. Affiliated with the Academy of Security Educators and Trainers, the journal presents the latest developments in theory, practice, research, and assessment with an emphasis on up-to-date methods, techniques, and technology.
Hexadecimal Numbering

Objective:
To learn how to convert between Hexadecimal, Decimal, and Binary numbers.

Tools and Materials:
Pencil and Paper
Bit-bashing worksheet

Background:
In the previous lab you learned how to convert between a base 2 numbering system (binary) and a base 10 numbering system (decimal). As Emeril says we will be “kicking it up a notch” here by adding in base 16 numbering systems (hexadecimal). Just like our decimal system used the numbers zero-one-two-three-four-five-six-seven-eight-nine to represent the 10 places in a base 10 system we use zero-one-two-three-four-five-six-seven-eight-nine-ten-eleven-twelve-thirteen-fourteen-fifteen to represent the 16 places in a base 16 system. The only difference is since we cannot distinguish a one-four from a fourteen we use letters for ten through fifteen. Therefore our base 16 system is coded:

- 0-9
- 10 A
- 11 B
- 12 C
- 13 D
- 14 E
- 15 F

It’s actually easy once you get used to it. Once again, just like our decimal and binary system, each column would be represented as a power with base 16. If we look at the “column headings” for five bits of hexadecimal numbers they become:

<table>
<thead>
<tr>
<th>16⁴</th>
<th>16³</th>
<th>16²</th>
<th>16¹</th>
<th>16⁰</th>
</tr>
</thead>
<tbody>
<tr>
<td>65536</td>
<td>4096</td>
<td>256</td>
<td>16</td>
<td>1</td>
</tr>
</tbody>
</table>

Let’s start with binary to hexadecimal conversions using octets…they are the easiest. Since there is eight bits these are easy:

1. We just divide the octet into two groups of 4 bits
2. Make new column headings
3. Add them up.
4. Then, with those totals, we use our decimal to hexadecimal conversion chart above to complete the conversion.
For example, let’s convert the binary octet 11001101 to hexadecimal.
1. We just divide the octet into two groups of 4 bits

```
  1 1 0 0 1 1 0 1
  1 1 0 0   1 1 0 1
```

2. Make new column headings

```
  8 4 2 1   8 4 2 1
  1 1 0 0   1 1 0 1
```

3. Add them up.

```
  8 4 2 1   8 4 2 1
  1 1 0 0   1 1 0 1

8+4=12   8+4+1=13
```

4. Then, with those totals, we use our decimal to hexadecimal conversion chart above to complete the conversion.

```
  12=C   13=D
```

The binary octet “11001101” is equivalent to the hexadecimal number “CD.”
Let’s do another one: convert 10010111 from binary to hexadecimal.
1. We just divide the octet into two groups of 4 bits

```
  1 0 0 1 0 1 1 1
  1 0 0 1   0 1 1 1
```

2. Make new column headings

```
  8 4 2 1   8 4 2 1
  1 0 0 1   0 1 1 1
```

3. Add them up.

```
  8 4 2 1   8 4 2 1
  1 0 0 1   0 1 1 1

8+1=9   4+2+1=7
```
4. Then, with those totals, we use our decimal to hexadecimal conversion chart above to complete the conversion.

\[
9 = 9 \quad \quad 7 = 7
\]

The binary octet “11001101” is equivalent to the hexadecimal number “97.” Don’t think in terms of decimal…this is NOT ninety-seven. In hexadecimal this is “nine-seven.”

You can convert from decimal to binary and then to hexadecimal. We use subscripts to denote which base of number we are using (2 for binary, 10 for decimal and 16 for hexadecimal). Try it with these:

1. \(143_{10}\)
2. \(244_{10}\)
3. \(78_{10}\)
4. \(128_{10}\)
5. \(191_{10}\)

Check your answers. Hopefully you are correct! If you have decimal numbers with more than 255 (our binary octet upper limit) then we have ways to convert them too. To convert a decimal number to hexadecimal we just keep dividing it by 16 until we get to zero. The remainders, in reverse order, are used to code the hexadecimal. For example let’s convert the decimal number 28436 to hexadecimal:

\[
\begin{align*}
28436 \text{ divided by } 16 &= 1777 \quad R \quad 4 \\
1777 \text{ divided by } 16 &= 111 \quad R \quad 1 \\
111 \text{ divided by } 16 &= 6 \quad R \quad 15 \\
6 \text{ divided by } 16 &= 0 \quad R \quad 6
\end{align*}
\]

The remainders, in reverse order, are 6-15-1-4. When we replace 15 with F we get our hexadecimal conversion of 6F14 (“six-F-one-four”). Ok…So I know a lot of you cheated and used a calculator. Here is a chart for the remainders converted to whole numbers:

<table>
<thead>
<tr>
<th>R</th>
<th>0/16</th>
<th>0.0000</th>
<th>R</th>
<th>8/16</th>
<th>0.5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>R 1</td>
<td>1/16</td>
<td>0.0625</td>
<td>R 9</td>
<td>9/16</td>
<td>0.5625</td>
</tr>
<tr>
<td>R 2</td>
<td>2/16</td>
<td>0.1250</td>
<td>R 10</td>
<td>10/16</td>
<td>0.6250</td>
</tr>
<tr>
<td>R 3</td>
<td>3/16</td>
<td>0.1875</td>
<td>R 11</td>
<td>11/16</td>
<td>0.6875</td>
</tr>
<tr>
<td>R 4</td>
<td>4/16</td>
<td>0.2500</td>
<td>R 12</td>
<td>12/16</td>
<td>0.7500</td>
</tr>
<tr>
<td>R 5</td>
<td>5/16</td>
<td>0.3125</td>
<td>R 13</td>
<td>13/16</td>
<td>0.8125</td>
</tr>
<tr>
<td>R 6</td>
<td>6/16</td>
<td>0.3750</td>
<td>R 14</td>
<td>14/16</td>
<td>0.8750</td>
</tr>
<tr>
<td>R 7</td>
<td>7/16</td>
<td>0.4375</td>
<td>R 15</td>
<td>15/16</td>
<td>0.9375</td>
</tr>
</tbody>
</table>
To convert a large hexadecimal number into decimal we just write down our hexadecimal codes from the bottom up and then multiply them with successively larger powers of 16. For example let’s convert the hex number 8C3B into decimal:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>48</td>
</tr>
<tr>
<td>C</td>
<td>12</td>
<td>3072</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>32768</td>
</tr>
</tbody>
</table>

\[ B \times 16^0 = 11 \times 1 = 11 \]
\[ 3 \times 16^1 = 3 \times 16 = 48 \]
\[ C \times 16^2 = 12 \times 256 = 3072 \]
\[ 8 \times 16^3 = 8 \times 4096 = +32768 \]

8C3B\(_{16}\) (hex) is equivalent to 35899\(_{10}\) (dec)

**Supplemental Lab or Challenge Activity:**

1. Here are some more to try converting. Be sure to include binary, decimal and hexadecimal conversions for each number.
   a. 2047\(_{10}\)
   b. 1011011101\(_2\)
   c. 9BBB\(_{16}\)
   d. 248\(_{16}\)
   e. 35898\(_{10}\)

2. Try adding hexadecimal conversions to that table you made in the binary numbering lab (from zero to 255).

**So What Have I Learned Here?**

In this lab you have learned about hexadecimal conversions. Hexadecimal is used for MAC addresses and for sending information over the Internet. Later, when you learn to use protocol inspectors, you will be able to “see” the actual codes sent in packet form over the network. Then you can double-check the hexadecimal codes with binary and decimal conversions. After all, we are not making you do math to be mean old fuddy-duddies. Be very certain you know how to do these conversions very well. I have heard the newer version of the test has some of these in there.

---

**The “Script Kiddie Cookbook” Available Mid-August 2004 at**

[http://www.lulu.com/learningbydoing](http://www.lulu.com/learningbydoing)

This book title is based upon the Anarchist Cookbook that came out in the 1980’s. In that tremendously popular book it showed you how to build bombs at home. Similarly the Script Kiddie Cookbook will show you step by step hacking and hacking methodology. You will go to jail if you use this information improperly!!!

Learn the basics about computer security with this book. This book assumes you know nothing about security and starts there. All tools and software used are freeware. Other books will build upon the materials in this book. Learn about the nuts and bolts of SPAM, port scanning, Knoppix tools, cookies, and bring it all together with exercises in hacking at legal gaming sites.

**GET THE BOOK THE GOVERNMENT DOES NOT WANT YOU TO OWN!!!**
Paper Lab: OSI Model and Encapsulation

Objective:
To be able to learn more about the OSI model, its layers, and their descriptions.

Tools and Materials:
Paper and pencil

Background:
In your textbook you have learned about the layers of the OSI model, what happens on each layer, and descriptions of each layer. You probably took the time to memorize exactly the definitions of each layer. I got news for you…on “the” test the definitions are completely different from the ones in the book. Wouldn’t it be nice if they did something consistent for once? Actually the definitions are similar, just completely worded differently. So here we will look at the definitions you were told and try to create some alternate wordings. Your test will probably have something like a drag and drop scenario for it so we will just use simple matching exercises here.

The OSI Model

There are seven layers in the OSI model. From bottom to top we number them from layer 1 to layer 7. They are the: physical, data link, network, transport, session, presentation, and application layers.

The reason we need to understand which layer is which number is to be able to decipher sales brochures. Sometimes they refer to layer 2 devices, of which we could think “bridges” or “switches.” (Switches? We don’t need no stinking switches).

As a memory device we can remember from the top down that All Presidents Seem To Need Data Processors or “All People Seem To Need Domino’s Pizza.” There are other mnemonic memory devices like something about taking spinach pizza always, but these seems to work best for most people

Let’s take a brief look at each layer:

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>identifies and establishes the availability of intended communication partners, synchronizes cooperating applications, and establishes agreement on procedures for error recovery and control of data integrity. “browsers”</td>
</tr>
<tr>
<td>Presentation</td>
<td>translates multiple data representation formats by using a common data representation format. “concerned with data structures and negotiation data transfer syntax” “encoding, representation of data, ASCII”</td>
</tr>
<tr>
<td>Session</td>
<td>synchronizes dialogue between presentation layer entities and manages their data exchange. Information is encapsulated into data blocks here.</td>
</tr>
<tr>
<td>Transport</td>
<td>Responsible for reliable network communication between end nodes and provides transport mechanisms for the est., maintenance, and termination of virtual circuits, transport fault detection and recovery and information flow control.</td>
</tr>
</tbody>
</table>
Network | Provides connectivity and path selection between two end systems where routing occurs. Segments are encapsulated into packets here.
---|---
Data Link | Concerned with physical addressing, network topology, and media access. Packets are encapsulated into frames here.
Physical | Describes the various types of networking media. Frames are converted into bits here. Defines the electrical and functional specifications for activating and maintaining the link between end systems.

note: stress underlined areas as “buzz words” to remember for each layer.

Let's take a peek at each layer…
Now when you go to communicate over the network your host computer will begin readying for transmission from the top down. Therefore, we will start with Layer 7: the Application layer.

**Layer 7: the Application layer**
The official definition is the application layers "identifies and establishes the availability of intended communication partners, synchronizes cooperating applications, and establishes agreement on procedures for error recovery and control of data integrity. “browsers” This is the layer the user will see. Correlation's here include FTP, HTTP, MS-Word, etc.

Visual representation: Application box (MS-WORD, etc), big eyes

**Layer 6: The Presentation layer**
The official definition of the presentation layer is that it "translates multiple data representation formats by using a common data representation format. "concerned with data structures and negotiation data transfer syntax” “encoding, representation of data, ASCII”

This is the layer that is in charge of "Super Secret Spy Stuff" and "Key" coding. This is where we compress and encrypt our information before sending. Examples here include ASCII and PKZIP.

Visual Representation: sunglasses and hat; big key

**Layer 5: The Session layer**
The official definition of the session layer is that it "synchronizes dialogue between presentation layer entities and manages their data exchange. Information is encapsulated into data blocks here."

This is the layer that says "HEY!" I want to establish a networking session. In fact, if you have internet access from your home computer then you may even see the message "establishing session" during the connection process.

Visual Representation: Big lips
Layer 4: The Transport layer
The official definition of the transport layer is that it "Responsible for reliable network communication between end nodes and provides transport mechanisms for the establishment, maintenance, and termination of virtual circuits, transport fault detection and recovery and information flow control."

This is the layer where information is readied for transmission. For example, if we were to make a large packaging machine about 60 feet long and 20 feet wide that we wish to ship we would have to "break it down" into smaller chunks before sending it. These chunks would be numbered 1 of x, 2 of x, 3 of x, etc. In this manner we could assure that all packages were sent and received. All of the chunks would be placed into the semi-trucks for transport. Could they be delivered now? Nope, lets move on to the next layer.

Visual Representation: Semi-truck

Layer 3: The Network layer
The official definition of this layer is that it "provides connectivity and path selection between two end systems where routing occurs. Segments are encapsulated into packets here."

Now before we can start sending out our shipment we need to give it a destination and the directions on how to get from here to there. This layer is also in charge of logical addressing.

Visual Representation: map

Layer 2: The Data Link layer
The official definition of this layer is that it is "concerned with physical addressing, network topology, and media access. Packets are encapsulated into frames here.

The data link layer is in charge of physical addressing and a little bit of error checking called "cyclic redundancy checking." CRC calculates the total size of the packets, divides the total size by a unique prime number (a number divisible only by itself and one) and attaches it to the packet. This is also the layer where the NIC card functions.

Visual Representation: MAC, CRC, LLC.

Layer 1: The Physical layer
The official definition of this layer is that it "describes the various types of networking media. Frames are converted into bits here. Defines the electrical and functional specifications for activating and maintaining the link between end systems.

The physical layer is, simply put, the media or cabling.

Visual Representation: cables
Encapsulation

As we move down the OSI model a process called encapsulation takes place. At the session layer the information is called "data." At the transport layer the data is converted into "segments." At the network layer the segments are encapsulated into "packets." At the data link layer the packets are now encapsulated into "frames." Finally, at the physical layer the frames are converted into "bits."

A good way to remember this is “Don’t Send People Free Beer.” Beer is on the physical layer because its macho. If you want to remember it from the bottom up (which might confuse you with the OSI model direction) you can remember “Been free people since democracy.”

Pay close attention to when the information headers and footers are added. This can be somewhat confusing. Let’s take a look at a make believe situation between two users communicating over the Internet. Suppose Joe wants to send an email to Casey. His message is 50,000 bytes in size at the application layer. This email is passed down to the presentation layer where it is compressed, encrypted, and formatted down to a message of 30,000 bytes in size (ok…so it really won’t be this neat but cut me a break it is easier to explain this way). Then the 30,000 byte compressed, formatted, and encrypted data is sent to the session layer. Here Joe’s computer establishes a session with Casey’s computer…

Session layer communication:
Joe: Hey Casey…can I hook up with you (no pun intended)
Casey: I acknowledge that you are requesting a hook up
Joe: I received your acknowledgement of my request for a hookup.
Casey: I received your acknowledgement of my acknowledgment of your request for a hookup.

Then the data is passed to the transport layer for numbering. Here the 30,000 byte data is broken down into 6 segments and numbered: 1 of 6, 2 of 6, 3 of 6, 4 of 6, 5 of 6 and 6 of 6. Handshaking and windowing takes place to finish the establishment of the session.

Transport layer communication:
Joe: I want to send information so how quickly can I send it?
Casey: I acknowledge that you are requesting to send information.
Joe: I received your acknowledgement of my request to send information.
Casey: I received your acknowledgement of my acknowledgment of your request to send information.
Casey: I am not busy so you can transmit at 22300 bps.
Joe: I acknowledge that you can transmit at 22300 bps.
Casey: I received your acknowledgement of my request to transmit at 22300 bps.
Joe: I received your acknowledgement of my acknowledgment of your ability to receive information at 22300 bps.

Then the transport layer segment is passed to the network layer. The network layer adds the source and destination ip addresses (logical addresses) plus some other stuff (we will look at later). Then the new “packet” is sent to the data link layer. There the data link
layer adds LLC, CRC, and MAC information. The LLC is just instructions on how to get from layer 1 to layer 3. MAC information is the hexadecimal, 48-bit, physical address of the source and destination. The CRC is an error-checking mechanism for the data link layer. It essentially works like this: Now that the “frame” is nearly completed the overall number of bits is divided by a unique prime number (a number divisible only by one and itself…17 and 31 are most common). With all the overhead of the headers and footers our individual frames may be 6808 bytes in size by now. So the CRC divides 6808 by 17 (I picked which one our network is using arbitrarily). and we get 400 with a remainder of 8. The 17 is attached along with the remainder of 8. When this frame gets to Casey the division will take place again. If the same remainder is attained then “Casey” will assume everything came over ok. Also, since all of our Ethernet, Token Ring, Frame Relay, ATM, etc. is found on the data link layer that information also is added (before the CRC stuff). Finally the entire frame is passed to the physical layer where it is converted from hex into decimal and transmitted over the network. On Casey’s computer the information is received, checked and re-assembled. In our case 6 chunks of information that are 6808 bytes are received (40,848). If we follow our same compression ratio of 5:3 then we would expect the 40,848 to be un-compressed to over 68,000 bytes. However, since all of the headers and footers are removed after being de-compressed our original message will be back to its original size of 50,000 bytes. This is why, when you download something from the Internet, a 100,000 byte download counts up to about 130,000 bytes before being “finished” but then is only 100,000 bytes when you look at it. Aha! Mysteries of the Internet Revealed! Even better than Geraldo and Capone’s Vault.

**Step-By-Step Instructions:**
Ok…so those are the definitions/encapsulations that they asked you to know. Let’s take a few seconds to re-write them in our own words.

<table>
<thead>
<tr>
<th>Layer</th>
<th>CISCO definition</th>
<th>Your definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>identifies and establishes the availability of intended communication partners,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>synchronizes cooperating applications, and establishes agreement on procedures for</td>
<td></td>
</tr>
<tr>
<td></td>
<td>error recovery and control of data integrity. “browsers”</td>
<td></td>
</tr>
<tr>
<td>Presentation</td>
<td>translates multiple data representation formats by using a common data representation format. “concerned with data structures and negotiation data transfer syntax” “encoding, representation of data, ASCII”</td>
<td></td>
</tr>
</tbody>
</table>
Session synchronizes dialogue between presentation layer entities and manages their data exchange. Information is encapsulated into data blocks here.

Transport Responsible for reliable network communication between end nodes and provides transport mechanisms for the est., maintenance, and termination of virtual circuits, transport fault detection and recovery and information flow control.

Network Provides connectivity and path selection between two end systems where routing occurs. Segments are encapsulated into packets here.

Data Link Concerned with physical addressing, network topology, and media access. Packets are encapsulated into frames here.

Physical Describes the various types of networking media. Frames are converted into bits here. Defines the electrical and functional specifications for activating and maintaining the link between end systems.

Let’s compare. My definitions of the OSI model layers are:

**Application**—Where most non-networking programs function. This is the layer where networking (like client-server) and the encapsulation process starts and ends.

**Presentation**—The second step in networking. This is where data is compressed, formatted or encrypted. The “super-secret-spy-stuff” layer.

**Session**—This is where networking “sessions” between two devices are started, managed, and terminated. The information is called “data.”

**Transport**—This is where the data is “chunked” into “segments” before being passed to the network layer. Each chunk/segment is labeled 1 of X, 2 of X, 3 of X, etc. This is the layer predominantly in charge of error control, even though each individual layer has its own error control (to a lesser extent).

**Network**—This is where each segment is given directions on how to get from here to there using logical addresses. After this information is added the segment is called a “packet.”
Data Link—Takes care of topologies and physical addresses. The packet is now called a “frame.”

Physical—Where the media is located. No intelligent processing takes place here just conversion to binary.

Matching:
Please match the definition on the left with the corresponding OSI layer on the right.

1. ____ Agreement of using ASCII is performed here.  Presentation
   Physical

2. ____ Signals are amplified here.  Session
   Transport

3. ____ Version of protocol used will be found here.  Data Link
   Application

4. ____ Responsible for terminating communication between network devices.  Network

Please match the item on the left with the corresponding OSI layer on the right.

1. ____ Manage communication session  Presentation
   Transport

2. ____ Capturing Packets  Session
   Network

3. ____ Flow Control  Application
   Physical

4. ____ Logical addressing  Data link

So What Have I Learned Here?
That they really want you to know your layers inside and out…not just an exact definition but other similar definitions. Let’s face it…it's enough to drive you friggin nuts. The only advice I can give is to memorize the one’s that are extremely technical, geeky, and just plain obnoxious. Then write your own definitions to check your understanding of the layers and have someone else (like a teacher or really knowledgeable friend) check them over for accuracy.

Hungry for more???
You are almost there for the CompTIA Network+ and Security+ Certifications too!
For more information: http://www.comptia.org/certification/default.aspx

Build successful careers for students everywhere…
**Protocols and the OSI Model**

*Objective:*  
To be able to identify protocols, protocol suites, and their relationships to the OSI model.

*Step-By-Step Instructions:*

1. Find a network protocol table or poster somewhere on the Internet. I used to have a cool site for ordering a protocol poster but they made it into a site that dumps all kinds of junk down onto your desktop. Shame on them. Try doing a search for “free protocol poster” or “communications protocols” posters from time to time. Protocols have overlap so be careful when filling out this chart.

### TCP/IP Suite

<table>
<thead>
<tr>
<th>OSI Model</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

### Novell Suite

<table>
<thead>
<tr>
<th>OSI Model</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Yeah…I know…I didn’t include layers 1 and 2…those are common to all suites and I will put them at the end.
<table>
<thead>
<tr>
<th>IBM Suite</th>
<th>OSI Model</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>ISO Suite</td>
<td>OSI Model</td>
<td>Protocol</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>
### DECnet Suite

<table>
<thead>
<tr>
<th>OSI Model</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

### XNS Suite (Xerox)

<table>
<thead>
<tr>
<th>OSI Model</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>OSI Model</td>
<td>Protocol</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**Appletalk Suite**

<table>
<thead>
<tr>
<th>OSI Model</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**Banyan Vines Suite**

<table>
<thead>
<tr>
<th>OSI Model</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
### Layer 2 Technologies: LAN’s

<table>
<thead>
<tr>
<th>OSI Model</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Layer 2 Technologies: WAN’s

<table>
<thead>
<tr>
<th>OSI Model</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Layer 1 Technologies

<table>
<thead>
<tr>
<th>OSI Model</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Supplemental Labs or Challenge Activities:
1. With which layers of the OSI model and Protocol Suite are these protocols associated?
   a. SMTP
   b. NetBIOS
   c. ASP
   d. SLIP
   e. PPP
   f. FTP
   g. HDLC
   h. CDP
   i. RIP
   j. Token Ring
   k. SCP
   l. CSMA/CD
   m. EIGRP

So What Have I Learned Here?
You have started looking at a “holistic” view of networking. It is extremely possible that you may see questions about protocols and their relationship to the OSI model on your CCNA. I would strongly recommend knowing the TCP/IP suite inside-out and being familiar with the Novell Suite at a minimum. Where do you think wireless fits in here?

---

Looking for college courses? We deliver.

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education.

Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

St. Petersburg College’s eCampus
http://e.spcollege.edu

Your time. Your place. Your future.
Objective:
To be able to learn more about the LAN topologies used in networking.

Tools and Materials:
Paper and pencil

Background:
In your textbook you have read about many topologies. Let’s take some time to go over the specifics of each topology. Many textbooks seem to broadly categorize three types of topologies as the “basics.” These include: bus, star, and ring.

A **bus topology** has all devices connected to a central backbone cable with terminating resistors on each end of the central backbone cable. This really is not used too much anymore since one computer, connector, or cable segment can cause the entire network to go down.

Bus Topology Diagram:

![Bus Topology Diagram](image)

Bus topologies typically used coaxial cabling (50 to 62 ohm…not the 75 ohm for your cable television). Names here include “thick net” and “thin net.”

**Star topologies** have all networking devices connected to a central device. In fact you have already built one in your earlier labs on small networks with a hub.

Star Topology Diagram:

![Star Topology Diagram](image)

Star topologies usually use category 5 or 5e UTP or STP cabling. Star topologies are used in Ethernet networks.
**Ring topologies** have every device connected to exactly two other devices. As a good example have your class stand up and hold hands to form a ring. Ok…so it’s a bit corny but it is a good “hands on” (so to speak) example of a ring topology.

Ring topology diagram:

![Ring Topology Diagram](image)

Ring topologies are used in FDDI networks too.

It is fairly certain that most larger networks fall into the general category called “hybrid” which means some of this and some of that.

**Hybrid Network:**

![Star-Ring Hybrid Network Diagram](image)
There are all kinds of other topologies that are just “more extreme” versions of the three basic topologies:

1. Extended Star
2. Mesh
3. Tree
4. Irregular
5. Cellular

**Extended Star**: Two or more star networks connected together with a backbone cable.

**Mesh or Full-mesh**: Every computer or networking device connected to every other computer or networking device (used primarily in frame relay networks).
**Tree**: Like a hard drive structure with folders and documents. (I just used workstations to show the overall structure…other networking devices would be included and used to pass network traffic).

![Tree Diagram](image)

**Irregular**: Free-form networking. (I just used workstations to show the overall structure…other networking devices would be included and used to pass network traffic).

![Irregular Diagram](image)

**Cellular**: Exacting cells with a networking device at the middle. Nodes and networking device use wireless networking. (I just used workstations to show the overall structure…other networking devices would be included and used to pass network traffic).

![Cellular Diagram](image)
Supplemental Lab or Challenge Activity:
1. Draw the network for your classroom and identify the LAN topology.
2. Draw the network for your floor or building and identify the LAN topology.
   Document each sub-network type (ie. A backbone ring to connect the star
topologies in each classroom).

So What Have I Learned Here?
In later labs you will look at LAN topologies from other companies (which will mostly
be hybrids…but you will “see” elements of our basics…bus, star, and ring). As your
familiarity with these topologies and network design grows so will your level of
understanding grow about the pro’s and con’s of each network topology.

Are you enjoying the materials? Well be on the lookout for some
other manuals and textbooks on
http://www.lulu.com/learningbydoing and
http://www.spcollege.edu/star/cisco

In preparation for release in 12-18 months:
Basham, M.J. Learning by Doing: Acceptable Use Policy's and their implementation in
networking.
Basham, M.J., Curtis, R., and Brown, J. Learning by Doing: Fundamental Principles of
Using Knoppix
Basham, M. J., Brown, J. and Curtis, R. Learning by Doing: Using Knoppix for security
testing: Principles, and Applications.
Networks.
Basham, M.J. Learning by Doing: CCNP Switching Essentials.

Some of the best books are brought to you by Lulu!

http://www.lulu.com/about/
**Broadcast and Collision Domains**

**Objective:**
To learn how to identify broadcast and collision domains in a network topology.

**Tools and Materials:**
Pencil and paper

**Background:**
In any networking design selection of networking devices can depend upon isolation of traffic using knowledge of broadcast domains and collision domains.

A broadcast domain is an area in which any “network broadcast” is sent to every device in the broadcast domain. For example, if a workstation is set up to get its IP address from a DHCP server it uses a “broadcast address” that is sent over the network to retrieve the IP address from the DHCP server. So, in a way, a broadcast address is like a maintenance channel. It exists so individual devices can broadcast messages to one or every device within the broadcast domain. By keeping the broadcast domains smaller we are reducing the overall network traffic. We use routers to create separate broadcast domains. Each interface on a router is a completely separate broadcast domain. Therefore broadcasts within one network on an interface will not pass to the network on another interface (unless we program the router to do so which is not likely).

A collision domain is an area where collisions can occur in a network. Using Layer 1 devices create one large collision domain. Each port on a Layer 2 device is its own collision domain reducing the possibility of collisions and errors down to nothing.

So let’s jump into defining and identifying collision and broadcast domains. Along the way you will also learn more about how networking devices function.

![Diagram of a network with workstations](image)

**Figure 1**—Small hubbed network.

Since no “intelligent functions” can take place with a hub (they only clean-up, amplify and re-time signals) we have one big broadcast domain and one big collision domain. The likelihood of collisions is high. A hub basically allows transmission on
only one port at a time. The hub allows port one “x” seconds to transmit (but it doesn’t send a notification to port 1 that it is their turn) then changes to port two if no information is transmitted. It allows port one to finish then changes to port two. It will allow port two “x” seconds to transmit and then it will change to port three if no information is transmitted. The process is repeated on port three, then four, then five and then to all the ports one at a time. But, as we have said, hubs are not intelligent. Once the hub finds information being transmitted over a port it does not go to the next port it starts back over at the first port. Therefore you want your more important devices on the first ports.

In our diagram let’s look at an example for workstation “A” to send information to workstation “D.” The information from workstation “A” enters the hub on port 1. The hub then makes duplicate copies of that information and sends it to each port (active or not). In this case workstations “B,” “C,” and “D” will receive the copies. The information is received on the workstations and the de-encapsulation process is started. The frame has the header and footer information removed. First the CRC process will reveal if the information is correct. Next, the destination MAC address is checked to see if it matches the MAC on the workstation (Is this for me?). If they match then the de-encapsulation process continues (which it does only on computer D). If they do not match (which it does not on computers B and C) then the frame and all its information is discarded and ignored. Therefore only the destination device (computer D), for which it was intended, will process the information.

![Diagram showing network traffic]

to: 00-00-00-00-00-04
from: 00-00-00-00-00-01

Workstation “A”
mac: 00-00-00-00-00-01

Workstation “B”
mac: 00-00-00-00-00-04

Workstation “C”

Workstation “D”

00-00-00-00-00-04

Figure 2—Workstation A sends a request to workstation D.

As we have seen with a hub making multiple copies of each incoming request the chances for a collision are high. Let’s look a bit deeper at what happens during a “collision.” Most textbooks and teachers will tell you workstations will “listen” before transmitting. Do they have ears? I do not think so. A NIC just monitors the transmitting pin and receiving pin for voltage for a short period of time. By detecting this voltage the workstation is “listening” to the network for transmissions. When the voltage is detected on both pins the networking devices “sees” this as a collision and grounds the media for a period of time (which stops the collision…this is called a “jam signal”). Then the workstation randomly picks a number of milliseconds to wait to re-transmitting its information (called the back-off algorithm).
This is why we must select our networking devices carefully: to reduce the possibility of collisions. Today higher-level networking devices, such as switches and routers, are available at lower costs, which make them more affordable for installation. Switches eliminate the possibility of collisions because each port is its own collision domain. With one device on a port we have absolutely no chance of a collision happening. Using a switch also “divides” up the available bandwidth from a backbone line to each port. Unlike a hub, our switch can have many simultaneous transmissions. The switch is therefore a more robust device that performs better in networks. We didn’t use them as much in our networks before because they used to be really expensive. In the past few years the prices have come down so much that it is not even worth buying hubs because switches are only a few dollars more. I can buy a 8 port switch for under a hundred dollars. So the only reason to use hubs is when you already have them and do not have the money to spend to upgrade. You should just “phase them in.”

In our previous example we demonstrated how collisions occur. In this example we replace the hub with a switch, which eliminates the possibility of collisions. Each port becomes its own collision domain. A switch, unlike a hub, also has the possibility to store information to be sent out later. That way, if workstation A and D were transmitting at the same time the switch could store information from one workstation while passing on the transmission from the other over the backbone.

A switch is an intelligent device. It allows us to change the priorities of our ports to determine who gets to transmit first in the event of tie. The information from the other port would be stored and transmitted later after the first one is done. Since the possibilities of two workstations transmitting at exactly the same time is remote, we usually won’t have to monkey around with it. I know…I know…I just said we use switches to eliminate collision problems…so why go through all of that hassle and expense to replace hubs with switches? First, as we have said switches do not cost much anymore. Second, a key word in networking design is “scalability” the ability to grow without replacing equipment. We get more functionality out of a switch than with a hub.
so why not just use it now? A switch is more scalable than a hub. And, third, switches are cool. Many of my cohorts and colleagues believe switching will become more prevalent in networking than routing. We use switches at the core of our networks, not routers. Switches only use layer 2 information to make decisions. Routers need layer 2 and 3 information to make decisions so they tend to be slower (in geek-speak: switches have less latency than routers).

So where were we? Oh yeah, switches eliminate collision domain problems. Let’s look at our network diagram again. Now we have many collision domains (one per port) and one big broadcast domain. Workstation A and D could communicate almost instantaneously with each other or to other ports and their devices.

But we still have that one big broadcast domain hanging out there…don’t get me wrong big broadcast domains aren’t necessarily bad but we would like to keep them as small as possible. As we said earlier a broadcast domain is used for network “maintenance.” One analogy for a broadcast domain may be the public address system in your classroom. The staff can make announcements to the whole school or can communicate with just an individual classroom. By keeping the broadcast domain as small as possible we keep our “overhead” traffic as minimal as possible and, therefore, lessen any possible network traffic.

You may have heard someone refer to Novell as a “chatty” network. What they really mean is there is a lot of network broadcasting on the broadcast channel. Each networking device in a Novell uses “SAP” (Service Advertising Protocol). Periodically every single device in a Novell network sends out a broadcast “here I am!” message over the broadcast channel (typically every 60 seconds). As you can deduce if you had 100 devices this could create a lot of traffic. Other protocol suites use the broadcast address channel, albeit to a lesser extent. TCP/IP uses the broadcast channel for ARP/RARP (Address Resolution Protocol, Reverse Address Resolution Protocol). These are used when the workstations are booted that need to find their IP or MAC addresses if they have not been “statically” configured. You will learn more about ARP/RARP later.

Now let’s say our company is growing so we need to add in another network.
Now we would have 8 collisions in our one broadcast domain. Would you think our link between the switches be considered a collision domain too? Gotta say no here because switches have the ability to store information and send it off later (geek speak: queueing). Therefore no collision possibility exists.

Now that we have multiple switches we have the possibility for excessive broadcasts that could slow our network down. Ok…with three or four workstations on each switch it would never get that bad, even with Novell, but cut me a break here ok? We could use a router to reduce our broadcast domain size. Each interface on a router, in fact, is its own broadcast domain. So let’s add a router into our network. Here we would have eight collision domains and two broadcast domains.
Supplemental Labs or Challenge Activities:
Let’s have you count up the number of collision domains and broadcast domains in several network types.

1. Collision Domains: ____________  Broadcast Domains: ________________

2. Collision Domains: ____________  Broadcast Domains: ________________

3. Collision Domains: ____________  Broadcast Domains: ________________


5. Collision Domains: ____________  Broadcast Domains: ________________
6. Collision Domains: ____________ Broadcast Domains: ________________

7. Collision Domains: ____________ Broadcast Domains: ________________

8. Collision Domains: ____________ Broadcast Domains: ________________

The redundant link will act as a backup in case the main link goes down. You will learn how to set up redundant links between switches in Part 2.

Ok…got the idea? Let’s start getting bigger!

10. Collision Domains: ____________  Broadcast Domains: ________________

11. Collision Domains: ____________  Broadcast Domains: ________________

This is a better design.

This is an “OK” design.

So What Have I Learned Here?
In this lab you learned how selecting networking devices can enhance or degrade network performance. You learned how switches and hubs work. You also learned how to identify broadcast and collision domains.

Earn your Bachelor’s Degree from St. Petersburg College in Technology Management

Transferring and having access to a 4-year degree has never been easier, more convenient, or cost effective.

St. Petersburg College allows students who have received an Associate degree (Associate in Arts or Associate in Science, depending on the program) to transfer into the Bachelor of Applied Science in Technology Management.

Opportunities Include:
- **On-line** and in traditional classroom courses
- Day and evening classes
- Full and part-time course schedules

Explore the opportunities waiting for you at St. Petersburg College!

For more information call: (727) 341-4772
E-mail: 4yeardegrees@spcollege.edu
Objective:
You will find here instructions on how and where to download a free protocol inspector called Ethereal. It’s not real pretty but it works…and it’s free. I use it throughout this book.

Step-By-Step Instructions:
1. Go to www.ethereal.com (note: only one “r” the site—figure 1-- with two “rr’s” is a magazine…you will know you are at the wrong page if you see something in French—figure 2).

Figure 1—The “right” site. 

Figure 2—The “wrong site.”

2. On the left-hand side of the “Information” window click on “Download.”
3. Scroll down until you find the link for the windows operating system (see figure 3).
4. Click on the link for “local archive” (see figure 3).

Figure 3—Click on “local archive.”

Figure 4—Click on Win_Pcap_3_0.exe (or most current version).

5. You need a driver library to make this work. Click on the Winpcap packet driver library link (see figure 4) and follow the instructions to save it first, then open it and install it. You may have to re-boot the machine. Make sure you get the latest or most
current version. As I am writing this I know I have a Winpcap version 3.1 beta for a different protocol inspector, so it will be changing again soon.

6. Then you just need to select `ethereal-setup-0.10.5a.exe` and start the “main” program downloading. Then just follow the instructions during the setup and installation.

7. To start a capture use “control+K” then select your NIC card. By default this thing likes to use MAC as an interface (yeah…no icmp with MAC).

8. Then click on “ok.” You should see the counters start for each protocol. It will look something like this:
9. Now we need to generate some traffic. We can ping the other workstation. You should see the ICMP counter increase by 8. Four icmp packets sent to destination and four returned (“echoed”) from the destination. Then click on stop. The packets that were captured will load into Ethereal. You should see something like:

Notice how we have three frames within the window. The top one shows us basic overall information about the packets captured. When we highlight on we are asking Ethereal to show us the contents of that packet. The middle frame is more user friendly. It shows us block by block what we are looking at. The bottom frame shows us the hexadecimal composition of the actual packet.

**Supplemental Labs or Challenge Activities:**


2. Go to the web and look up 2001 Senate Bill 1562 that allows any law enforcement agent to “capture” packets from the internet at any time for any purpose…no subpoena required. They say they can only look at the first 65 bytes of header and footer information but we know better. Using your protocol inspector find out how much they can really see and cannot see.

3. What exactly is the WinPcap file? Its actually really important so go out and look it up on the web.

Keep track of updates and changes at [http://www.spcollege.edu/star/cisco](http://www.spcollege.edu/star/cisco). Scroll to the bottom of the page and click on the “Lab Manual Edits.”
Electronic technology continues to infiltrate industry, electrostatic discharge threatens operations causing costly damage, injury and downtime. Pro-Cote Inc. can protect your work environment against destructive static shock with Valspar ESD Epoxy Flooring Systems.

Certified Conductive and Static Dissipative applications for every need:

- Facilities with highly sensitive electronic equipment
- Computer facilities
- Electronic manufacturing & assembly
- Facilities containing AGV lines
- Clean rooms
- Manufacturing/processing flammable or combustible materials
- Munitions plants

"On Site Certification"

Specialty Applications:

- Non-skid coatings
- Chemical resistant urethanes
- Epoxy coatings
- Moisture cured urethane
- Seamless epoxy floor systems
- Waterborne systems
- VOC compliant systems
- Concrete leveling
- Concrete resurfacing
- Control joint repair

The Management at Pro-Cote Inc. wants to put its expertise to work for you. With over 55 years of combined experience, our goal is to be the best specialty coatings applicator company, period.

Application Experience

- Industrial floor coatings
- Manufacturing production areas
- Distribution and warehouse floors
- Loading docks and ramps
- Safety zones and work areas
- Battery charging areas
- UPS rooms
- Parking garages

Let us help with your specification needs.

"Safety Solutions"

Pro-Cote Inc. is a proud installer of Valspar Flooring Systems

www.valsparflooring.com

Pro-Cote Inc.
P. O. Box 1766
Largo, FL 33779
Toll Free (877) 588-4300

www.pro-cote-usa.com
Free Protocol Inspector: Packetyzer

Objective:
You will find here instructions on how and where to download a free protocol inspector called Packetyzer. It has a few more pretty features than Ethereal, it is based upon the Ethereal “engine” if you will…and it too is free.

Step-By-Step Instructions:
1. Open a web window and type in http://www.packetyzer.com or http://www.networkchemistry.com/products/packetyzer/ both will work.

2. Either scroll to the bottom of the page or click on download.

Download

Packetyzer 2.0.0
- Release notes
- Built with Ethereal 0.10.3 and winpcap 3.01a.
- Winpcap 3.1b will also work with this release, however 3.1b has problems on some computers so it is not bundled with the release.

Packetyzer 0.7.3
- Release notes
- Built with Ethereal 0.10.0 and winpcap 3.01a.

The Packetyzer source code can be found in CVS on SourceForge.

3. Click on Packetyzer 2.0.0.
4. Pick a download mirror and away you go!

5. If it does not work then try:
   [http://voxel.dl.sourceforge.net/sourceforge/packetyzer/Packetyzer_2.0_0_Setup.exe](http://voxel.dl.sourceforge.net/sourceforge/packetyzer/Packetyzer_2.0_0_Setup.exe)

6. Now you just need to follow the instructions and install the software. It may even prompt you to go get the newest winpcap file too.

7. Now you should see an icon on your desktop (unless you told it not to put one there). Click on it and you will see the main window:

8. Next just click on the start button and you should start capturing packets. You may have to reboot your machine after installing the package in order to get it to work correctly and start correctly. Too bad the Canadian’s don’t get a royalty on every reboot, eh?
9. One difference from Ethereal is the packets show up in more detail while they are being collected. Go ahead and have some fun...send out some icmp packets and you can look at them later. After a bit you can hit the stop button and more thoroughly analyze the packets. Here is one of my samples:

10. Ok...a bit similar to Ethereal except on the bottom we see some little tabs. These are the pretty things. Click on the protocols tab. You will get a bar chart view of the protocols in use.
11. You can see all of the same features, things in hex, summaries, and packet structures. Which one do I think is better? The one that works best for you.

12. You can still sort in ascending or descending order by clicking on summary. This sorted my protocols alphabetically. You can quickly see all ARP packets grouped together here. Notice how I have highlighted the target MAC address in the left pane and the corresponding hex information is also highlighted in the lower right panel. We'll talk about this a bit more in the next lab.

Some of the best books are brought to you by Lulu!

http://www.lulu.com/about/
Ethernet Packet Structures

Objective:
To learn about the structure of Ethernet packets.

Background:
So far we have been talking about networking and packets passing over the network. In this lab we will look at the precise structure of packets. Later when we use protocol inspectors you will be able to understand the information better.

Ethernet
Ethernet generally refers to a standard developed by a consortium of the Digital Equipment Corporation (DEC), Intel, and Xerox. It is one of the most widely used encapsulation standards in use for networking today. There have been many versions and revisions to it over the past twenty years. So trying to “nail-down” the exact structure of an Ethernet packet is as easy as nailing jello to the wall. Simply put, you need to be more specific about which Ethernet packet structure you want to examine. There have many different types of Ethernet, or “flavors” if you will, and we will look at the two most common ones: the “generic Ethernet” and “Ethernet SNAP.” Basically our two Ethernet packet structures are the same except the SNAP packet uses part of the data field for LLC sub-layer and SNAP information. In either case the minimum/maximum size of our Ethernet packet is 64-1518 bytes. If the information in the data field will be smaller than the minimum size allowed then it will be “padded” with contiguous zeros to fill the data field up to the minimum size.

802.2/802.3 Ethernet (RFC 894)

<table>
<thead>
<tr>
<th>Preamble</th>
<th>SOF</th>
<th>DA</th>
<th>SA</th>
<th>Type</th>
<th>Data</th>
<th>FCS</th>
</tr>
</thead>
</table>

Figure 1—Generic Ethernet packet structure.

This “Standard for the Transmission of IP Datagrams Over Ethernet Networks” was written by Charles Hornig in 1984 (ftp://ftp.isi.edu/in-notes/rfc894.txt).

Stripped by the NIC:
The preamble can vary in length. The preamble basically is used to help set up the transmission and reception of the information through synchronization. The actual amounts of bits have varied over the years but the principle is still the same: a series of alternating zeroes and ones encompass the preamble. Some of these can be lost during transmission but that is ok. The incoming stream of bits “establishes” that the reception of a packet has started. Most agree on 62 bits. (In hex: 1555555555555 In binary: 0101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101). You will not see this with a protocol sniffer because it is stripped and dumped. What? You don’t believe me? Good, never believe what you read…go out and
test it for yourself! Here I have the packet shown with frame header information highlighted in the left panel. Notice how the corresponding information does NOT show up in the hexadecimal panel on the lower right. Very interesting, indeed.

The **Start of Frame Delimiter (SOF)** further helps to set up the transmission and reception of the information and synchronization. This is only a 2-bit portion with just two one’s. No matter how many zeros and one’s come before the SOF the NIC does nothing until it gets to the one-one (SOF). This information is stripped by the NIC and the NIC can “do its work” on the rest of the packet. (In hex: 3 In binary: 11) You will not see this with a protocol sniffer because it is stripped and dumped.

For example: 0101010101010101010101010101010101010 11

**Preamble**  **SOF**

**Used in de-encapsulation:**

The **Destination Address (DA)** is the physical address (MAC) of the networking device the information is going to be sent to. This is 48 bits in hexadecimal. This will be the first “bits” of information you will see with a protocol inspector. Look in the packet above and you will see it is the first part of the packet you will see. This packet is an ARP broadcast. In the left you will see the broadcast MAC address set to FF’s. Well what do you know? That is the first data shown in hex on the lower right panel. Seeing is believing!
The **Source Address (SA)** is the physical address (MAC) of the networking device sending the information. This is 48 bits in hexadecimal.

The **Type** indicates what types of request will follow. This will be given in hexadecimal. This field is usually 2 bytes. A **0800** in the type field indicates an IP datagram will follow. A **0806** in the type field indicates an ARP request will follow. A **0835** in the type field indicates a RARP request will follow. Let’s “see” this from an ARP packet I captured on my network using Packetyzer. I highlighted the ARP type in the left pane.

Current type codes can be found at [http://www.iana.org/numbers.html#](http://www.iana.org/numbers.html#) or [http://www.cavebear.com/CaveBear/Ethernet/type.html](http://www.cavebear.com/CaveBear/Ethernet/type.html)

```
Type code For:
@  0000-05DC  IEEE802.3 Length Field (0.1:1500.)
+  0101-01FF  Experimental
+*  0800  Internet Protocol (IP)
+*  0806  Address Resolution Protocol (ARP) (for IP and for CHAOS)
  0BAD  Banyan Systems
+  6010-6014  3Com Corporation
  7034  Cabletron
  8037  IPX
+  8060  Little Machines
+  8068  General Dynamics
+  8069  AT&T
+  809B  EtherTalk (AppleTalk over Ethernet)
+  80E0-80E3  Allen-Bradley
+  80F3  AppleTalk Address Resolution Protocol (AARP)
+  8138  Novell, Inc.
  86DD  IP version 6
  AAAA  DECENT?  Used by VAX 6220 DEBNI
```

Some of the more commonly used type codes.
The **Data** is what it sounds like…it’s the “meat” of the information transmitted. For “generic” Ethernet this can be as small as 46 bytes and up to 1500 bytes. The first part of the data field contains the IP header information. See the discussion below on the composition of the data field for both types of Ethernet packets.

The **Frame Check Sequence (FCS)** is the CRC information for error control. This is 4 bytes in hexadecimal. There are many different error control calculations. (Is it a coincidence there are many flavors of Jello © too?) I described one in an earlier lab using unique prime numbers. Another FCS calculation is called “AUTODIN II.” It is calculated using this formula:

\[
(X^{32} + X^{26} + X^{23} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X^{1} + 1)
\]

**802.2/802.3 Ethernet (RFC 1042)**

<table>
<thead>
<tr>
<th>Preamble</th>
<th>SOF</th>
<th>DA</th>
<th>SA</th>
<th>Length</th>
<th>LLC</th>
<th>SNAP</th>
<th>Data</th>
<th>FCS</th>
</tr>
</thead>
</table>

Figure 2—Ethernet SNAP packet structure.

The “Standard for the Transmission of IP Datagrams Over IEEE 802 Networks” was written by Postel and Reynolds in 1988 ([ftp://ftp.isi.edu/in-notes/rfc1042.txt](ftp://ftp.isi.edu/in-notes/rfc1042.txt)). This is more commonly used today.

**Stripped by the NIC:**

The **preamble** can vary in length. The preamble basically is used to help set up the transmission and reception of the information through synchronization. The actual amounts of bits has varied over the years but the principle is still the same: a series of alternating zeroes and ones encompass the preamble. Some of these can be lost during transmission but that is ok. The incoming stream of bits “establishes” that the reception of a packet has started. Most agree on 62 bits. (In hex: 15555555555555 In binary: 0101010101010101010101010101010101010101010101010101010101010101010101010101010101). You will not see this with a protocol sniffer because it is stripped and dumped.

The **Start of Frame Delimiter (SOF)** further helps to set up the transmission and reception of the information and synchronization. This is only a 2-bit portion with just two one’s. No matter how many zeros and one’s come before the SOF the NIC does nothing until it gets to the one-one (SOF). This information is stripped by the NIC and the NIC can “do its work” on the rest of the packet. (In hex: 3 In binary: 11) You will not see this with a protocol sniffer because it is stripped and dumped.

**Used in de-encapsulation:**
The **Destination Address (DA)** is the physical address (MAC) of the networking device the information is going to be sent to. This is 48 bits in hexadecimal. This will be the first “bits” of information you will see with a protocol inspector.

The **Source Address (SA)** is the physical address (MAC) of the networking device sending the information. This is 48 bits in hexadecimal.

The **Length** indicates how much information will follow (but not including the CRC information).

<table>
<thead>
<tr>
<th></th>
<th>802.2 LLC</th>
<th>802.2 SNAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSAP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSAP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>con</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Org</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The **Destination Service Access Point (DSAP)** field determines what protocol this is coming from (Novell/IP etc). The DSAP field is usually set to 0xaa for Ethernet. This is 1 byte.

The **Source Service Access Point (SSAP)** field determines what protocol this is going to (Novell/IP etc). The DSAP field is usually set to 0xaa for Ethernet. This is 1 byte.

The **Control (con)** is 1 byte long and is usually set to a hexadecimal 03 for Ethernet.

The 802.2 SNAP packet is composed of two fields:

The **Organization Code (Org)** is 3 bytes that are all usually set to zeros. In hexadecimal that would be 000000.

The **Type** indicates what types of request will follow. This will be given in hexadecimal. This field is usually 2 bytes. A **0800** in the type field indicates an IP datagram will follow. A **0806** in the type field indicates an ARP request will follow. A **0835** in the type field indicates a RARP request will follow. Current type codes can be found at [http://www.iana.org/numbers.html#](http://www.iana.org/numbers.html#)

The **Data** is what it sounds like…it’s the “meat” of the information transmitted. For “generic” Ethernet this can be as small as 46 bytes and up to 1500 bytes. The first part of the data field contains the LLC information, then the SNAP information and finally the IP header information. See the discussion below on the composition of the data field for both types of Ethernet packets.
The **Frame Check Sequence (FCS)** is the CRC information for error control. This is 4 bytes in hexadecimal. There are many different error control calculations. (Is it a coincidence there are many flavors of jello too?) I described one in an earlier lab using unique prime numbers.

**IP Data Field Composition**

The “Internet Protocol” Standard was written by Postel in 1981 ([ftp://ftp.isi.edu/in-notes/rfc791.txt](ftp://ftp.isi.edu/in-notes/rfc791.txt)). Geeze...it almost sounds like the egg came before the chicken? Well anyway, the IP data field is begun with a header portion of 20 bytes unless options are used.

<table>
<thead>
<tr>
<th>Ver</th>
<th>Hlen</th>
<th>TOS</th>
<th>Length</th>
<th>ID</th>
<th>Flags</th>
<th>FO</th>
<th>TTL</th>
<th>Prot</th>
<th>HC</th>
<th>SA</th>
<th>DA</th>
<th>Opt</th>
<th>Data</th>
</tr>
</thead>
</table>

**Figure 3—IP Data Field Composition**

The **Version** field is 4 bits. This is usually set for IP version 4 (IPv4) although IPv6 is emerging quickly. IPv4 uses 4 bytes and IPv6 uses 6 bytes. In hexadecimal IPv4 is denoted with a 45. IPv6 is denoted with 0x86dd.

The **Header Length** field is also 4 bits. It indicates how many 32-bit portions are in the IP header (including options). The maximum is 60 bytes.

I also put an icmp packet I captured below showing the IP section expanded:
The Type-of-Service field is 8 bits long. The first three bits are not used anymore. The next four are the “type of service” bits and the last bit is always set to zero because it is not used. Only one of the four “type of service” bits can be set to a one at a time while all other bits are set to zero. These indicate what type of service will be performed. The types of service are given by:

<table>
<thead>
<tr>
<th>Type of Service</th>
<th>Binary</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal service</td>
<td>0000</td>
<td>0x00</td>
</tr>
<tr>
<td>NNTP (Usenet news)</td>
<td>0001</td>
<td>0x02</td>
</tr>
<tr>
<td>IGP/SNMP</td>
<td>0010</td>
<td>0x04</td>
</tr>
<tr>
<td>FTP data/SMTP data/DNS zone xfr/</td>
<td>0100</td>
<td>0x08</td>
</tr>
<tr>
<td>Telnet/Rlogin/DNS UDP query/SMTP command phase/TFTP</td>
<td>1000</td>
<td>0x10</td>
</tr>
</tbody>
</table>

The Length field is the length of the IP datagram portion in bytes (maximum size of 65536 bytes).

The Identification field contains a unique number for each sent packet. It is 16 bits and given in hexadecimal.

The Flags field uses one bit of it’s 3 bits to identify that “this packet is part of a larger packet that has been fragmented.”

The Fragment Offset field contains the extra information required with a fragmented packet. The last of this 13-bit field is able to tell the sending node to “never fragment the packet.” If fragmentation is needed and this bit is set it will generate an error message and the information will not be processed. Ahh…playground of the hackers.

The Time to Live (TTL) field sets the maximum number of hops (or routers) that the packet can pass through on the way to its destination.

The Protocol (Prot) field shows which protocol was used to encapsulate and create the data. This field is 8 bits long.

The Header Checksum (HC) is an error control mechanism for this point to the end of the data field. It is 16 bits long.

The Source Address (SA) is the logical address (IP) of the networking device sending the information. This is 32 bits in hexadecimal. Notice how in IP the source address comes before the destination address.

The Destination Address (DA) is the logical address (IP) of the networking device the information is going to be sent to. This is 32 bits in hexadecimal.
The Options (Opt) field can vary in length and is set to accommodate options with IP including security. Again, playground for hackers. Pad bytes of 0 are added here if needed to make the minimum Ethernet packet size.

Last the data field comes. This will vary based upon which type of Ethernet is encapsulating it.

**Internet Control Message Protocol (ICMP)**

I didn’t have this in my first edition of the text but since then Cisco is upping the ante and expecting entry-level folks to memorize the ICMP types, or at least the more common ICMP types. Since I like playing with my protocol inspector why don’t we just go ahead and capture some icmp packets.

You have been using icmp already in previous labs (unless of course you have been skipping around). Yeah…it is better known to newbies as “ping.” Just go out and start the protocol inspector of your choice capturing packets and try a couple of pings to workstations. Just for fun let’s ping everybody at once. How? By sending a broadcast, ie. if you are on the 192.168.1.0 network with a mask of 255.255.255.0 then your broadcast address is 192.168.1.255. You just need to pre-pend the ping command:

```
Ping 192.168.1.255
```
Notice in the upper right panel I was receiving replies from all active workstations, devices, etc. Also notice a Ping request was type 8.

“ICMP” portion of packet in left panel (always last)

<table>
<thead>
<tr>
<th>Type</th>
<th>ICMP meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Ping reply</td>
</tr>
<tr>
<td>3</td>
<td>Destination unreachable</td>
</tr>
<tr>
<td>8</td>
<td>Ping request</td>
</tr>
</tbody>
</table>

Go out to the web and find some other ICMP type codes. Ok, now I am going to take a pre-emptive strike. So far we have seen Cisco wanting ARP codes and ICMP codes…it is only a matter of time before they want another commonly used protocol: DHCP. You guessed it…I already have some packets captured. I turned my protocol inspector on to capture packets, went to my DOS prompt and typed ipconfig, then ipconfig /release, then ipconfig /renew. This generated my DHCP packets. Let’s look at one:

**Bootp Message Types**

<table>
<thead>
<tr>
<th>Value</th>
<th>Message Type</th>
<th>Hex Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Boot Reply</td>
<td>00</td>
</tr>
<tr>
<td>1</td>
<td>Boot Request</td>
<td>01</td>
</tr>
</tbody>
</table>

**DHCP Message Type 53 Values**

<table>
<thead>
<tr>
<th>Value</th>
<th>Message Type</th>
<th>Reference</th>
<th>Hex code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DHCPDISCOVER</td>
<td>[RFC2132]</td>
<td>01</td>
</tr>
<tr>
<td>2</td>
<td>DHCPOFFER</td>
<td>[RFC2132]</td>
<td>02</td>
</tr>
<tr>
<td>3</td>
<td>DHCPREQUEST</td>
<td>[RFC2132]</td>
<td>03</td>
</tr>
<tr>
<td>4</td>
<td>DHCPDECLINE</td>
<td>[RFC2132]</td>
<td>04</td>
</tr>
<tr>
<td>5</td>
<td>DHCPACK</td>
<td>[RFC2132]</td>
<td>05</td>
</tr>
<tr>
<td>6</td>
<td>DHCPNAK</td>
<td>[RFC2132]</td>
<td>06</td>
</tr>
<tr>
<td>7</td>
<td>DHCPRELEASE</td>
<td>[RFC2132]</td>
<td>07</td>
</tr>
<tr>
<td>8</td>
<td>DHCPINFORM</td>
<td>[RFC2132]</td>
<td>08</td>
</tr>
<tr>
<td>9</td>
<td>DHCPFORCERENEW</td>
<td>[RFC3203]</td>
<td>09</td>
</tr>
</tbody>
</table>
Next can you think of any others for you to explore? DNS, Netbios, CDP, STP? Good choices.

**Supplemental Lab or Challenge Activity:**
1. Go out and research the latest RFC’s related to IP addressing and Ethernet structure.
2. Go out to the website for National Semiconductor and download the technical specifications for Ethernet cards. They are very technical but have some good information.

**So What Have I Learned Here?**
You have learned the complicated structures of Ethernet and some other protocols in networking. Plus, you got to play with some protocol inspectors some more. Don’t worry I will have you using them throughout the labs. Even if I don’t tell you to use them, feel free to do so. Seeing is believing.

Keep track of updates and changes at [http://www.spcollege.edu/star/cisco](http://www.spcollege.edu/star/cisco) Scroll to the bottom of the page and click on the “Lab Manual Edits.”
**Paper Lab: Subnetting**

**Objective:**
To learn, in a progressive manner, more about subnets, subnet masking, and IP design.

**Background:**
In this lab many different questions (multiple choice, true-false, essays) are used to bring you up to speed on subnetting. This will give you more practice learning about subnetting that does not jump back and forth between topics too much. Each of my students seemed relieved to have something like this…not just here’s topic, here’s two questions and let’s jump ahead, then back. Since IOS version 12 we can use the first and last subnet in our network design. In a nutshell (how did I get in this nutshell?) Cisco added the command `ip-subnet zero` to let us accomplish this. For these exercises please assume the command is enabled. On your Cisco exam it will tell you whether or not the command is used.

**Changing MAC/IP addresses and Network devices**
1. Bridges make low-level, simple comparisons and decisions about whether or not to forward traffic on a network.
   A. True
   B. False.

2. If the bridge determines that the destination MAC address carried by a data packet is part of the same network segment as the source, it does not forward the data to other segments of the network.
   A. False
   B. True

3. Bridges solve the problem of too much traffic on a network by dividing the network into segments and filtering traffic based on the MAC address.
   A. True
   B. False.

4. When a bridge forwards data on a network, it determines precisely what segment of the network the data will be forwarded to.
   A. True
   B. False

5. When a bridge makes a decision about whether to forward data on a network or not, it uses only the IP address carried by the data in its header.
   A. False
   B. True
6. Which of the following definitions best describes what a frame is?
   A. Router or access server, or several routers or access servers, designated as a buffer between any connected public networks and a private network. It ensures security of the private network.
   B. 32-bit address assigned to hosts using TCP/IP. It belongs to one of five classes and is written as 4 octets separated with periods.
   C. Logical grouping of information sent as a data link layer unit over a transmission medium.
   D. Something used with art to give it another unique perspective.

7. At which of the following layers of the OSI model does routing occur?
   A. Physical layer
   B. Data link layer
   C. Network layer
   D. Transport layer

8. At which of the following layers of the OSI model does bridging occur?
   A. Physical layer
   B. Data link layer
   C. Network layer
   D. Transport layer

9. At which of the following layers of the OSI model is the MAC address located?
   A. Physical layer
   B. Data link layer
   C. Network layer
   D. Transport layer

10. If a workstation is moved within a network, then what will happen to its MAC and IP addresses?
    A. its MAC address and IP address will stay the same
    B. its MAC address will change but the IP address will stay the same
    C. its IP address will change but the MAC address will stay the same
    D. both IP and MAC address will change

11. If a workstation is moved from one network to another network, then what will happen to its MAC and IP addresses?
    A. its MAC address and IP address will stay the same
    B. its MAC address will change but the IP address will stay the same
    C. its IP address will change but the MAC address will stay the same
    D. both IP and MAC address will change
12. Routers pass packets between ______________?
   A. servers on the different networks
   B. routers on the same network
   C. hosts on the different networks
   D. hubs on the same network

13. Which part of the IP address does a router ignore during path determination?
   A. the host address
   B. the network address
   C. the source address
   D. the destination address

14. MAC addresses use a __________ scheme while IP addresses use a __________ scheme.
   A. hierarchical, flat
   B. flat, hierarchical
   C. flat, layered
   D. layered, flat

15. Which type of address is included in an IP header?
   A. source MAC, source IP
   B. destination IP, destination MAC
   C. source IP, destination IP, source MAC
   D. source and destination IP and MAC addresses

**IP addresses**
Are the following statements TRUE or FALSE?

1. If a device on network A is moved to network B, its IP address will change.
   A. True
   B. False

2. IP addresses are used to identify a machine on a network and the network to which it is attached.
   A. True
   B. False

3. Each network connected to the Internet has a unique network number.
   A. False
   B. True

4. The network portion of every IP address is assigned by the local network administrator.
   A. True
   B. False
5. How many bits are in an IP address?
   A. 4
   B. 8
   C. 32
   D. 16

6. How many bytes are in an IP address?
   A. 4
   B. 8
   C. 32
   D. 16

7. What is the minimum decimal value in an octet?
   A. 0
   B. 1
   C. 2
   D. 8

8. What is the maximum decimal value in a byte?
   A. 0
   B. 255
   C. 8
   D. FF

9. How many bits are in a byte?
   A. 2
   B. 4
   C. 6
   D. 8

10. How many bytes are in a MAC address?
    A. 2
    B. 4
    C. 6
    D. 8

**Classes of IP addresses**
1. To which class of IP address would the IP address of 197.22.103.221 belong?
   A. class "A"
   B. class "B"
   C. class "C"
   D. class “D”
   E. class “E”
2. Which of the following dotted notations cannot represent an IP address?
   A. 301.188.12.77
   B. 167.78.35.202
   C. 122.31.22.226
   D. 254.254.254.254

3. In a class "A" network using an IP addressing scheme, the first sixteen bits are used for the network part of the address, and the last two octets are reserved for the host part of the address.
   A. True
   B. False

4. To what class of network would the following IP address belong: 144.26.108.15?
   A. Class "A" network
   B. Class "B" network
   C. Class "C" network
   D. Class “D” network

5. To what class of network would the IP address, 18.12.245.10, belong?
   A. Class "A" network
   B. Class "B" network
   C. Class "C" network
   D. Class “D” network

6. In the IP address, 190.233.21.12, how many octets have been assigned by the NIC?
   A. One
   B. Two
   C. Three
   D. Four

7. In the IP address, 88.224.73.201, how many octets could be assigned locally by the network administrator?
   A. One
   B. Two
   C. Three
   D. Four

8. Select the IP address below which would belong to the largest network.
   A. 69.22.214.158
   B. 144.144.144.3
   C. 220.91.144.222
   D. 255.255.255.255
9. Which of the following best describes a class "B" network?
   A. network.network.host.host
   B. network.network.network.host
   C. network.host.host.host
   D. host.network.host.network

10. There are three classes of commercial networks.
    A. False
    B. True

11. IP addresses with numbers 224 through 255 are reserved for multicast and experimental purposes.
    A. True
    B. False

12. A class "C" network address would have all binary 0s in its final octet.
    A. True
    B. False

13. A class "B" network address would have all binary 0s in its final two octets.
    A. True
    B. False

14. Which of the following is an example of a class "C" network address?
    A. 196.25.10.0
    B. 113.0.0.0
    C. 113.22.104.0
    D. 74.255.255.255

15. Which of the following best describes a class “C” network?
    A. network.network.host.host
    B. network.network.network.host
    C. network.host.host.host
    D. host.host.host.network

16. Which of the following best describes a class “A” network?
    A. network.network.host.host
    B. network.network.network.host
    C. network.host.host.host
    D. host.host.host.network

17. Which of the following is a class “C” IP address?
    A. 220.15.64.126
    B. 191.15.64.126
    C. 127.15.64.126
    D. 242.15.64.126
18. Select the IP address for the smallest network.
   A. 220.15.64.126
   B. 191.15.64.126
   C. 127.15.64.126
   D. 242.15.64.126

19. How many octets have been assigned by InterNIC in a class “C” network?
   A. one
   B. two
   C. three
   D. four

20. If you have a class “A” IP address, then how many bytes have been assigned to
    you for your hosts?
   A. one
   B. two
   C. three
   D. four

**Binary to decimal conversions**
1. Which of the following decimal numbers equals the binary number 11111111?
   A. 128
   B. 254
   C. 255
   D. 17

2. How would the IP address 197.15.22.31 be expressed in a binary numbering scheme?
   A. 11000101.00001111.00010110.00011110
   B. 11000101.00001111.00010110.00011111
   C. 11000101.00001111.00010110.00010111
   D. 11000101.00001101.00010110.00011110

3. How would the IP address 197.15.22.127 be expressed in a binary numbering scheme?
   A. 11000101.00001111.00010110.01111111
   B. 11000101.00001111.00010110.01111110
   C. 11000101.00001111.00010110.11111110
   D. 11000101.00001111.00010111.11111110

4. In binary notation, the subnet mask for a Class “B” network may be given as:
   11111111.11111111.11111110.00000000. What would this be in dotted decimal?
   A. 256.256.255.0
   B. 256.255.254.0
   C. 255.255.254.0
   D. 254.254.254.0
5. What would the correct binary sequence be for a subnet range that borrowed three bits?
   A. 111,110,101,100,011,010,001,000
   B. 000,001,011,010,100,110,101,111
   C. 111,101,110,100,010,011,001,000
   D. 000,001,010,011,100,101,110,111

6. What is the binary to decimal conversion for 01010101?
   A. 128
   B. 127
   C. 85
   D. 4

7. What is the binary to decimal conversion for 01111110?
   A. 126
   B. 63
   C. 85
   D. 124

8. What is the binary to decimal conversion for 00010000?
   A. 15
   B. 32
   C. 1
   D. 16

9. What is the binary to decimal conversion for 01100110?
   A. 102
   B. 103
   C. 4
   D. 104

10. What is the binary to decimal conversion for 00001000?
    A. 8
    B. 12
    C. 16
    D. 4

11. What is the decimal to binary conversion for 17?
    A. 01000111
    B. 00010001
    C. 10001001
    D. 11101110
12. What is the decimal to binary conversion for 128?
A. 01000110
B. 01001000
C. 10000000
D. 01111111

13. What is the decimal to binary conversion for 220?
A. 01000111
B. 11010001
C. 00101001
D. 11011100

14. What is the decimal to binary conversion for 240?
A. 11110000
B. 111000001
C. 10111001
D. 11101110

15. What is the decimal to binary conversion for 191?
A. 01000100
B. 10111111
C. 10001001
D. 11101010

Broadcast and subnet addresses
1. Which of the following definitions best describes a “broadcast?”
   A. Data packet that will be sent to all nodes on a network segment.
   B. Section of a network that is bounded by bridges, routers, or switches.
   C. Binary digit used in the binary numbering system that can be 0 or 1.
   D. Screaming at the top of your lungs until you can’t breathe.

2. Which of the following is an example of a class “C” broadcast address?
   A. 190.12.253.255
   B. 190.44.255.255
   C. 221.218.253.255
   D. 221.218.253.0

3. In a class "C" subnet address up to six bits can be borrowed from the host field.
   A. True
   B. False

4. Which of the following is a valid class “B” IP broadcast address with no subnets?
   A. 68.140.74.0
   B. 129.37.0.255
   C. 129.37.0.0
   D. 190.37.255.255
5. Which of the following is reserved for the broadcast address in 198.64.74.x/27?
   A. .0
   B. .127
   C. .192
   D. .254

6. Which of the following is a valid class “C” IP subnet number (/27)?
   A. .191
   B. .127
   C. .128
   D. .129

7. Which of the following is a valid class “B” IP subnet broadcast address?
   A. 10101011.01011101.00010000.01011110
   B. 00101011.01011101.00010000.01111111
   C. 10110110.01011101.00000000.01111111
   D. 11100110.01011101.00000000.01111111

8. Which type of IP address can borrow one bit from the last octet to create subnets?
   A. Class “C” IP addresses
   B. Class “B” IP addresses
   C. None can borrow 1 bit from the last octet
   D. Class A, B, and C can borrow 1 bit from the last octet
   E. Both Class “A” and “B”

9. Which of the following best describes the address 147.30.74.1
   A. Class “A” host address
   B. Class “A” broadcast address
   C. Class “B” host address
   D. Class “B” subnet address

Subnetting possible vs. useable
Are the following statements TRUE or FALSE?
1. Subnet addresses are assigned locally.
   A. False
   B. True

2. Subnet addresses include only a network number and a host number.
   A. True
   B. False

3. Each time the number of bits borrowed from an eight bit octet decreases, the
decimal value representing that octet in the subnet mask increases by a power of two
   A. True
   B. False
4. How many possible subnets can be created if four bits are borrowed from the host field?
   A. 2
   B. 4
   C. 8
   D. 16

5. How many possible subnetworks can be created if five bits are borrowed from the host field?
   A. 5
   B. 8
   C. 16
   D. 32

6. How many possible subnetworks can be created if six are borrowed from the host field?
   A. 6
   B. 12
   C. 32
   D. 64

7. How many actual subnets can be created if four bits are borrowed from the host field? (no ip-subnet zero)
   A. 2
   B. 4
   C. 6
   D. 14
   E. 16

8. How many actual subnetworks can be created if five bits are borrowed from the host field? (no ip-subnet zero)
   A. 15
   B. 20
   C. 25
   D. 30

9. How many possible subnetworks can be created if six are borrowed from the host field?
   A. 6
   B. 16
   C. 62
   D. 64
10. On a class "C" network with three bits borrowed for subnets to which subnetwork would the IP subnet and host range 01100001 belong? (no ip-subnet zero)
   A. second subnet
   B. third subnet
   C. fourth subnet
   D. fifth subnet

11. How would the subnetwork 01100001 field for a Class “C” IP address with six useable subnets be expressed in binary numbers?
   A. 001111
   B. 01111
   C. 0111
   D. 011

12. How would the third useable subnet range of a Class “C” IP address with eight possible subnets be expressed in decimal numbers? (no ip-subnet zero)
   A. 64
   B. 96
   C. 128
   D. 32

13. How would the decimal number 220 be expressed as a binary number written as an octet?
   A. 11011100
   B. 11011101
   C. 01101110
   D. 11101101

14. How would the sixth possible subnetwork field of a Class “C” IP address be expressed in binary numbers?
   A. 100
   B. 101
   C. 110
   D. 111

15. To what subnetwork on a Class “C” network with three bits for a subnet would a fourth octet expressed as 10101101 belong? (no ip-subnet zero)
   A. first
   B. sixth
   C. fifth
   D. seventh
16. How would the host field be expressed in binary numbers of a Class “C” IP address which has 6 useable subnets for host number 13? (no ip-subnet zero)
A. 01101
B. 01100
C. 01110
D. 01111

17. Which of the following best describes the maximum number of bits that can be borrowed in a Class “C” network? (no ip-subnet zero)
A. 6
B. 8
C. 14
D. 12

18. Which of the following best describes the maximum number of bits that can be borrowed in a Class “B” network? (no ip-subnet zero)
A. 14
B. 6
C. 8
D. 4

19. If two bits are borrow from the host field of a Class “C” network, then how many possible subnetworks can be created?
A. 16
B. 4
C. 8
D. 2

20. If four bits are borrowed from the host field of a Class “B” network, then how many subnetworks can be created? (no ip-subnet zero)
A. 16
B. 32
C. 8
D. 4

21. If four bits are borrowed from the host field of a Class "B" network, then how many possible hosts per subnetwork can be created?
A. 256
B. 4096
C. 16
D. 8
22. If two bits are borrowed from the host field of a Class “C” network, then, how many possible hosts per subnetwork can be created?
   A. 2048
   B. 256
   C. 64
   D. 32

23. If we have 4 possible subnets in our network then how many bits have been borrowed from the host field?
   A. 4
   B. 3
   C. 2
   D. 6

24. If we have 4 possible subnets in our network then what will the range of binary host field numbers be for the first subnetwork?
   A. 0000-1111
   B. 00000000-11111111
   C. 000000-111111
   D. 0000-1111

25. If we have 4 possible subnets in our network then what decimal value would be assigned to an octet expressed as 01011011?
   A. .191
   B. .67
   C. .91
   D. .92

26. If we have 2 possible subnets in our network then what would the binary subnetwork field number be for the decimal host number expressed as .196?
   A. 01
   B. 10
   C. 11
   D. 00

27. In a network with two bits borrowed for subnets, what would the binary host field number be for the decimal host number expressed as .49?
   A. 011001
   B. 110001
   C. 00110001
   D. 111001
Subnet masking

1. How would the subnet mask 255.255.255.0 be represented in dotted binary notation?
   A. 11111111.11111111.11111111.00000000
   B. 11111111.11111111.11111111.00000000
   C. 11111111.11111111.11111111.11111111
   D. 11111111.11111111.11111111.10000000

2. If only seven bits are borrowed in a Class “B” network then what would the subnet mask be in dotted decimal notation?
   A. 255.255.255.0
   B. 255.255.254.0
   C. 254.255.255.0
   D. 254.254.254.0

3. What would the subnet mask be in dotted decimal notation if only five bits were borrowed from the third octet in a class “B” address? (no ip-subnet zero)
   A. 255.255.254.0
   B. 255.255.255.0
   C. 255.255.248.0
   D. 254.254.248.0

4. What would the subnet mask be in dotted decimal notation if only one bit were borrowed from the third octet in a Class “A” address? (no ip-subnet zero)
   A. 128.255.128.0
   B. 255.255.255.0
   C. 255.255.128.0
   D. cannot borrow only one bit

5. Subnet masks tell devices which part of an address is the network number including the subnet and which part is the host.
   A. True
   B. False

6. Subnet masks are 16 bits long and are divided into two octets.
   A. False
   B. True

7. Subnet masks have all 0’s in the network and subnetwork portions of their addresses.
   A. False
   B. True
8. Binary bits in the subnet mask are used to represent which of the following:
   A. host bits
   B. subnet bits
   C. network bits
   D. both b and c

9. What will the use of subnets do regarding the amount of broadcast traffic?
   A. decrease, because broadcasts are not forwarded outside
   B. decrease, because it will take less time for a host to get broadcasts from
      the router
   C. increase, because packets are forwarded to all subnets
   D. increase, because bandwidth will decrease

**Router functions**
1. In the graphic below (on the next page), if device A3 is sending data to device C3, out of what port will the router send the data?
   A. A5
   B. C4
   C. C1
   D. A4

2. In the graphic below (on the next page), how many IP addresses does the router have?
   A. 1
   B. 15
   C. 4
   A. 5

3. In the graphic, if device A2 wants to send data to device A4, will the router forward the data to Network B?
   A. Yes
   B. No

4. How many ports does the router in this graphic have?
   A. 8
   B. 4
   C. 1
   D. 5
**Whole enchilada problems**

1. Which of the following is the dotted decimal notation value of the host portion of a Class “A” IP address 38.0.53.228 with a subnet mask of 255.255.252.0?
   A. 0.228
   B. 53.228
   C. 1.228
   D. 5.228

2. Which of the following subnet masks will not be applicable to a Class “C” IP address but can be used with a Class “B” IP address?
   A. 255.255.5.0
   B. 255.255.255.192
   C. 255.255.255.240
   D. 255.255.255.128

3. Which of the following is a valid address for a Class “A” IP address with a subnet mask of 255.255.240.0?
   A. 38.255.240.2
   B. 38.0.192.0
   C. 38.0.240.255
   D. 38.255.255.255

4. Which of the following is a valid Class “B” IP address with a subnet mask of 255.255.255.224?
   A. 18.200.3.55
   B. 130.0.0.1
   C. 154.255.0.31
   D. 147.255.0.48
5. Which of the following is the first available address for a Class “A” IP address of 2.x.x.x. with a subnet mask of 255.255.255.128?
A. 2.1.1.1  
B. 2.0.0.129  
C. 2.1.2.3  
D. 2.0.0.1

6. Which of the following addresses is a valid address when using a subnet mask of 255.255.255.192?
A. 2.0.0.0  
B. 129.1.0.63  
C. 177.255.255.195  
D. 215.1.8.188

Having trouble with the “whole enchiladas?” Hint: Look to eliminate any addresses where subnet portion or host portions contain all zeros or all ones.

Hungry for more???

You are almost there for the CompTIA Network+ and Security+ Certifications too! For more information:

http://www.comptia.org/certification/default.aspx

Building successful careers for students everywhere…
Network Design with Subnets

Objective:
To learn how to design networks from “essay” type information.

Background:
In this lab you will be presented with a variety of networking scenarios. For each you are to design the networks, subnets, and IP addresses. Each one here will be progressively more difficult. Do not become upset if you have trouble with this…sometimes it takes doing this many times before some people “get it.” It’s actually like getting struck by lightning. After many times of not getting it you feel like lightning knocks you out of your chair and you suddenly get it. So let’s keep hammering the examples so everyone can get it…after all we learn by doing. There are many different ways that these can be done…so the answers I give are not necessarily the only answers.

Real Estate Office
You are working as an independent consultant for a real estate broker. He has 16 agents and one receptionist working for him. There are three printers and one file server in the office. He wants to have Internet access and email accounts for everyone with a DSL line. Please design him a network for the least amount of money possible. Those small businesses typically do not have a lot of money. Don’t forget to include your expenses (figure $150 an hour for installation and setup).

Veterinarian’s Office
Your cousin is a vet in the Jacksonville, Florida area. He has asked you to help design and set up a network for him as inexpensively as possible. (Since it’s for family you are doing it for free). He has a main office in Mandarin where he spends 5 days (all but Wednesday) with his receptionist (who does scheduling on the database server), an office manager (who does accounting, billing, etc on the database server), and his office computer (where he keeps all his medical stuff). He also has a dot matrix and a laser jet printer there. He would like to connect to the Internet with a DSL line and have dial-in access to his home computer. His office in St. Augustine (open only on Wednesdays) will have a computer for the doctor and for the receptionist. They need to have access to the database server at the main office (use dial-in via the PSTN). There is a laser jet at the St. Augustine office.

ABC Packaging Company—Part 1
You are working as the network administrator for ABC Packaging Company in Tarpon Springs. You are to design a network that focuses upon scalability and adaptability. There are five departments: Administration (14 people, 5 printers), Engineering (22 people, 5 printers, 1 file server), Production (5 people), Accounting (11 people, 4 printers, 1 database and file server), and Sales/Marketing (11 people, 4 printers, 1 file server). Each department will require a separate subnet. The servers will have their own subnet. Be sure to connect them to the Internet with a T-1 line.
Website Company
You are the network administrator for an upstart website publishing company. They have offices in two adjacent buildings on different floors. Lately, they have realized the costs of their individual Internet accounts far exceeds the costs of installing and maintaining a T-1 line. As the network guru you are to design a network that will utilize FDDI between the buildings. The west building uses floors 3, 4, and 5 for the sales and admin staff. Here you will want to use a CISCO Catalyst 5000 with a FDDI module, a management module, and a 24-port switch module. From there each floor will distribute access via a CISCO 1924 switch to each of its 20 nodes (workstations, servers, and printers). The east building uses floors 1 through 5 for the design and engineering staff. Here you will want to use a CISCO Catalyst 5500 with a FDDI module, a management module, and a 24-port switch module. You will also have a CISCO 2610 router with T-1 module, and a Kentrox CSU/DSU for your full T-1 line. Your ISP, ComBase has sold you two blocks of 62 IP addresses: 198.74.56.x (1-62) and (65-126). Combase will also provide the DNS services, unlike most ISP’s where more than 24 IP’s are ordered. Design your network, including cabling and grounds, to include all IP’s, subnet masks, gateways, and anything else you need to include.

This space for rent.

Send an email today to Basham.Matt@spcollege.edu to find out how your company can advertise in this space!

Hitting a million targeted readers provides good value for your advertising dollars.
Quickie Subnetting

When borrowing three bits

<table>
<thead>
<tr>
<th>Subnet number</th>
<th>1st available host IP</th>
<th>Last available host IP</th>
<th>Broadcast Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td><strong>30</strong></td>
<td>31</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>62</td>
<td>63</td>
</tr>
<tr>
<td>64</td>
<td>65</td>
<td>94</td>
<td>95</td>
</tr>
<tr>
<td>96</td>
<td>97</td>
<td>126</td>
<td>127</td>
</tr>
<tr>
<td>128</td>
<td>129</td>
<td>158</td>
<td>159</td>
</tr>
<tr>
<td>160</td>
<td>161</td>
<td>190</td>
<td>191</td>
</tr>
<tr>
<td>192</td>
<td>193</td>
<td>222</td>
<td>223</td>
</tr>
<tr>
<td>224</td>
<td><strong>225</strong></td>
<td><strong>254</strong></td>
<td>255</td>
</tr>
</tbody>
</table>

*Italicized*—useable only with the `ip subnet-zero` command

When borrowing three bits the wildcard mask becomes “31”

When borrowing four bits

<table>
<thead>
<tr>
<th>Subnet number</th>
<th>1st available host IP</th>
<th>Last available host IP</th>
<th>Broadcast Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td><strong>14</strong></td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td>48</td>
<td>49</td>
<td>62</td>
<td>63</td>
</tr>
<tr>
<td>64</td>
<td>65</td>
<td>78</td>
<td>79</td>
</tr>
<tr>
<td>80</td>
<td>81</td>
<td>94</td>
<td>95</td>
</tr>
<tr>
<td>96</td>
<td>97</td>
<td>110</td>
<td>111</td>
</tr>
<tr>
<td>112</td>
<td>113</td>
<td>126</td>
<td>127</td>
</tr>
<tr>
<td>128</td>
<td>129</td>
<td>142</td>
<td>143</td>
</tr>
<tr>
<td>144</td>
<td>145</td>
<td>158</td>
<td>159</td>
</tr>
<tr>
<td>160</td>
<td>161</td>
<td>174</td>
<td>175</td>
</tr>
<tr>
<td>176</td>
<td>177</td>
<td>190</td>
<td>191</td>
</tr>
<tr>
<td>192</td>
<td>193</td>
<td>206</td>
<td>207</td>
</tr>
<tr>
<td>208</td>
<td>209</td>
<td>222</td>
<td>223</td>
</tr>
<tr>
<td>224</td>
<td>225</td>
<td>238</td>
<td>239</td>
</tr>
<tr>
<td>240</td>
<td><strong>241</strong></td>
<td><strong>254</strong></td>
<td>255</td>
</tr>
</tbody>
</table>

*Italicized*—useable only with the `ip subnet-zero` command

When borrowing four bits the wildcard mask becomes “15”
Objective:
To use your subnet knowledge to design an IP addressing scheme for the John’s Brewhouse Restaurant Network.

Tools and Materials:
Paper and pencil

Background:
John Harvard’s Brewhouse is a microbrewery/restaurant chain in New England. They have locations in Cambridge (MA), Framingham (MA), Wayne (PA), Springfield (PA), Pittsburgh (PA), Manchester (CT), Wilmington (DE), Providence (RI), Lake Grove (NY), and Washington DC. Three network topologies are provided here. You task is to design an IP addressing scheme that will address all current needs as well as future expandability. If you see anything that may want to address feel free to note it. Scalability, adaptability, reliability and performance are the key issues in this design. You will be using private addressing in your network. All lines are 10BaseT unless noted.

Lab Design:
Typical Restaurant:
Restaurant Consulting Services (RCS) Danvers, Mass.

http://www.networkcomputing.com/1005/1005centerfoldtext.html

The “Script Kiddie Cookbook” Available Mid-August 2004 at
http://www.lulu.com/learningbydoing

This book title is based upon the Anarchist Cookbook that came out in the 1980’s. In that tremendously popular book it showed you how to build bombs at home. Similarly the Script Kiddie Cookbook will show you step by step hacking and hacking methodology. You will go to jail if you use this information improperly!!!

Learn the basics about computer security with this book. This book assumes you know nothing about security and starts there. All tools and software used are freeware. Other books will build upon the materials in this book. Learn about the nuts and bolts of SPAM, port scanning, Knoppix tools, cookies, and bring it all together with exercises in hacking at legal gaming sites.

GET THE BOOK THE GOVERNMENT DOES NOT WANT YOU TO OWN!!!
Paper Lab: Variable Length Subnet Masking (VLSM)

Objective:
To learn how to implement VLSM in subnet design.

Background:
When designing networks it is preferable to be as efficient as possible when assigning IP addresses. As we have seen in previous labs sometimes we even need to use contiguous (sequential) numbers for our subnet schemes. As your skills in networking and networking design increase you will need to know how to efficiently utilize VLSM (RFC 1219).

Tools and Materials:
Paper and pencils
Super VLSM chart (I used to have a link to this one really cool site: http://www.henninger.net/downloads/ccna/tools/subnettable.pdf
But the guy may have went crazy insano and put a password access to it…oh well it was outdated anyway. Maybe you could send Dale an email to thank him…I guess along with your $4,000 for the CCNA classes you get to use “the site.”) SO, my new one is at http://www.spjc.edu/star/cisco/Matt/New%20Subnetting%20VLSM%20tables.xls or http://www.spjc.edu/star/cisco/Matt/list_of_current_papers_and_brief.htm and look for it. It looks nasty in the Excel spreadsheet but it prints nicely on one page.

Lab Diagram:

Problems:
For the network diagrammed design an IP addressing scheme using VLSM to be as efficient as possible with IP address distribution.

1. You have been assigned the class “C” private IP address by the upper-level IT staff. Other divisions have other Class “C” IP addresses. For now, you only need to know you have the 192.168.70.0/24 network to design.
2. You have been assigned the class “B” private IP address by the upper-level IT staff. Other divisions have other Class “B” IP addresses. For now, you only need to know you have the 172.168.128.0/18 network to design.
3. You have been assigned the class “A” private IP address by the upper-level IT staff. Other divisions have other Class “A” IP addresses. For now, you only need to know you have the 10.16.0.0/12 network to design.

**Supplemental Lab or Challenge Activity:**
If the router to HQ was used for DHCP could you set this network up with RIP and make it work? Try it.

Let’s go through one example using the above network design and a class “C” network address given as 212.14.17.x/24.

1. Determine largest network needed: 57 IP’s. This will fit into a network in our first column (62 hosts max). So we put down 212.14.17.64/26 for that network and color out the ip address ranges from .64 to .124 on our chart (all the way across the chart). Our actual usable addresses are .65 to .126...the columns all the way on the left are not that specific.

2. Determine the next largest network needed: 39 IP’s. This will fit into a network in our first column (62 hosts max). So we put down 212.14.17.128/26 for that network and color out the ip address ranges from .128 to .188 on our chart (all the way across the chart). Our actual usable addresses are .129-.190.

3. Determine the next largest network needed: 28 IP’s. This will fit into a network in our second column (30 hosts max). So we put down 212.14.17.32/27 for that network and color out the ip address ranges from .32 to .60 on our chart (all the way across the chart). Our actual usable addresses are .33-.62.

4. Determine the next largest network needed: 24 IP’s. This will fit into a network in our second column (30 hosts max). So we put down 212.14.17.192/27 for that network and color out the ip address ranges from .192 to .220 on our chart (all the way across the chart). Our actual usable addresses are .193-.222.

5. Determine the next largest network needed: 14 IP’s. This will fit into a network in our third column (14 hosts max). So we put down 212.14.17.16/28 for that network and color out the ip address ranges from .16 to .28 on our chart (all the way across the chart). Our actual usable addresses are .17-.30.

6. Determine the next largest network needed: 12 IP’s. This will fit into a network in our third column (14 hosts max). So we put down 212.14.17.224/28 for that network and color out the ip address ranges from .224 to .236 on our chart (all the way across the chart). Our actual usable addresses are .225-.238.

7. Determine the next largest network needed: 6 IP’s. This will fit into a network in our fourth column (6 hosts max). So we put down 212.14.17.8/29 for that network and color out the ip address ranges from .8 to .12 on our chart (all the way across the chart). Our actual usable addresses are .9-.14.

8. Determine the next largest network needed: 2 IP’s. This will fit into a network in our fifth column (2 hosts max). So we put down 212.14.17.4/30 for that network and color out the ip address ranges from .4 to .8 on our chart (all the way across the chart). Our actual usable addresses are .5-.6.

9. Don’t forget about those serial lines between our routers! They need subnets with IP’s too. For those we picked, basically what is left. 212.14.17.240/30 (useable .241-
.242), 212.14.17.244/30 (useable .245-.246), and 212.14.17.248/30 (useable .249-.250).

These are the addresses for this lab...can you “see” the variable length subnet mask?

- 212.14.17.x/24
- 212.14.17.64/26
- 212.14.17.128/26
- 212.14.17.32/27
- 212.14.17.192/27
- 212.14.17.16/28
- 212.14.17.224/28
- 212.14.17.8/29
- 212.14.17.128/29
- 212.14.17.240/30
- 212.14.17.16/28
- 212.14.17.224/28

So What Have I Learned Here?
In this lab you learned about VLSM. This is a topic in the CCNP classes. So why did I put it here? Simple, I have seen it on the CCNA test AND it makes sense. I have no idea why it is introduced in the CCNP stuff and not here. It makes more sense as an extension to subnetting. We learned about discontiguous routes and classful boundaries earlier. Now, with your knowledge that RIP does not pass subnet mask information you can make an intelligent decision not to use VLSM if you are using RIP. See how it all starts to come together? Let’s look at the difference between static and dynamic routing. You have already been doing dynamic routing with the router rip command.

Looking for college courses? We deliver.

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education.

Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

St. Petersburg College’s eCampus

http://e.spcollege.edu

Your time. Your place. Your future.
What’s Wrong with these Subnets?

The objective of these labs is to give you some hands-on experience in troubleshooting subnetting problems before you even start putting these into the routers. So I put in a bunch of them here with some of the more common faults. Being able to do these well will help you with a big chunk of the test items and make your hands-on labs go more smoothly.

Subnetting example #1

192.168.1.1/24  192.168.1.3/24
192.168.1.2/24          192.168.1.4/24
192.168.1.5/24 192.168.1.6/24

So what is wrong with this and what are the possible fixes? For the first couple I will fill in the answers…after that you will have to check with your answer guide or instructor. This one has a few different answers. First, there are no subnets used here. There is one big flat subnet. The routers will not know how to route their information. They use the subnets to know which interface to send information out. One possible solution is to change all the ip’s on the serial lines to usable ip’s with 30 bit masks (ie. 1-2, 5-6, 9-10). Or, we could switch subnets altogether (ie. 192.168.1.0, 192.168.2.0, and 192.168.3.0 with a 24 bit mask).

Subnetting example #2

192.168.1.1/30  192.168.1.5/30
192.168.1.2/30          192.168.1.6/30
192.168.1.9/24 192.168.1.10/24

So what is wrong with this and what are the possible fixes? Possible solution #1: switch the mask on the bottom serial line to 30-bit masks. Possible solution #2: switch the ip’s to a different subnet (ie. 192.168.2.0/24). Possible solution #3: switch the ip’s to a different subnet within the subnet range with the remain usable ip numbers (on the bottom serial line switch them to 192.168.1.253/24 and 192.168.1.254/24).
Subnetting example #3

172.16.1.1/30               172.16.1.5/30

172.16.1.2/30                  172.16.1.6/30

172.16.1.9/24  172.16.1.10/24

So what is wrong with this and what are the possible fixes?

Subnetting example #4

172.16.1.1/16               192.168.1.1/24

172.16.1.2/16                       192.168.1.2/24

10.0.0.1/8  10.0.0.2/8

So what is wrong with this and what are the possible fixes?

Subnetting example #5

172.16.8.1/16               192.168.10.1/30

172.16.18.2/16          192.168.10.2/30

10.0.0.6/30  10.0.0.7/30

So what is wrong with this and what are the possible fixes?
Subnetting example #6

192.168.1.1/28               192.168.1.5/29
192.168.1.2/30          192.168.1.6/29
192.168.1.22/30 192.168.1.23/30

So what is wrong with this and what are the possible fixes?

Subnetting example #7

10.0.0.94/28               192.168.10.1/30
10.0.0.95/28                  192.168.10.2/30
10.0.0.81/30  10.0.0.82/30

So what is wrong with this and what are the possible fixes?

Subnetting example #8

222.13.14.15/26               192.168.10.1/30
212.13.14.14/26          192.168.10.2/30
10.255.0.254/30 10.255.0.253/30

So what is wrong with this and what are the possible fixes?
They said “I was crazy!”

How do you like the idea of free and low-cost textbooks? Well here is how you can help me in the effort to bring more of these types of books to you. They say there is safety in numbers, well I can promote this type of textbook production way better if I hear from you personally. Send me an email at Basham.Matt@spcollege.edu just to tell me you are using the book, you like the idea of cheaper high-quality books, or to just say “hello.” Together we will change education for the better!
Switch Maintenance

Objective:
In this lab you will learn the basics of switch maintenance including telnetting/using a web browser to console into a switch, resetting a switch and password recovery on a switch.

Tools and Materials:
(1) workstation
(1) console cable
(1) switch (I used a 1900 here)
(1) straight through cable

Lab Design:

Lab Design:

192.168.1.1/24

192.168.1.2/24

192.168.1.1 gw

Step-By-Step Instructions:
Each of these topics are really too small for an individual lab so I lumped them all together in this one. Before we can do these first two we need an IP address, mask, and gateway on the workstation and an IP address and mask on the switch. To set up the switch from the main menu select:

1. [I] IP configuration
2. [I] IP address
   a. 192.168.1.1
3. [S] Subnet mask
   a. 255.255.255.0
4. Try to telnet into the switch from the workstation.
5. Then, like our routers, we need a password in order to be able to telnet into this device:
   a. [X] Exit to previous menu
6. [M] Menus
7. [C] Console Settings
8. [M] Modify password  (Different switches and IOS’s have different menus)
   a. cisco
   b. cisco
   c. enter
Telnetting/using a web browser to console into a switch:

1. Without an IP address and subnet mask you cannot telnet into a switch. If you have put one on it then just start telnet and use the ip address with the telnet port.

```
C:\Documents and Settings>telnet
Microsoft Telnet>open 192.168.1.1
```

After only a couple of seconds you should see something like this (It will be a DOS prompt…I saved my screen shot from Win 98…the information will be the same)

Look at the picture on the right:

Notice how you no longer have the IP configuration option available.

2. Guess what? You can also get to your switch over the web. Just type that IP address in a web page and see what happens. It’s really cool with pictures and everything. You should see something like the picture above on the right:

Remember how we just put in a password? Yup…we use it only…no user name required.

3. After putting in the password and clicking on “ok” you should see:
So how cool is that? You cannot tell from this picture but you can actually “see” if a port is active…nice when you are not in front of the switch. You can click on the port and view the statistics or even make changes.

4. But wait…there is more. You can also access the switch through the web browser. Scroll down and click on Fast etherchannel management and there will be a hyperlink for “telnet.” This will actually bring up a hyperterminal session to the switch. You will see this (notice how there are only two choices now in the User Interface Menu):

![HyperTerminal Session](image)

**Resetting a switch:**

1. Resetting a switch is really simple. First start by selecting [M] for menus.
2. Then select [S] for system management.
3. Select [F] for reset to factory defaults.
4. Select [yes].
5. Then select [R] for reload.
6. Select [yes] and watch the switch reload. Its just that simple!

**Password recovery:**

1. You thought the last one was easy? Heck…this is the easiest password recovery you will ever do. Just unplug the switch (its ok…no matter what the configuration is saved…it’s not like a router where you have to do a copy to save the config…sounds like a good test question).
2. When the switch reboots just watch the hyperterminal screen. During the boot it will ask you if you want to reset the password like this:
Just click on “yes” to clear the passwords or ignore the message altogether to keep the current ones in use. Most people miss it because they are too busy watching all the blinking lights, talking with someone, or off getting their Dew.

Supplemental Lab or Challenge Activity:
1. Try doing these labs (this one and the ones to follow) using the command line interface. Some people have seen questions related to this on tests or on practice test CD-roms.
2. Try setting up usernames with passwords for telnet access with your switch.
3. Try repeating this lab with a 2900, 2950, or 3550 switch.

So What Have I Learned Here?
In this lab you learned about some miscellaneous, yet nifty, features about switches and maintaining switches. In the next lab you will start learning about the Spanning Tree Protocol.

This space for rent.

Send an email today to Basham.Matt@spcollege.edu to find out how your company can advertise in this space!

Hitting a million targeted readers provides good value for your advertising dollars.
Basic STP

Objective:
To learn how to construct and understand Spanning Tree Protocol (STP) connections, to view and understand spanning tree states with a protocol inspector, and to construct and configure redundant backbones between switches.

Tools and Materials:
Three (3) cross-over cables
Three CISCO switches (1900 series)
Two (2) straight-through cables
Two Windows PC workstations with Hyperterminal and Ethereal installed

Lab Diagram:

Background:
The main function of the Spanning-Tree Protocol (STP) is to allow us to set up redundant back up lines in case of emergency between switches. When a main line between two of the switches becomes dysfunctional the switch, through its STP states (Blocking, Listening, Learning, Forwarding, Disabled), implements the Spanning Tree Algorithm (STA) when a “link down” is detected. By default the switch checks the condition of its ports every 30 seconds. In other words, when a main line goes down, the redundant backbone should come up within 30 seconds (although sometimes it takes up to about 60 seconds with default settings). STP is implemented on switches, by default, for VLANs 1-64. This means all you have to do is plug in your redundant backbone (a cross over cable) into any available port between switches because all switches in their default state have all ports assigned to VLAN 1.

The switch uses priorities to determine which lines are the main lines and which are the redundant backbones. The values can be 0 through 255. The lower number has the higher priority (the main lines). By default each 10BaseT port is assigned a priority of 128 and each 100BaseT port is assigned a priority of 10. On our 1900 series switches this means that the Ax and Bx ports will be selected as main backup lines before ones using the numbered (1-12 or 1-24) ports. In practice, we use the Ax and Bx lines to set our “Trunks” or backbone lines. Since the Ax and Bx lines are typically used for high speed this works best. In the next lab you will be configuring the backbone lines by changing the settings (cost, priority, etc) on each port to determine statically which will be the main backbones and which will be the redundant backbones.
Step-By-Step Instructions:
1. You should set each switch back to its factory default settings. The power should be turned off when you are finished re-setting.

Test default Spanning Tree Settings:
1. Make sure the power is turned off on all of the switches. For ease, place each switch on top of each other. For this lab, the top switch will be called “SW-A,” the middle switch will be called “SW-B,” and the bottom switch will be called “SW-C.”
2. Plug one end of a crossover cable into port “Ax” on SW-A and the other end into port “Bx” on SW-B.
3. Plug one end of a crossover cable into port “Ax” on SW-B and the other end into port “Bx” on SW-C.
4. Plug one end of a crossover cable into port “Ax” on SW-C and the other end into port “Bx” on SW-A. You have now created a loop in your switches.
5. Turn on the power. After the switches cycle through their start-up procedures one by one the lights over the Ax and Bx ports should change from amber-colored (Problem or not functioning) to green-colored (OK-operational). One of the lights should change back to amber. This line was chosen to be the redundant backbone because all priorities are equal in default mode.
6. Let’s test the backup line. Unplug any one of the cables that appears with green lights on both ends. In about 60 seconds or so the redundant backbone line amber light will turn green. This indicates the switch is going through the five STP states.
7. Plug the back up line back in…it will return back to its original state in only a couple of seconds.

Test the ability to ping from (PC)-to (switch)-to (switch)-to (switch)-to (PC):
1. Connect a PC workstation (PC-A) to SW-A using a straight-through cable.
2. Change the TCP/IP settings to IP: 192.168.1.1 and S/M 255.255.255.0.
3. Connect a PC workstation (PC-B) to SW-B using a straight-through cable.
4. Change the TCP/IP settings to IP: 192.168.1.2 and S/M 255.255.255.0.
5. Test the connectivity from PC-A to PC-B by pinging. This should be successful.
6. Start an Ethereal capture on workstation “B.”
7. Let’s test the backup line. Unplug any one of the cables that appears with green lights on both ends.
8. WHILE THE LIGHT IS STILL AMBER—test the connectivity from PC-A to PC-B by pinging. It should not work.
9. Within 60 seconds the redundant backbone line amber light will turn green.
10. Test the connectivity from PC-A to PC-B again. This should be successful again.
11. Stop the capture. Let’s see what we have in figure 1.

Keep track of updates and changes at http://www.spcollege.edu/star/cisco Scroll to the bottom of the page and click on the “Lab Manual Edits.”
Manually select main and redundant backbones:

1. Plug one end of a crossover cable into port “Ax” on SW-A and the other end into port “Bx” on SW-B.
2. Plug one end of a crossover cable into port “Ax” on SW-B and the other end into port “Bx” on SW-C.
3. Start an Ethereal capture on workstation “B.”
4. Plug one end of a crossover cable into port “18” on SW-C and the other end into port “18” on SW-A. You have now created a loop in your switches. The cables in the Ax and Bx ports will have priorities of 10 (since they are 100BaseT by default) and the #18 ports will have priorities of 128. The higher priority cables will have the lower priority numbers. Do not use the Ax or Bx for either end of the cable.
5. The light over the #18 ports on one end should be green and amber on the other. This line was chosen to be the redundant backbone because of its manually static priority setting in the default mode was a higher priority number (and therefore the last one to be enabled in this scenario). Stop the capture and let’s see our STP state with a cost of 10. See figure 2.
6. We are looking at one with a cost of 110 because the 100 is added to the 10 for a total cost between two devices. Our “pure” cost for that line is 10.

7. Let’s test the backup line. Unplug any one of the Ax/Bx cables that appears with green lights on both ends. Within 60 seconds the redundant backbone line amber light will turn green. This indicates the switch is going through the five STP states. Repeat steps 2-4 to return cabling to their original settings.

Supplemental Activity or Challenge Lab:
1. Try doing this lab with as many switches as you can. Sounds silly but it can be tricky.
2. Start a ping storm by using many very large icmp packets. See what this does to your network performance and the time it takes for STP to bring up backup lines.
   Geeze…you thought it took long before.

So What Have I Learned Here?
To set up redundant lines between switches we just need to know which ports to use for best service. It really doesn’t matter which ones we use but certain ones are more preferred to others. In the next lab we will change settings.
Basic STP with One Router

**Objective:**
To learn how to add a router into a switched network using a redundant backup line with STP.

**Tools and Materials:**
(2) workstations
(4) straight through cables (st)
(2) console cables
(1) Cross over cable (xo)
(2) 1900 series switches
(1) 2500/2600 series router

**Lab Diagram:**

**Step-By-Step Instructions:**
1. Cable the lab as shown. Have your instructor set up the router for you. Ok. Now the fun starts. Use the 83.x.x.x network with a 16-bit mask. Oh don’t get complacent with the easy numbers. Pick your own routing protocol to use.
2. Ping from workstation “A” to “B.” Ping from each workstation to the loopback adapter. Use trace route for all three pings to verify the paths.
3. Use “sh ip route” to verify routes on the router.
4. Use debug stp on the router to see the changes in stp states over the network. Take one of the main lines down and view the router messages.
5. Repeat steps 2-3 again with the main line down.

**So What Have I Learned Here?**
How to add a router into a switched network using back up lines and STP. In the next lab you will work with the “metrics” with STP for selecting back up lines. *Yeah, I set you up to fail in this lab, but for a very good reason, sorry.*
Intermediate STP

Objective:
To be able to understand STP states, cost parameters, root bridges, priorities, ports and port fast mode.

Tools and Materials:
(4) switches
(4) cross over cables
(2) straight through cables
(2) console cables
(2) workstations

Lab Design:

Background:
In the last lab we learned about basic STP construction. We learned Spanning-tree frames called bridge protocol data units (BPDU’s) are sent and received by all switches in the network at regular intervals (usually every 2 seconds) and are used to determine the spanning tree topology. STP is implemented on switches, by default, for VLANs 1-64. This means all you have to do is plug in your redundant backbone into any available port. There are five states for every switch port:
1. Blocking—port does not participate in frame-forwarding; port does not learn new addresses
2. Listening—same as blocking, but switch is actively trying to bring the port into the forwarding state; the port does not learn new addresses
3. Learning—port does not participate in frame-forwarding; port does learn new addresses; the switch is trying to change the port to frame-forwarding
4. Forwarding—port does participate in frame forwarding; port does learn new addresses
5. Disabled—port is removed from operation; administrative intervention is required to enable the port
For each port, there are five parameters that may be changed for each port. Each of these affects which port connections are utilized as the main backbones and which are the redundant backbones:

1. **State**—Blocking, Listening, Learning, Forwarding, Disabled
2. **Forward Transitions**—number of times STP changes forwarding states. This number increases when STP detects network loops
3. **Path Cost**—inversely proportional to LAN speed; path costs range from 1 to 65,535—lower number means higher speed connection; default is 100.
4. **Priority**—ranges from 0 to 255 (used in basic lab); 10BaseT priority is 128; 100BaseT priority is 10
5. **Port Fast Mode**—using this will accelerate the time it takes to bring a port into the forwarding state from blocking; Use Port Fast-Mode enabling on ports only for end station attachments; default for 10BaseT is enabled; default for 100BaseT is disabled; by default STP discovery is 30 seconds (don’t confuse this with BPDU’s every 2 seconds)

With all switches reset to their factory defaults how do you think one backbone takes priority over the others if we use all 100BaseT connections? If all costs are equal, then the switch uses the MAC addresses to determine which ones will be the main and which ones will be the backup (redundant) lines.

There are three steps involved in the Spanning Tree process: (1) Electing a root bridge, (2) electing root ports, and (3) electing designated ports.

The root bridge is the bridge from which all other paths are decided. Only one switch can be the root bridge. The selection process uses the lowest bridge priority number first and then uses the lowest bridge ID number (the MAC address). The switches use the BPDU’s to elect a root bridge. When a switch first powers up, it will assume the role of root bridge until it is told otherwise. The default setting for CISCO 1900 series switches is 32768.

Next the switches will search for any redundant paths or loops using BPDU’s. An election of main and backup paths is made using costs. By default, port cost is usually based upon bandwidth (as we saw in the basic lab). The port with the lowest root path cost will be elected as the root port/path. Any time a switch has a direct connection to the root switch it will serve as the root port, regardless of path cost.

The designated port is the port that is advertising the lowest costs to the root bridge. When all three steps are complete the Spanning Tree is finished being set up.

For this lab we will use private IP addressing with one subnet. You can use mixed subnet addresses but only by activating more complicated settings on the switches and/or using routers. Using different subnets will not allow you to ping with this topology.

**Step-By-Step Instructions:**
You should set each switch back to its factory default settings. The power should be turned off when you are finished re-setting.

**Calculate and identify root bridge and main and redundant backbones:**
1. Now then…this is a bit different than our three-switch configuration in the last lab. In that lab no matter which line was disconnected, each line still had a direct
connection to the root switch. That is why we have added a fourth switch to this lab. Now each switch will not have a direct connection so we will have to do some research first. At this point no changes have been made to our switches (ie. we are still set to factory defaults). Turn on each switch (make sure there are no cable connections to any switch). Put a console cable from the switch console port into your PC workstation.

2. Start hyperterminal (9600-8-N-1). Follow these choices: (1) select [I] for IP configuration or (2) select [M] for menus, [N] for network management, [I] for IP configuration, and then write down the MAC address of the switch (it will appear as “Ethernet address”):

<table>
<thead>
<tr>
<th>Switch</th>
<th>MAC Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW-A</td>
<td><strong><strong>-</strong></strong>-<strong><strong>-</strong></strong>-<strong><strong>-</strong></strong></td>
</tr>
<tr>
<td>SW-B</td>
<td><strong><strong>-</strong></strong>-<strong><strong>-</strong></strong>-<strong><strong>-</strong></strong></td>
</tr>
<tr>
<td>SW-C</td>
<td><strong><strong>-</strong></strong>-<strong><strong>-</strong></strong>-<strong><strong>-</strong></strong></td>
</tr>
<tr>
<td>SW-D</td>
<td><strong><strong>-</strong></strong>-<strong><strong>-</strong></strong>-<strong><strong>-</strong></strong></td>
</tr>
</tbody>
</table>

***Don’t forget to move the console cable to the console port of each switch. Right now you cannot telnet into each switch easily. It is quicker just to move the console cable.***

3. From these MAC addresses you should be able to determine which switch by default will be the root bridge. Calculate which crossover cable will be selected as the backup line from their MAC addresses. Circle lowest MAC address as 1st, next to lowest as 2nd, etc.

<table>
<thead>
<tr>
<th>Switch</th>
<th>Root Bridge</th>
<th>Backup Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW-A</td>
<td>Ax - Bx</td>
<td>1st  2nd   3rd  4th</td>
</tr>
<tr>
<td>SW-B</td>
<td>Ax - Bx</td>
<td>1st  2nd   3rd  4th</td>
</tr>
<tr>
<td>SW-C</td>
<td>Ax - Bx</td>
<td>1st  2nd   3rd  4th</td>
</tr>
<tr>
<td>SW-D</td>
<td>Ax - Bx</td>
<td>1st  2nd   3rd  4th</td>
</tr>
</tbody>
</table>

The backup line will be the line between the highest two MAC addresses (3rd and 4th). (The light on Ax for 3rd will be amber).

4. Turn off the power to the switches and remove the console cable.

5. Plug one end of a crossover cable into port “Ax” on SW-A and the other end into port “Bx” on SW-B.

6. Plug one end of a crossover cable into port “Ax” on SW-B and the other end into port “Bx” on SW-C.

7. Plug one end of a crossover cable into port “Ax” on SW-C and the other end into port “Bx” on SW-D.
8. Plug one end of a crossover cable into port “Ax” on SW-D and the other end into port “Bx” on SW-A. You have now created a loop in your switches.

9. Turn on the power. After the switches cycle through their start-up procedures one by one the lights over the Ax and Bx ports should change from amber-colored (Problem or not functioning) to green-colored (OK-operational). One of the lights should change back to amber. Were you right? Remember different groups on different groups of switches will have different answers…it all depends upon the MAC addresses.

**Manual selection of main and redundant backbones by changing port costs and priorities**

1. Disconnect the backbone cable that is not connected to the root bridge and is not selected as the redundant backbone.

   ![Diagram of switches with cable connections]

   - SW-A: Ax → Bx (Root bridge, lowest MAC address)
   - SW-B: Ax → Bx (unplug this one)
   - SW-C: Ax → Bx
   - SW-D: Ax → Bx (redundant backbone darkened)

   If your lab setting appears like the above drawing, then select the line between SW-B (Ax) and SW-C (Bx) to be disconnected. All remaining lights should be green. Didn’t work? Don’t worry theoretically it should…we found it working about 60% of the time…you will see lots of this and how reality doesn’t always agree with theory, especially on the Cisco exams.

2. Switch the crossover cable which you just disconnected to any two ports on SW-C and SW-D (let’s just use port #7 on each). Note: this will vary dependent upon which one is the root bridge. This line should become a redundant backup, mostly because of the lower priority for the slower speed (10BaseT instead of 100BaseT). This line will now become the redundant backbone. We just forced it to be by using our knowledge of default port priority settings. (Just like we did in the last lab).

3. Reconnect that cable back into the Ax and Bx ports.

4. Remove one of the main crossover cables that is attached to the root bridge (like the one between SW-A (Ax) and SW-B (Bx) above).

5. Give it about 60 seconds for the STP to switch the redundant backbone to a main backbone.

6. Connect that crossover cable to ports #7 on SW-A and SW-B. This should reconfigure as the new redundant backbone because of the lower port priority of 10BaseT connections. It should change back almost immediately.

7. Now let’s go in and change the port costs for these ports. Put the console cable into the switch with the amber light of the redundant backbone line. Use [M] menus, [P] port configuration, [select port number 7], and then [C] cost. Change this value to 1. When you hit enter you should almost immediately see the line change from amber to green (from backup to main). The line with the next lowest priority will become the
redundant backup line. If you change the end of the line at the port where you changed the priority (for example from port 7 to port 5) the line will become a redundant backbone again.

8. Change the cost of port 7 back to 100 and return the line back to the Ax-Bx ports.
9. Repeat if needed on the Ax-Bx ports.

Supplemental Lab or Challenge Activity:
1. Use your protocol inspector to capture and view STP packets with your changes.

_So What Did I Learn Here?_
Now you can manually configure backbones between switches and automatically set priorities for backbone selection using the port configuration menu and costs. Just remember this is dependent upon the MAC addresses, with all other factors set to default. This lab also does not work well with three switches because each line will still be connected to the root bridge. To work well you really need at least 4 switches for this lab. In the next couple of labs you will be adding routers to this “flat-switching” network.

---

**Hungry for more???

You are **almost there** for the _CompTIA Network+_ and _Security+_ Certifications too!
For more information:

[http://www.comptia.org/certification/default.aspx](http://www.comptia.org/certification/default.aspx)

**Building successful careers for students everywhere…**
Basic VLAN

**Objective:**
To learn how to construct and understand how to use basic Virtual LAN’s in a network.

**Tools and Materials:**
(1) CISCO switch (1900 series)  
(2) straight-through cables  
(2) Windows PC workstations with Hyperterminal and Ethereal installed  
(1) console cable

**Lab Diagram:**

```
  4 st
 NIC       NIC
workstation “A”     workstation “B”  
192.168.1.1/24     192.168.1.2/24
```

**Background:**
Virtual Lan’s (VLAN’s) are used to keep devices from communicating to each other without the services of a layer 3 device (router). If you were designing a school it would be nice to use a VLAN for teachers and a VLAN for students. No communication would be possible without the use of a router. So let’s get to the “learning by doing!”

**Step-By-Step Instructions:**
1. Set up and cable the lab as shown. The switch requires no ip address, mask or gateway.  
2. Ping from workstation A to B using DOS. It should work just fine.  
3. Now let’s put the teachers on one VLAN and the students on another. From the switch console let’s create the two VLANs:  
   a. Click on [M] for menus  
   b. Click on [V] for VLANs  
   c. Click on [A] for add a VLAN (this will become VLAN #2)  
   d. Click on [1] for “Ethernet” type VLAN  
   e. Click on [S] to save and exit  
   f. Click on [V] for VLANs  
   g. Click on [A] for add a VLAN (this will become VLAN #3)  
   h. Click on [1] for “Ethernet” type VLAN  
   i. Click on [S] to save and exit
4. Now we need to assign ports to the VLAN’s:
   a. Click on [E] for VLAN membership
   b. Click on [V] for VLAN assignment
   c. **Type in the ports to assign for the VLAN: 4-12 (I have a 24-port switch)**
   d. Click on [2] to assign them to VLAN #2
   e. Click on [E] for VLAN membership
   f. Click on [V] for VLAN assignment
   g. **Type in the ports to assign for the VLAN: 13-24 (I have a 24-port switch)**
   h. Click on [3] to assign them to VLAN #3
   i. All done! You can exit back to the main menu.

** We typically do not want to use VLAN #1…we reserve it for network management functions…I saved 3 ports on my 24 port switch for VLAN #1…If you take the semester 7 “Building CISCO Switched Multi-Layered Networks” then you will learn more about using VLAN 1…for now restrict users to VLAN #2 and above.

5. Try pinging again from workstation A to B using DOS. It should not work now. The VLAN’s “electrically separate” the two networks…it’s kind of like using two switches.

Supplemental Lab or Challenge Activity:
1. Add a protocol inspector and observe the VLAN information.
2. Go to CISCO’s website and research VLAN information.
3. Try setting up a switch with 5 VLAN’s.

So What Have I Learned Here?
VLANs are nice to use in large networks. Instead of physically separating network users from each other with separate (and sometimes expensive devices) we can now do it logically without using added equipment. In the next lab we will add a router into our little lab design and see how it improves or messes up our network

Some of the best books are brought to you by Lulu!

http://www.lulu.com/about/
**Objective:**
To learn how to construct and understand how to use basic Virtual LAN’s in a network.

**Tools and Materials:**
(1) CISCO switch (1900 series)
(4) straight-through cables
(2) Windows PC workstations with Hyperterminal and Ethereal installed
(1) console cable
(1) Router (I used a 2610)

**Lab Diagram:**

![Lab Diagram](image)

**Background:**
Notice in this lab that we have two subnets now…this is required for our two different ports on our router. So with our VLAN’s, especially because they are on different subnets, now they really should not be able to communicate…right? Wrong. Remember our VLAN’s can act as substitutes for equipment…this is a lab we have done several times before EXCEPT that we used multiple switches…we can redo it with one switch and some VLANs configured on it to save on equipment. As a matter of fact they can communicate just fine and dandy.

**Step-By-Step Instructions:**
1. Set up and cable the lab as shown. The switch requires no ip address, mask or gateway. Pick out the IP addresses for the router Ethernet ports that would work with the IP addresses assigned to the workstations. Don’t forget to add a routing protocol and advertise/publish your networks. Try to ping from workstation “A” to “B” it should not work. Have your instructor set up the router for you.
2. Now let’s put the teachers on one VLAN and the students on another (pick which one is which). From the switch console let’s create the two VLANs:
   a. Click on [M] for menus
   b. Click on [V] for VLANs
   c. Click on [A] for add a VLAN (this will become VLAN #2)
   d. Click on [1] for “Ethernet” type VLAN
   e. Click on [S] to save and exit
   f. Click on [V] for VLANs
   g. Click on [A] for add a VLAN (this will become VLAN #3)
   h. Click on [1] for “Ethernet” type VLAN
   i. Click on [S] to save and exit

3. Now we need to assign ports to the VLAN’s:
   a. Click on [E] for VLAN membership
   b. Click on [V] for VLAN assignment
   c. **Type in the ports to assign for the VLAN: 4-12 (I have a 24-port switch)
   d. Click on [2] to assign them to VLAN #2
   e. Click on [E] for VLAN membership
   f. Click on [V] for VLAN assignment
   g. **Type in the ports to assign for the VLAN: 13-24 (I have a 24-port switch)
   h. Click on [3] to assign them to VLAN #3
   i. All done! You can exit back to the main menu.

** We typically do not want to use VLAN #1…we reserve it for network management functions…I saved 3 ports on my 24 port switch for VLAN #1…If you take the semester 7 “Building CISCO Switched Multi-Layered Networks” then you will learn more about using VLAN 1…for now restrict users to VLAN #2 and above.

3. Try pinging again from workstation A to B using DOS. It should work. The VLAN’s “electrically separate” the two networks but the router allows communication between them.

4. Now, one of the things we like to do in networking is not use VLAN 1, VLAN 2, VLAN 3, etc. Instead we like to use things like VLAN20, VLAN 40, VLAN 80. As we have seen in this lab everytime you create a new VLAN it assumes an incrementally higher VLAN number by default. But, does that mean to get to VLAN 80 you have to first create 78 other VLAN’s? (VLAN 1 exists by default). Thankfully no, there is a way to do that. Let’s assume you are using VLAN 40 and in your network. Let’s create it now.
a. From the VLAN menu select “A” to add a VLAN:

        Catalyst 1900 - Virtual LAN Configuration
        --------------------- Information ---------------------
        VTP version: 1
        Configuration revision: 11
        Maximum VLANs supported locally: 1005
        Number of existing VLANs: 5
        Configuration last modified by: 0.0.0.0 at 00-00-0000 00:00:00
        --------------------- Settings ---------------------
        [N] Domain name
        [V] VTP mode control Server
        [F] VTP pruning mode Disabled
        [O] VTP traps Enabled
        --------------------- Actions ---------------------
        [L] List VLANs [A] Add VLAN
        [M] Modify VLAN [D] Delete VLAN
        [E] VLAN Membership [S] VLAN Membership Servers
        [T] Trunk Configuration [W] VTP password
        [P] VTP Statistics [X] Exit to Main Menu

        Enter Selection: A

b. Select “1” for Ethernet (You won’t see the “1” but you will see “Ethernet”):

        Enter Selection: A

        This command selects the type of VLAN to be added.
        The following VLAN types can be added:
        Select a VLAN type [1-5]: Ethernet

c. Select “N” to change the number of the VLAN:

        Catalyst 1900 - Add Ethernet VLAN
        --------------------- Settings ---------------------
        [N] VLAN Number 2
        [V] VLAN Name VLAN0002
        [I] 802.10 SAID 100002
        [M] MTU Size 1500
        [L] Translational Bridge 1 0
        [J] Translational Bridge 2 0
        [T] VLAN State Enabled
        [S] Save and Exit [X] Cancel and Exit

        Enter Selection: N

d. Then change the number to “40”

        This command selects the unique VLAN identifier of a VLAN.
        Configuration change only takes effect when the VLAN SAVE command is
        executed.
        Enter VLAN Number [2-1001]:
        Current setting ===> 2
        New setting ===> 40
e. Click on “S” to save the information

Catalyst 1900 - Add Ethernet VLAN
-------------------------------  Settings -------------------------------
[N] VLAN Number               40
[V] VLAN Name                 VLAN0002
[I] 802.10 SAID                100002
[M] MTU Size                  1500
[L] Translational Bridge 1    0
[J] Translational Bridge 2    0
[T] VLAN State                Enabled
[S] Save and Exit             [X] Cancel and Exit

Enter Selection:  S

f. Now you should be done. You can go out and check it by listing the VLAN’s and selecting “all” if you want. The only thing remaining is to associate some ports with it.

Troubles? What is that you say? Sometimes a good troubleshooting technique for you to use is to eliminate as many unknowns as you can to help deduce the actual trouble. In this lab there is a possibility of you having a problem with your router or your switch. The easiest way to start is to eliminate the switch altogether and use cross-over cables from the workstations into the router. Then test everything again. If everything is fine, then you know your problem is within the switch. Otherwise your problem is most likely in your router. Once you have fixed your router you can add your workstations (one at a time) back in through the switch and keep testing it every time you change one item. Before too long everything will be up and running just hunky-dory (just keep swimming, just keep swimming). This is what I like to call “baby steps” troubleshooting…one thing at a time. It works too when building the monster labs. Don’t try to do everything all at once…use baby steps to set up one thing, test, and troubleshoot if needed. One other problem I have encountered with this lab is speed mis-matches between 1900 switches and 2600 routers. Sometimes you will have to change the port speed of the Ethernet interface on the router to “half-duplex” if you are using a 1900 switch. You would think the router could handle slower speeds from its default “auto” setting, but that is just not the case. Another reason why I like hands-on and not simulators. Sims are set up so that everything is compatible. In the real world almost nothing is compatible.

Supplemental Lab or Challenge Activity:
1. Add a protocol inspector and observe the VLAN information. You will have to put one on each subnet…alas a limitation of our mighty Ethereal…it only collects information from the directly attached subnet.
2. Go to CISCO’s website and research VLAN information.

So What Have I Learned Here?
It’s ok if you are confused right now…I showed you this cool tool for saving on resources and then wiped out any hope by adding a router. Later on you will learn about access control lists (ACL’s) on routers…these will allow you to deny communications between VLAN’s once again if you want…so buck up! You are coming along nicely. In the next lab we take this design a step further by creating a partially meshed “flat-switching” network with four switches. That’s right…we are going to lose the router and set up redundancy between several switches and VLANs.
Intermediate VLAN

Objective:
To learn how to construct and understand how to configure VLAN’s in a partially-meshed flat-switching network.

Tools and Materials:
(4) CISCO switch (1900 series)
(4) straight-through cables (st)
(4) Windows PC workstations with Hyperterminal and Ethereal installed
(1) console cable
(4) Cross-over cables (xo)

Lab Diagram:

Step-By-Step Instructions:
1. Set up and cable the lab as shown. Do not forget to use cross-over cables from switch to switch.
2. A should only be able to ping to a.
3. B should only be able to ping to b.

Supplemental Lab or Challenge Activity:
1. How would you use the Ax and Bx ports for faster connectivity?
2. Why do you think we used “master” VLAN switches? I know we could have done this cheaper and easier with only two switches. Draw that diagram with only two switches. As you progress you will see why I did this lab in this manner.

So What Have I Learned Here?
In this quick little lab you learned about setting up a partially meshed VLAN network. For most of the labs for this section you will build upon this design.
Objective:
To learn how to construct and a network using VLAN’s and STP for redundancy.

Tools and Materials:
(4) CISCO switch (1900 series)
(4) straight-through cables (st)
(4) Windows PC workstations with Hyperterminal and Ethereal installed
(1) console cable
(6) Crossover cables (xo)

Lab Diagram:

Step-By-Step Instructions:
1. Set up and cable the lab as shown. Do not forget to use crossover cables from switch to switch. On the top redundant cable we will be connecting VLAN 2 with redundancy. Plug it into port 7 on each lower switch. On the lower redundant cable we will be connecting VLAN 3 with redundancy. Plug it into port 17 on each lower switch.
2. A should only be able to ping to a.
3. B should only be able to ping to b.
4. Now lets test the backup for VLAN 2. Unplug the crossover cable in port 5 on the lower left switch in our diagram. This will force the crossover cable between ports 7 to become active. Once STP has had a chance to activate that line then A should be able to ping a once again. Go ahead and plug the crossover cable back into port 5.
5. Now lets test the backup for VLAN 3. Unplug the crossover cable in port 15 on the lower left switch in our diagram. This will force the crossover cable between ports 17 to become active. Once STP has had a chance to activate that line then A should be able to ping a once again. Go ahead and plug the crossover cable back into port 15.
Supplemental Lab or Challenge Activity:
1. How would you use the Ax and Bx ports for faster connectivity?
2. Where else could we add redundancy? Be creative.

So What Have I Learned Here?
The numbers of labs left keep getting smaller and the hits just keep getting bigger! We are learning how to mix VLAN’s and STP…but are not adding in any routers just yet. We will do a couple of other labs and then come back to this design for our WECIL’s.

Pro-Cote Inc.
Specialty Coatings Applicators
877-588-4300

Certified Conductive and Static Dissipative Floor applications for every need:
- Facilities with highly sensitive electronic equipment
- Computer facilities
- Electronic manufacturing & assembly
- Facilities containing AGV lines
- Clean rooms
- Manufacturing/processing flammable or combustible materials
- Munitions plants

Pro-Cote Inc. is a proud installer of Valspar Flooring Systems
www.pro-cote-usa.com

Pro-Cote Inc.
P. O. Box 1766
Largo, FL  33779
Toll Free (877) 588-4300
Subnetting Example: ABC Packaging

Objective:
To use your subnet knowledge to design an IP addressing scheme for the ABC Packaging.

Tools and Materials:
Paper and pencil

Background:
(from Part 1) You are working as the network administrator for ABC Packaging. You are to design a network that focuses upon scalability and adaptability. There are five departments: Administration (14 people, 5 printers), Engineering (22 people, 5 printers, 1 file server), Production (5 people), Accounting (11 people, 4 printers, 1 database and file server), and Sales/Marketing (11 people, 4 printers, 1 file server). Each department will require a separate subnet. The servers will have their own subnet. Be sure to connect them to the Internet with a T-1 line. You task is to design an IP addressing scheme that will address all current needs as well as future expandability. If you see anything that may want to address feel free to note it. Scalability, adaptability, reliability and performance are the key issues in this design. You will be using private addressing in your network.

Continued:
Ok…great…you just got your wonderful network designed and implemented, so now you know why it needed to be adaptable: the “eccentric” president read an article in the “Harvard Business Review” (yeah…he could almost understand the big words) and wanted to implement a divisional team format. Sounds good to everyone but it is really going to test your knowledge of networking to make it work. Every division will have engineers, accountants, and sales people. Where before they all were in their own little area connected to a switch, now they are scattered everywhere. You could buy tons of switches to make that work OR you could use your knowledge of switching technology to move them around nicely and easily. The new divisions are: north (5 engineers, 1 accountant, and 2 sales people), south (4 engineers, 1 accountant, and 2 sales people), east (4 engineers, 1 accountant, and 2 sales people), west (5 engineers, 1 accountant, and 2 sales people), special projects/ R&D (4 engineers, 1 accountant, and 2 sales people), and the administration/production staff (6 accountants, 1 sales person, and 19 production).
**Basic VTP**

**Objective:**
In this lab you will learn the basics of the Virtual Trunking Protocol (VTP). Also you will learn how and why it is used with switches in networks.

**Tools and Materials:**
(3) CISCO switch (1900 series)
(3) straight-through cables (st)
(3) Windows PC workstations with Hyperterminal and Ethereal installed
(1) console cable
(2) Crossover cables (xo)

**Lab Diagram:**

Virtual Trunking Protocol (VTP) allows us to control network broadcasts from one switch leg to another. In our diagram above if we sent a broadcast from workstation B (for example, ping 192.168.1.255) then each switch and workstation would receive that broadcast message. Sometimes we may find our networks becoming congested and need to control those broadcasts a little bit better, especially in Novell networks. VTP is “off” by default on each port of a switch. This will allow all broadcasts through. If we enable (by turning VTP “on”) then we will stop ALL broadcasts to that port. It is kind of a double-edged sword because you cannot really be selective about which broadcasts to allow through…you can only select all of them. If we enable VTP on the bx port on the top switch you will stop any broadcasts from reaching workstation c.

**Step-By-Step Instructions:**
1. Set up and cable the lab as shown. Do not forget to use crossover cables from switch to switch.
2. Start an Ethereal capture.
3. Ping from b to c.
4. Stop the capture. You should see good icmp request and reply statements. It should look something like this:

![Ethereal capture screenshot](image)

Figure 1—Good icmp request and replies seen.

5. a should be able to ping to b and c.
6. b should be able to ping to a and c.
7. c should be able to ping to a and b.
8. Now let’s go and “enable” VTP on the bx port on the top switch:
   a. From the main menu, click on [M] for menus.
   b. Click on [V] for VLAN assignments
   c. Click on [T] for Trunk Configuration (only A and B are allowed to be trunks)
   d. Type in [b] to make changes to port bx
   e. Click on [T] for trunking (off by default)
   f. Type “1” to enable VTP (turn it on)
   g. Exit all the way out to the main menu if you want.
9. Start the Ethereal capture again.
10. Ping from workstation b to c again. (It should not work...“Request Timed Out”).
11. Stop the Ethereal capture. You should see only icmp requests...no replies anywhere...this is because the VTP stops the requests from getting through. You should see something like figure 2 on the next page.
Figure 2—Only ping requests with VTP enabled.

**THEORY STUFF**

Ok, technically you can use VTP as a trunking protocol to “make” one line carry many different subnets or to combine multiple lines to “appear” as one line. Let’s take a peak at that here.

![Diagram of network configuration with VLANs and IP addresses](image)

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000</td>
<td>127.0.0.1</td>
<td>192.168.30.2</td>
<td>STP</td>
<td>Conf. Root = 127.0.0.1:fd00:c2b6:b241:8a6e:0000:0000:0000:0000</td>
</tr>
<tr>
<td>2</td>
<td>2.003318</td>
<td>CISCO_cba:7e:41</td>
<td>192.168.30.2</td>
<td>STP</td>
<td>Conf. Root = 127.0.0.1:fd00:c2b6:b241:8a6e:0000:0000:0000:0000</td>
</tr>
<tr>
<td>5</td>
<td>8.004699</td>
<td>CISCO_cba:7e:41</td>
<td>192.168.30.2</td>
<td>STP</td>
<td>Conf. Root = 127.0.0.1:fd00:c2b6:b241:8a6e:0000:0000:0000:0000</td>
</tr>
<tr>
<td>7</td>
<td>12.923939</td>
<td>192.168.1.12</td>
<td>192.168.1.2</td>
<td>ICMP</td>
<td>Echo (ping) request</td>
</tr>
<tr>
<td>8</td>
<td>12.655870</td>
<td>CISCO_cba:7e:41</td>
<td>192.168.1.2</td>
<td>STP</td>
<td>Conf. Root = 127.0.0.1:fd00:c2b6:b241:8a6e:0000:0000:0000:0000</td>
</tr>
<tr>
<td>9</td>
<td>12.927046</td>
<td>192.168.1.2</td>
<td>192.168.1.2</td>
<td>ICMP</td>
<td>Echo (ping) request</td>
</tr>
<tr>
<td>10</td>
<td>13.031395</td>
<td>192.168.1.2</td>
<td>192.168.1.2</td>
<td>ICMP</td>
<td>Echo (ping) request</td>
</tr>
<tr>
<td>12</td>
<td>14.949954</td>
<td>192.168.1.2</td>
<td>192.168.1.2</td>
<td>ICMP</td>
<td>Echo (ping) request</td>
</tr>
<tr>
<td>14</td>
<td>18.056872</td>
<td>CISCO_cba:7e:41</td>
<td>192.168.1.2</td>
<td>STP</td>
<td>Conf. Root = 127.0.0.1:fd00:c2b6:b241:8a6e:0000:0000:0000:0000</td>
</tr>
</tbody>
</table>

workstation “a”
192.168.10.2/24
VLAN10

workstation “b”
192.168.20.2/24
VLAN20

workstation “c”
192.168.30.2/24
VLAN30
The only problem with this is we tend to not really “see” too many routers with more than one or two Ethernet interfaces. Well, technically we can using “trunking” to push everything down one Ethernet interface by using sub-interfaces. An example of this:

![Diagram of a network setup with workstations and a router, showing VLAN configuration.]

```
routing(config)#interface fa0/0
routing(config-if)#no shut
routing(config-if)#interface fa0/0.10
routing(config-subif)#encapsulation dot1q 10
routing(config-subif)#ip address 192.168.10.1 255.255.255.0
routing(config-subif)#no shut
routing(config-if)#interface fa0/0.20
routing(config-subif)#encapsulation dot1q 20
routing(config-subif)#ip address 192.168.20.1 255.255.255.0
routing(config-subif)#no shut
routing(config-if)#interface fa0/0.30
routing(config-subif)#encapsulation dot1q 30
routing(config-subif)#ip address 192.168.30.1 255.255.255.0
routin(config-subif)#no shut
routin(config-subif)#exit
routin(config)#exit
```

Ok, so the theory goes. That is the idea for you to realize it at this point. The reality is that you need a pretty high-powered little router (like a 3500 or 7000 series router with at least IOS 12.2 or greater). So you know the theory of it...later in CCNP land you can actually start doing it.
One other “use” of trunking is to take “multiple” lines and make them appear like one big fat pipe. For example, if we have one 100 Mbps line between two 2950 switches then we know we have 100 Mbps between them. Easy enough, right? Right! So, we extend that logic and have four 100 Mbps lines between those same two 2950 switches. Now we “appear” to have 400 Mbps. In actuality we do, however it is still four 100 Mbps that are “trunked” together. Again, something else to follow-up on in CCNP land.

Supplemental Lab or Challenge Activity:
1. Someone asked me why we didn’t just enable VTP on the port for workstation C on the lower left switch. Well that is another option too. Can you think of reasons to do this or to not do this?
2. Go out to CISCO and research VTP. Is this associated with VLAN’s in any way?

So What Have I Learned Here?
In this lab you have learned on method to control broadcasts to a port or switch. I really would not have included this here but I have heard some students mention basic VTP might have been on their test (hint, hint, wink, wink). I really cannot say for sure because we are not allowed to discuss test items. It was not on mine.
Router on a “stick” lab/More on VTP

Objective:
In this lab you will accomplish three main goals: (1) to learn how to set up a configuration known as “a router on a stick,” (2) to learn how to perform trunking operations on CAT2950 switches, and (3) to gain valuable information to help you pass your final exam and (ultimately) the CCNA exam. For my lab I used two Cisco Catalyst 2950 switches using 12.1.20(AE1) and an older Cisco 2610 router using 12.0.3(T3).

```
   172.16.1.1/16
Loopback 0

SW_VTP_1    SW_VTP_2
fa0/1       fa0/0
fa0/2       fa0/1
fa0/9       fa0/17
fa0/10      fa0/18
PC1         PC2
PC3         PC4
```

```
VLAN 1 addressing
IP Address:  192.168.1.2/24  192.168.1.3/24
Default-gateway:  192.168.1.1/24  192.168.1.2/24

PC addressing
IP  Gateway  VLAN Membership
PC1  192.168.30.5/24  192.168.30.1  VLAN30
PC2  192.168.30.6/24  192.168.30.1  VLAN30
PC3  192.168.40.5/24  192.168.40.1  VLAN40
PC4  192.168.40.6/24  192.168.40.1  VLAN40

VLAN 1

router e0/0 config:
e0/0.1  192.168.1.1/24
e0/0.30 192.168.30.1/24
e0/0.40 192.168.40.1/24
```

Port mapping on both switches:
VLAN 1  fa0/1-fa0/8
VLAN30  fa0/9-fa0/16
VLAN40  fa0/17-fa0/24

Configure the router with the basics (you can add passwords and other stuff if you would like the practice):

```
Router>
Router>en
Router#config t
Router(config)#line con 0
Router(config-line)#logging syn
Router(config-line)#exec-t 0 0
Router(config)#int loop 0
Router(config-if)#ip address 172.16.1.1 255.255.255.0
Router(config-if)#no shut
```

Configure the “main” Ethernet interface and bring it up:

```
Router(config-if)#int e0/0
Router(config-if)#no shut
```
Configure a sub-interface for each VLAN (including the management VLAN). The Cisco Catalyst can use either ISL or 802.1q. Since I am using an older 2610 router that can only use 802.1q I need to also make that change. I have also added in a description on each interface. Be careful using cut and paste on these:

```plaintext
Router(config-if)#int e0/0.1
Router(config-subif)#description MGT VLAN1
Router(config-subif)#encapsulation dot1q 1
Router(config-subif)#ip address 192.168.1.1 255.255.255.0

Router(config-if)#int e0/0.30
Router(config-subif)#description LEFT
Router(config-subif)#encapsulation dot1q 30
Router(config-subif)#ip address 192.168.30.1 255.255.255.0
Router(config-subif)#exit

Router(config-if)#int e0/0.40
Router(config-subif)#description RIGHT
Router(config-subif)#encapsulation dot1q 40
Router(config-subif)#ip address 192.168.40.1 255.255.255.0
Router(config-subif)#exit
```

Now all we need to do is add a routing protocol (I chose to use EIGRP…try it with OSPF too):

```plaintext
Router(config)#router eigrp 38
Router(config-router)#network 172.16.1.1
Router(config-router)#network 192.168.1.1
Router(config-router)#network 192.168.30.1
Router(config-router)#network 192.168.40.1
```

Configure the “middle” network device (a.k.a “SW_VTP_1”) with the basics:

```plaintext
Switch>
Switch>en
Switch#config t
Switch(config)#line con 0
Switch(config-line)#logging syn
Switch(config-line)#exec-t 0 0
Switch(config-line)#exit
Switch(config)#hostname SW_VTP_1
```

Now we need to set up VLAN 1. This is the management VLAN and all of our traffic will originate and arrive over VLAN 1 from the router to this switch. Therefore, we need an “overall” IP address on VLAN1 to enable routing. We also have to add a default gateway statement, just like we have to do when configuring TCP/IP on workstations:

```plaintext
SW_VTP_1(config)#int vlan 1
SW_VTP_1(config-if)#ip address 192.168.1.2 255.255.255.0
SW_VTP_1(config-if)#exit
SW_VTP_1(config)#ip default-gateway 192.168.1.1
SW_VTP_1(config)#exit
```
Next we have to set up our VLAN database. I don’t think it is really necessary right now, but later when you get into CCNP you will learn about VTP Client/Server relationships and why this information is needed on the switches:

```
SW_VTP_1(vlan)#vlan database
SW_VTP_1(vlan)#vtp domain MATT
SW_VTP_1(vlan)#vtp server
SW_VTP_1(vlan)#vlan 1 name MGT
SW_VTP_1(vlan)#vlan 30 name LEFT
SW_VTP_1(vlan)#vlan 40 name RIGHT
SW_VTP_1(vlan)#exit
```

Finally we just need to set up our ports for VLAN’s. On our first 8 ports we will configure them for trunking and leave them on their default setting of VLAN1 membership.

```
SW_VTP_1(config)#int range fa0/1 – 8
SW_VTP_1(config-if)#switchport mode trunk
SW_VTP_1(config-if)#exit
```

On ports 9-16 we will re-assign them from VLAN1 to VLAN30.

```
SW_VTP_1(config)#int range fa0/9 – 16
SW_VTP_1(config-if)#switchport access vlan 30
```

On ports 17-24 we will re-assign them from VLAN1 to VLAN40.

```
SW_VTP_1(config-if)#int range fa0/17 – 24
SW_VTP_1(config-if)#switchport access vlan 40
```

Configure the “end” network device (a.k.a “SW_VTP_2”). Any differences between switch 1’s configuration and switch 2’s configuration are italicized and underlined:

```
Switch>
Switch>en
Switch(config)#line con 0
Switch(config-line)#logging syn
Switch(config-line)#exec-t 0 0
Switch(config-line)#exit
Switch(config)#hostname SW_VTP_2
SW_VTP_2(config)#int vlan 1
SW_VTP_2(config)#ip address 192.168.1.3 255.255.255.0
SW_VTP_2(config)#exit
SW_VTP_2(config)#ip default-gateway 192.168.1.2
SW_VTP_2(config)#exit
SW_VTP_2(vlan)#vlan database
SW_VTP_2(vlan)#vtp domain MATT
SW_VTP_2(vlan)#vtp server
SW_VTP_2(vlan)#vlan 1 name MGT
SW_VTP_2(vlan)#vlan 30 name LEFT
SW_VTP_2(vlan)#vlan 40 name RIGHT
SW_VTP_2(vlan)#exit
```
SW_VTP_2#config t
SW_VTP_2(config)#int range fa0/1 – 8
SW_VTP_2(config-if)#switchport mode trunk
SW_VTP_2(config-if)#exit
SW_VTP_2(config)#int range fa0/9 – 16
SW_VTP_2(config-if)#switchport access vlan 30
SW_VTP_2(config-if)#int range fa0/17 – 24
SW_VTP_2(config-if)#switchport access vlan 40

To verify you should be able to ping from each workstation to the loopback address (172.16.1.1) and each workstation should be able to ping each other as well (irregardless of VLAN membership).

So, you should see by now that trunking can be used to limit broadcasts in a switching network (hierarchical or flat) and trunking is also used to carry inter-VLAN traffic. There are some more uses but these are two you need for now.

“Double-Dog Dare Do-overs”:
1. You could pull HTTP services from your live classroom line down to the PC’s through this lab.
2. You could set up DHCP on the router and have each PC address obtained dynamically.
3. Set up a reflexive ACL on the router and use PAT with overloading.
4. Use ACL’s to restrict packet movement between VLAN 30 and VLAN40. Heck, I “Triple-Dog Dare” you to get VLAN30 pinging to VLAN40, but not allowing VLAN40 to ping VLAN30 (one-way only).
5. Go out to cisco.com and research two terms: port-channel and fast Ethernet grouping. You can then add multiple VLAN1 connections between the two switches and aggregate the total bandwidth. Once done, you can check to see if it is working properly with the `show cdp neighbors` command.
Router Toolbox for this lab:

```
en
config t
line con 0
logging syn
exec-t 0 0
exit
int loop 0
ip address 172.16.1.1 255.255.255.0
no shut
int e0/0
no shut
int e0/0.1
encapsulation dot1q 1
ip address 192.168.1.1 255.255.255.0
int e0/0.30
encapsulation dot1q 30
ip address 192.168.30.1 255.255.255.0
int e0/0.40
encapsulation dot1q 40
ip address 192.168.40.1 255.255.255.0
exit
router eigrp 38
network 172.16.1.1
network 192.168.1.1
network 192.168.30.1
network 192.168.40.1

en
config t
line con 0
logging syn
exec-t 0 0
exit
hostname SW_VTP_1
int vlan 1
ip address 192.168.1.2 255.255.255.0
exit
ip default-gateway 192.168.1.1
exit
vlan database
vtp domain MATT
vtp server
vlan 1 name MGT
vlan 30 name LEFT
vlan 40 name RIGHT
exit
config t
int range fa0/1 – 8
switchport mode trunk
exit
int range fa0/9 – 16
switchport access vlan 30
int range fa0/17 – 24
switchport access vlan 40
```
en
config t
line con 0
logging syn
exec-t 0 0
exit
hostname SW_VTP_2
int vlan 1
ip address 192.168.1.3 255.255.255.0
exit
ip default-gateway 192.168.1.2
exit
vlan database
vtp domain MATT
vtp server
vlan 1 name MGT
vlan 30 name LEFT
vlan 40 name RIGHT
exit
config t
int range fa0/1 – 8
switchport mode trunk
exit
int range fa0/9 – 16
switchport access vlan 30
int range fa0/17 – 24
switchport access vlan 40
Using a 2950 Switch

Objective:
There are many schools that use only 2950’s for their switches. In this lab you will learn how to set up the basics on the switch, configure interfaces, and set up VLAN’s.

Tools and Materials:
(1) workstation
(1) console cable
(1) 2950 switch (I used IOS version 12.1 here…there are still some “issues” with these commands and all other switches)

Lab Design:

Step-By-Step Instructions:
1. Set up and cable the lab as shown. Use a console cable from COM1 on the workstation into the console port on the back of the switch. Open a hyperterminal session on the workstation. Turn the power on to the switch by plugging it in. Put in a “n” or “no” to not enter the initial configuration. You should see something like:

```
% Please answer 'yes' or 'no'.
Would you like to enter the initial configuration dialog?
[yes/no]: n
Press RETURN to get started!
```

```
00:09:24: %LINK-5-CHANGED: Interface Vlan1, changed state to administratively down
00:09:25: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
Switch>
```

2. Switch to enable mode by typing an “en” or “enable” at the prompt:

```
Switch>en
Switch#
```

If you are prompted for a password then someone else has been there first and has put in an “enable” password. You will have to have your instructor or lab technician clear this out. Be sure to have them reset the switch to the factory default settings.
3. Now, let’s double check and make sure everything is set to defaults for this particular IOS and switch version. Here is what I saw using a “\texttt{sh run}” or “\texttt{show run}” command to see the running configuration file on the switch (some blank lines have been edited out to save some trees):

```
Switch#sh run
Building configuration...
Current configuration : 1449 bytes
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
hostname Switch
ip subnet-zero
spanning-tree extend system-id
!
interface FastEthernet0/1
  no ip address
!
interface FastEthernet0/2
  no ip address
!
(I took out interfaces FastEthernet 0/3 through 0/22…they are all the same with no ip address…just saving a page and some trees)

!
interface FastEthernet0/23
  no ip address
!
interface FastEthernet0/24
  no ip address
!
interface GigabitEthernet0/1
  no ip address
!
interface GigabitEthernet0/2
  no ip address
!
interface Vlan1
  no ip address
  no ip route-cache
  shutdown
!
ip http server
!
!
line con 0
line vty 5 15
!
end

Switch#
```
4. The first thing we will want to do is set up some basics on the switch that will keep us from screaming our head off. Here is what I recommend (just read this for now...I will explain line-by-line in a minute):

```
Switch>
Switch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname Matt_switch

Matt_switch(config)#line vty 0 ?
<1-15> Last Line number
<cr>

Matt_switch(config)#line vty 0 15
Matt_switch(config-line)#password cisco
Matt_switch(config-line)#login
Matt_switch(config-line)#exit
Matt_switch(config)#line con 0
Matt_switch(config-line)#logging syn
Matt_switch(config-line)#exec-t 0 0
Matt_switch(config)#enable secret cisco
Matt_switch(config)#enable password class
```

Let’s break this down a bit. First I switched into configuration mode. Having the prompt with a carat is called the “user mode” and you cannot do anything but look at how the switch is performing. Any changes require you to be in the configuration mode first. This is that sequence of commands:

```
Switch>
Switch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#
```

Next, I changed the name of the switch. When you have many switches in your network this will help you keep them apart when configuring several at the same time. Here is that command. Notice how the prompt changes immediately:

```
Switch(config)#hostname Matt_switch
Matt_switch(config)#
```

Hostnames must be one contiguous group of characters and numbers. But, aha! I can use an underscore mark to make it appear like I have multiple words (21 character maximum). Here are a few good hostnames:

```
hostname mattswitch1
hostname May_I_Momma_Dogface
hostname Orlando_switch
hostname OrlSwitch1%
```
Next, I want to configure the virtual terminal lines. In the “standard” curriculum you are told to only configure the first five lines (vty 0 through 4). Well, the newer equipment comes available with more than five lines so you want to be sure you get them all. So first we find out how many lines we have and then configure it for all of them.

```
Matt_switch(config)#line vty 0 ?
   <1-15>  Last Line number
   <cr>

Matt_switch(config)#line vty 0 15
Matt_switch(config-line)#password cisco
Matt_switch(config-line)#login
Matt_switch(config-line)#exit
Matt_switch(config)#
```

The vty lines are used during telnet sessions to the switch. If you do not configure a password or add the capability to login in to a vty session then you will not be able to telnet into the switch. It’s a catch-22, if you do not use it then you cannot telnet into the switch. But if you do, then you open a possible security hole that may allow anyone to telnet in to the switch. If you only configure the first five telnet lines then you may also open a security hole on all remaining lines. You do not have to use the same password on all lines. You can make a configuration like this too:

```
Matt_switch(config)#line vty 0 4
Matt_switch(config-line)#password cisco
Matt_switch(config-line)#login
Matt_switch(config)#line vty 5
Matt_switch(config-line)#password matt
Matt_switch(config-line)#login
Matt_switch(config)#line vty 6 15
Matt_switch(config-line)#password lophtcrack
Matt_switch(config-line)#login
```

The fun never stops…right? Next, I used the `exit` command to exit from configuring the vty lines and then used the `line con 0` command to switch into configuring the console line. You can do it like the top or the bottom example here:

```
Matt_switch(config-line)#exit
Matt_switch(config)#line con 0
or,
Matt_switch(config-line)#line con 0
Matt_switch(config-line)#
```

Unlike the vty lines there is only one console line on the 2950 switch. You can verify this for good measure:

```
Matt_switch(config)#line con ?
   <0-0>  First Line number
   <cr>
Matt_switch(config)#line con 1
```
Ok…so let’s get in and configure our console line. This is where all console messages are sent to by default on the switch. It makes sense because that is the port that is connected to our hyperterminal session. I know, I know, but I saw the question on a practice test.

```plaintext
Matt_switch(config)# line con 0
Matt_switch(config-line)# logging syn
Matt_switch(config-line)# exec-t 0 0
```

Let’s look at what I did…I got into line configuration mode and enabled logging synchronous. This is helpful to you when setting up the switch. Sometimes messages will interrupt what you are doing. If you have this command enabled then the switch console session will repeat what you had typed before the interruption. Nice, huh?

The executive timeout command acts sort of like a screen saver. Without this command you could run to the restroom and come back and find your self having to hit enter to get back into the switch at the user mode, typing enable, and then entering the password to get back into privileged mode. What a pain. Of course in the “real world” you really don’t want to do this so the IOS has a way to set the time out with the little numbers at the end….it is sort of a start and stop if you will. Setting it to 0 0 will never time out the session. Setting it to 0 60 will have it time out after 60 seconds. Unlike the hostname command this sometimes takes a bit to kick in…setting it to 0 1 will totally torque off someone, so use it only on special occasions. The last two commands are password settings for use with your switch. The enable secret password is used to access the privileged mode on your switch.

```plaintext
Matt_switch(config)# enable secret cisco
Matt_switch(config)# enable password class
```

Enable password is something that Cisco drums into your head for tests. That’s all you need to know about it for now.

5. Another thing you may want to do is to configure the interfaces on the 2950 switch. The first thing you need to decide is whether you are configuring one interface or a whole group with the same settings. Since you have way more Ethernet ports on the switch than you usually do with a router you can do ranges to configure multiple ports at once. Let’s say for example we want to set ports 1 through 12 to be 10 MB per second and the rest of the ports to be 100 MB per second. Here is the sequence of commands we could use to do all of them as a two separate range commands. Notice that there is a space between the “1” and the dash and the dash and the “12.”

```plaintext
Matt_switch# config t
Matt_switch(config)# interface range fastethernet0/1 - 12
Matt_switch(config-if-range)# speed 10
Matt_switch(config)# interface range fastethernet0/12 - 24
Matt_switch(config-if-range)# speed 100
```
That is all fine and jim dandy but it usually is best to set all the ports so they can autonegotiate how fast they can communicate. If you set the speed to 10 MB and more is available then guess what? You will still only get 10MB max. You might as well set it up for maximum efficiency. The only times I can think of where you would want to scale it back is to limit someone from watching lots of video or doing audio streaming when they should be working. Then you can slow them down (with the permission of the boss of course).

```plaintext
Matt_switch(config)#interface range fastethernet0/1 - 24
Matt_switch(config-if-range)#speed auto
```

This command is enabled by default so if you look at your running configuration to see if it is there you will not see anything. Just know that is really is there. You would think they would have another mode that would allow you to see all of the default commands. Well, if they do eventually get one I want the royalties and call shotgun on that one!

Now there are a couple of additions to the 2950 that the 1900’s really did not have, the addition of two uplink/downlink gigabit ports. The earlier 1900’s have an “A” and a “B” port capable of 100 Mbps. This is analogous to that, except that it is gigabit speed. In order to use these you need a Gigabit Interface Converter (GBIC). This is nothing more than a transceiver (or plug in converter module) that will usually be a fiber optic connection module. This is where your connection from the main wiring closet will come in to the switch. Those too are configurable.

```plaintext
Matt_switch(config)#interface range gigabitethernet0/1 - 2
Matt_switch(config-if-range)#speed auto
```

One nice feature is the description command. This will allow you to add a comment about an interface. It is particularly helpful with the gigabit interfaces like so:

```plaintext
Matt_switch(config)#interface range gigabitethernet0/1
Matt_switch(config-if-range)#description main line to MDF
Matt_switch(config)#interface range gigabitethernet0/2
Matt_switch(config-if-range)#description backup line to MDF
```

6. Lastly, you may want to configure VLAN’s on the 2950 switch. Doing this will require you to be in VLAN server mode. By default you are in the VTP client mode. Don’t believe me? Good! Let’s try it out.

```plaintext
Matt_switch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Matt_switch(config)#vlan 10
VTP VLAN configuration not allowed when device is in CLIENT mode.
```
Isn’t that about enough to drive you nuts? Well first we need to get into the VLAN database to make the switch. Notice how we do not get into terminal configuration mode first.

```
Matt_switch# vlan data
Matt_switch(vlan)# vtp ?
    client       Set the device to client mode.
    domain       Set the name of the VTP administrative domain.
    password     Set the password for the VTP administrative domain.
    pruning      Set the administrative domain to permit pruning.
    server       Set the device to server mode.
    transparent  Set the device to transparent mode.
    v2-mode      Set the administrative domain to V2 mode.
```

Matt_switch(vlan)# vtp server
Setting device to VTP SERVER mode.
Matt_switch(vlan)# exit
APPLY completed.
Exiting....
Matt_switch#

Now let’s go back again and set up those VLAN’s. Like the VLAN’s on our 1900’s there is a two-step process. First we create the VLAN and then we apply it. To create it:

```
Matt_switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
Matt_switch(config)# vlan 10
```

I believe the media type is set to Ethernet by default but I add it in just to be safe. Now to apply it:

```
Matt_switch# config t
Matt_switch(config)# interface range fastethernet0/1 - 12
Matt_switch(config-if-range)# switchport access vlan 10
```

To confirm this exit the configuration mode and do a show run. You should see something like this (I omitted a bunch of stuff and put just the pertinent stuff):

```
interface FastEthernet0/12
    switchport access vlan 10
    no ip address

interface FastEthernet0/13
    no ip address

interface Vlan1
    no ip address
    no ip route-cache
    shutdown
```
Oops! Notice our Vlan1 is shutdown. I didn’t see any Vlan 10 listing though. That is because we need to go back and bring it up to show up in our running configuration:

```
Matt_switch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Matt_switch(config)#vlan 10
Matt_switch(config-vlan)#no shut
02:50:21: %LINK-3-UPDOWN: Interface Vlan10, changed state to up
Matt_switch(config-vlan)#
```

Then we can double-check it with our running-configuration (Ok…so it worked for me once…none of this could come up the next time I did it…):

```
interface Vlan10
  no ip address
  no ip route-cache
```

One last thing you may do is to configure an IP address on a switch. When we did it from the menus on 1900’s it was easy. Here too. It just combines using VLAN’s and interfaces (Ok…so it worked for me once…none of this could come up the next time I did it…).

```
Matt_switch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Matt_switch(config)#vlan 1
Matt_switch(config-vlan)#no shut
02:50:21: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
Matt_switch(config-vlan)#ip address 192.168.1.2 255.255.255.0
Matt_switch(config-vlan)#ip default-gateway 192.168.1.1
```

There are just so many things to do with the switches. Where to start is easy. Where to stop is difficult. The best thing you can do to learn more is to go out to the Cisco website and look up all the different command options available for your specific 2950 and IOS version. Try starting with VTP and STP on your switch. When you are done with your work or even intermittently you should be sure to save your work:

```
Matt_switch#copy run start
Destination filename [startup-config]?
Building configuration...
[OK]
```

```
or,
```

```
Matt_switch#wr
Building configuration...
[OK]
Matt_switch#
```

Later on after you learn about ACL’s on routers come on back and put some ACL’s on your switches. Yeah, I said it…you can do that here too.
Clearing a password from a 2950 switch:

Hold the “mode” button down while attaching the power cord
When the STAT light goes off, then you can let it go.

Your prompt should be this:

```
switch:
```

```
switch: flash_init
Initializing Flash...
flashfs[0]: 143 files, 4 directories
flashfs[0]: 0 orphaned files, 0 orphaned directories
flashfs[0]: Total bytes: 3612672
flashfs[0]: Bytes used: 2729472
flashfs[0]: Bytes available: 883200
flashfs[0]: flashfs fsck took 86 seconds
....done Initializing Flash.
Boot Sector Filesystem (bs:) installed, fsid: 3
Parameter Block Filesystem (pb:) installed, fsid: 4
```

```
switch:

!--- This output is from a 2900XL switch. Output from a
!--- 3500XL, 3550 or 2950 will vary slightly.
```

```
switch: load_helper
```

```
switch: rename flash:config.text flash:config.old
```

```
switch: boot
```

```
Loading "flash:c3500xl-c3h2s-mz.120-5.WC7.bin"....done Initializing Flash.
```

Using a 4000/5000 Switch

Objective:
There are many schools that use only 2950’s for their switches. In this lab you will learn how to set up the basics on the switch, configure interfaces, and set up VLAN’s.

Tools and Materials:
(1) workstation
(1) console cable
(1) 4006/5000 switch

Lab Design:
**Background:**

Before we begin I wanted to explain a bit about the “monster” that is the 4000 and 5000 series of switches. For starters, they are really more of a layer 3 switch, in other words they are mostly layer 2 switches with layer 3 (routing) functionality. As you will see there will be a switch and a router to configure within this device. Later you will see how these function come together in the whole enchilada crazy insano labs.

First off the Catalyst family is a modular switch, meaning you can interchange modules (also known as “blades”) to change the functionality of the overall device. For example, you could fill the chasis of a 5000 that has five blade slots with one supervisor engine (minimum one “soup” engine required), three 12 port fast Ethernet switching modules and one FDDI card. Each catalyst is also number according to how many blade ports are contained within it. For example a catalyst 5005 is a catalyst 5000 with 5 blade ports. A catalyst 5513 is a catalyst 5500 with 13 blade ports.

Next you will see two or three power supply ports that may have two or three power supplies in them. If we have two power supplies is one a redundant power supply? You may think so but, in fact, they are both required to be turned on in order for the switch to work properly.

The next “oddity” with the catalyst is the console port on the supervisor engine. It really depends on each specific catalyst which type of cable is used to console into the catalyst. Some, like the catalyst 5005, uses a console cable. Others, like a catalyst 5513 uses a straight through cable. Just know if one cable does not work then try the other until you get the scripts in a hyperterminal session. Yeah, I know, weird. But Cisco has bought, subcontracted, or developed various parts of them from all over the world and really didn’t, in my opinion, provide consistency specifications for them. Disorder by dissemination! Resistance is futile! With this in mind let’s get going!

**Step-By-Step Instructions:**

1. Set up and cable the lab as shown. Turn the power on to the catalyst switch and open a hyperterminal session.
2. When the power comes up and the switch “settles in” you may be asked for a password. If so, put it in (ask your instructor). If not, then let’s see our default configuration using the `show config` command. Be ready…this is going to burn a few pages (I did take out some blank lines and compress for spacing a bit):

```plaintext
Console> (enable) show config
begin
#version 4.5(13a)
set password $1$FMFQ$HfZR5DUszVHIRhrz4h6V70
set enablepass $1$FMFQ$HfZR5DUszVHIRhrz4h6V70
set prompt Console>
set length 24 default
set logout 20
set banner motd ^C^C
```

275
#system
set system baud  9600
set system modem disable
set system name
set system location
set system contact
#snmp
set snmp community read-only      public
set snmp community read-write     private
set snmp community read-write-all secret
set snmp rmon disable
set snmp trap disable module
set snmp trap disable chassis
set snmp trap disable bridge
set snmp trap disable repeater
set snmp trap disable vtp
set snmp trap disable auth
set snmp trap disable ippermit
set snmp trap disable vmps
set snmp trap disable entity
set snmp trap disable config
set snmp trap disable stpx
set snmp trap disable syslog
set snmp extendedrmon vlanmode disable
set snmp extendedrmon vlanagent disable
set snmp extendedrmon enable
#ip
set interface sc0 1 0.0.0.0 0.0.0.0 0.0.0.0
set interface sc0 up
set interface sl0 0.0.0.0 0.0.0.0
set interface sl0 up
set arp agingtime 1200
set ip redirect   enable
set ip unreachable   enable
set ip fragmentation enable
set ip alias default         0.0.0.0
#Command alias
!
vmps
set vmps server retry 3
set vmps server reconfirminterval 60
set vmps tftpserver 0.0.0.0 vmps-config-database.1
set vmps state disable
#dns
set ip dns disable
#tacacs+
set tacacs attempts 3
set tacacs directedrequest disable
set tacacs timeout 5
#authentication
set authentication login tacacs disable console
set authentication login tacacs disable telnet
set authentication enable tacacs disable console
set authentication enable tacacs disable telnet
set authentication login local enable console
set authentication login local enable telnet
set authentication enable local enable console
set authentication enable local enable telnet

#bridge
set bridge ipx snaptoether   8023raw
set bridge ipx 8022toether   8023
set bridge ipx 8023rawtofddi snap

#vtp
set vtp mode server
set vtp v2 disable
set vtp pruning disable
set vtp pruneeligible 2-1000
clear vtp pruneeligible 1001-1005

#spantree
#uplinkfast groups
set spantree uplinkfast disable
#backbonefast
set spantree backbonefast disable

#vlan 1
set spantree enable     1
set spantree fwddelay 15    1
set spantree hello    2     1
set spantree maxage   20    1
set spantree priority 32768 1

#vlan 1003
set spantree enable     1003
set spantree fwddelay 15    1003
set spantree hello    2     1003
set spantree maxage   20    1003
set spantree priority 32768 1003
set spantree portstate 1003 block 0
set spantree portcost 1003 62
set spantree portpri  1003 4
set spantree portfast 1003 disable

#vlan 1005
set spantree enable     1005
set spantree fwddelay 15    1005
set spantree hello    2     1005
set spantree maxage   20    1005
set spantree priority 32768 1005
set spantree multicast-address 1005 ieee

#cgmp
set cgmp disable
set cgmp leave disable

#syslog
set logging console enable
set logging server disable
set logging level cdp 2 default
set logging level mcast 2 default
set logging level dtp 5 default
set logging level dvlan 2 default
set logging level earl 2 default
set logging level fddi 2 default
set logging level ip 2 default
set logging level pruning 2 default
set logging level snmp 2 default
set logging level spantree 2 default
set logging level sys 5 default
set logging level tac 2 default
set logging level tcp 2 default
set logging level telnet 2 default
set logging level tftp 2 default
set logging level vtp 2 default
set logging level vmps 2 default
set logging level kernel 2 default
set logging level filesys 2 default
set logging level drip 2 default
set logging level pagp 5 default
set logging level mgmt 5 default
set logging level mls 5 default
set logging level protfilt 2 default
set logging level security 2 default
set logging level server facility LOCAL7
set logging server severity 4
set logging buffer 500
set logging timestamp enable
#ntp
set ntp broadcastclient disable
set ntp broadcastdelay 3000
set ntp client disable
clear timezone
set summertime disable
#permit list
set ip permit disable
#drip
set tokenring reduction enable
set tokenring distrib-crf disable
#igmp
set igmp disable
#standby ports
set standbyports disable
#module 1 : 2-port 100BaseTX Supervisor
set module name 1
set vlan 1 1/1-2
set port channel 1/1-2 off
set port channel 1/1-2 auto
set port enable 1/1-2
set port level 1/1-2 normal
set port duplex 1/1-2 half
set port trap 1/1-2 disable
set port name 1/1-2
set port security 1/1-2 disable
set port broadcast 1/1-2 100%
set port membership 1/1-2 static
set cdp enable 1/1-2
set cdp interval 1/1-2 60
set trunk 1/1 auto isl 1-1005
set trunk 1/2 auto isl 1-1005
set spantrree portfast 1/1-2 disable
set spantrree portcost 1/1-2 19
set spantrree portpri 1/1-2 32
set spantrree portvlancost 1/1 0
set spantrree portvlancost 1/2 0
set spantrree portvlancost 1/1 cost 18
set spantrree portvlancost 1/2 cost 18
#module 2 : 12-port 100BaseTX Ethernet

278
set module name 2
set module enable 2
set vlan 1 2/1-12
set port enable 2/1-12
set port level 2/1-12 normal
set port duplex 2/1-12 half
set port trap 2/1-12 disable
set port name 2/1-12
set port security 2/1-12 disable
set port broadcast 2/1-12 0
set port membership 2/1-12 static
set cdp enable 2/1-12
set cdp interval 2/1-12 60
set trunk 2/1 auto isl 1-1005
set trunk 2/2 auto isl 1-1005
set trunk 2/3 auto isl 1-1005
set trunk 2/4 auto isl 1-1005
set trunk 2/5 auto isl 1-1005
set trunk 2/6 auto isl 1-1005
set trunk 2/7 auto isl 1-1005
set trunk 2/8 auto isl 1-1005
set trunk 2/9 auto isl 1-1005
set trunk 2/10 auto isl 1-1005
set trunk 2/11 auto isl 1-1005
set trunk 2/12 auto isl 1-1005
set spantree portfast 2/1-12 disable
set spantree portcost 2/1-12 19
set spantree portpri 2/1-12 32
set spantree portvlanpri 2/1 0
set spantree portvlanpri 2/2 0
set spantree portvlanpri 2/3 0
set spantree portvlanpri 2/4 0
set spantree portvlanpri 2/5 0
set spantree portvlanpri 2/6 0
set spantree portvlanpri 2/7 0
set spantree portvlanpri 2/8 0
set spantree portvlanpri 2/9 0
set spantree portvlanpri 2/10 0
set spantree portvlanpri 2/11 0
set spantree portvlanpri 2/12 0
set spantree portvlancost 2/1 cost 18
set spantree portvlancost 2/2 cost 18
set spantree portvlancost 2/3 cost 18
set spantree portvlancost 2/4 cost 18
set spantree portvlancost 2/5 cost 18
set spantree portvlancost 2/6 cost 18
set spantree portvlancost 2/7 cost 18
set spantree portvlancost 2/8 cost 18
set spantree portvlancost 2/9 cost 18
set spantree portvlancost 2/10 cost 18
set spantree portvlancost 2/11 cost 18
set spantree portvlancost 2/12 cost 18
#module 3 empty
#module 4 empty
#module 5 empty
#switch port analyzer
!set span 1 1/1 both inpkts disable
Ok…right off the bat we can see this default configuration is huge. Well, ok, it is not so default because there are a couple of passwords, but it is there. You can see we have a slightly different language/programming style than we used with our 2950’s. The programming for the 2950’s more closely resembles the programming style of the 2500/2600 routers you will use later. The catalyst 4000/5000 series uses what is called a “set” based programming language. Instead of using “enable password cisco” to set the enable password we would now use “set enablepass” to start the process.

3. Now that we have seen our basic default configuration let’s go ahead and put some basic commands to use here. Let’s set up an enable password and change the name of the prompt. First, let’s use our help function and see what commands are available:

```
Console> (enable) ?
Commands:
configure Configure system from network
disable Disable privileged mode
disconnect Disconnect user session
download Download code to a processor
enable Enable privileged mode
help Show this message
history Show contents of history substitution buffer
ping Send echo packets to hosts
quit Exit from the Admin session
reconfirm Reconfirm VMPS
reload Force software reload to linecard
reset Reset system or module
session Tunnel to ATM or Router module
set Set, use 'set help' for more info
show Show, use 'show help' for more info
slip Attach/detach Serial Line IP interface
switch Switch to standby <clock|supervisor>
telnet Telnet to a remote host
test Test, use 'test help' for more info
traceroute Trace the route to a host
upload Upload code from a processor
wait Wait for x seconds
write Write system configuration to terminal/network
```

Then, let’s also look at the options available with our set command:

```
Console> (enable) set help
Set commands:
set alias Set alias for command
set arp Set ARP table entry
set authentication Set TACACS authentication
set banner Set message of the day banner
set bridge Set br., use 'set bridge help' for more info
set cam Set CAM table entry
```
Lot’s of good information in there. Let’s use some of these commands. First, to change the enable password you type in set enablepass and then you need to put in the old password and the new password, twice.

```
Console> (enable) set enablepass
Enter old password:
Enter new password:
Retype new password:
Password changed.
```

Alas! As you have come to expect by now (like using different cables to console into catalyst switches) all commands do not work the same with the catalyst switches. If you followed the logic of setting the enable password you would expect to type in set prompt and then be prompted for the new prompt name, but that is not so (see below). Instead the switch wants it all at once!

```
Console> (enable) set prompt
Usage: set prompt <prompt_string>
```
Notice how the prompt changes immediately after changing to the new name. Did you also notice by now that the prompts are different than you saw with the 2950’s? Yeah, no more carats (>) or pound symbols (#), just a big empty space…the final frontier.

4. Next, let’s set up our “vty” line. You really do not have one, the closest thing you have is an “sc” and “me” interfaces. An “sc” interface is an in-band interface and a “me” is an Ethernet managemet interface. You need to have one of these configured to allow you to telnet into the catalyst switch. Basically this will apply an IP address to our switch. The command to do this is the set interface command. You just need to add the ip address and network mask to the end of the command like so:

```
Cat_Switch (enable) set interface sc0 192.168.1.2 255.255.255.0
Interface sc0 IP address and netmask set.
```

The next thing we may want to do is assign the sc0 interface to another VLAN. By default sc0 is set to VLAN 1. Here is the command in case you want to change sc0 from VLAN 1 to VLAN 20:

```
Cat_Switch (enable) set interface sc0 20
VLAN 20 does not exist.
```

You will note I left in the error message at this point. Of course we have not created a VLAN yet. Most instructions I have seen have you try to do this command too early on and I wanted to point it out to you.

5. So, let’s make some VLAN’s and try again! First we have to set up our VTP domain and give it a name. This is a network-wide domain that is used to communicate between all other switches. There is actually vtp packets sent to communicate the information between the switches with vlan information, including adding, deleting or modifying the vlan’s. It really takes up VLAN a notch doesn’t it?

```
Cat_Switch (enable) set vtp domain matt
VTP domain matt modified
```

Once we have created our vtp domain or set it up to communicate with the same domain name in our network then we can set up our vlans. In this example I want to create a VLAN 20 (named “loophole”) that uses the first 10 ports on my Fast Ethernet blade that has been put into blade port #2. Then I want to creat a VLAN 30 (named “amaffew”) that uses the other two ports. Notice how we have to call the interfaces out first by the blade port number add a slash and then the port numbers (2/1-10).
Your numbers may vary because I have the soup engine in my first blade port and the Fast Ethernet blade in my second blade port.

Cat_Switch (enable) set vlan 20 2/1-10
VLAN 20 modified.
VLAN 1 modified.
VLAN Mod/Ports
---- -----------------------
20 2/1-10

Cat_Switch (enable) set vlan 20 name loophole
Vlan 20 configuration successful
Cat_Switch (enable) set vlan 30 2/11-12
VLAN 30 modified.
VLAN 1 modified.
VLAN Mod/Ports
---- -----------------------
30 2/11-12

Cat_Switch (enable) set vlan 30 name amaffew
Vlan 30 configuration successful
Cat_Switch (enable)

Let’s just double check those vlans:

Cat_Switch (enable) show vlan
VLAN Name                             Status    IfIndex Mod/Ports,
Vlans
---- -------------------------------- --------- ------- -----------
1    default                          active    5
20   loophole                         active    23    2/1-10
30   amaffew                          active    24    2/11-12
1002 fddi-default                     active    6
1003 token-ring-default               active    9
1004 fddinet-default                  active    7
1005 trnet-default                    active    8

You can see our five default vlan’s and the two vlan’s we just created.

6. Next, we should probably add a gateway to our catalyst switch. This will tell the switch where to send all packets when they come to the switch. You will notice that earlier I used the ip address 192.168.1.2/24 for the catalyst switch. Usually I use the first available ip address for the border device, in this case a router Ethernet interface (192.168.1.1/24). So let’s add that in as our gateway:

    Cat_Switch (enable) set ip route default 192.168.1.1
    Route added.

Cat_Switch (enable)

7. There are just so many things you can do with these things and I am sure you will have a blast when you get to the CCNP switching class if you enjoyed this so far. Some of the topics you will see there include (some of these are my additions):
   i. Catalyst family maintenance and upkeep
   ii. Configuring Port Fast
iii. Configuring Uplink Fast
iv. Configuring Backbone Fast
v. Router on a stick
vi. Configuring Router Switch Modules
vii. Hot Swappable Routing Protocol (HSRP)
viii. Trunking
ix. SNMP with Switches
x. Protocol inspectors and packet analysis with switches
xi. DHCP on Catalyst switches
xii. Encapsulation variations for switches
xiii. InterVLAN routing issues with current Cisco IOS’s
xiv. AAA with Catalyst switches
xv. ACL’s with switches
xvi. Security functions on switches
xvii. Wireless networking with switches
xviii. Wireless security with switches
xix. Setting up DNS servers and using them with switches
xx. 6500 switch basics
xxi. VOIP basics with 6500 switches
xxii. Holy Enchilada! Maximum Crazy Insano Labs!

Keep track of updates and changes at [http://www.spcollege.edu/star/cisco](http://www.spcollege.edu/star/cisco) Scroll to the bottom of the page and click on the “Lab Manual Edits.”

---

**Part 2 Command Review**

**Objective:**
To list all commands utilized in Part 2 of this textbook.

**Step-by-Step Instructions:**
1. For each of the commands give a description of the command, the prompt for configuration, and any abbreviations for that command. You will have to list the commands here. ☺

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Command</th>
<th>Shortcut</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prompt</td>
<td>Command</td>
<td>Shortcut</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Whole Enchilada/Crazy Insano Lab #1 (WECIL): Switching

Objective:
To put all or most of the concepts together into one large lab. In this lab we will be simulating a school with 3 rooms using VLANs and STP.

Tools and Materials:
(5) CISCO switches (1900 series)
(6) straight-through cables (st)
(6) Windows PC workstations with Hyperterminal and Ethereal installed
(1) console cable
(10) Crossover cables (xo)

Lab Diagram:

Step-By-Step Instructions:
1. Devise an IP addressing scheme for the network shown. Be sure to include subnet masks and gateways for devices. Include an MDF/IDF drawing and a Hierarchical design drawing.
2. Cable the lab as shown.
3. All VLAN 2 devices should have communication to all VLAN 2 devices only.
4. Test your redundant lines for VLAN 2.
5. All VLAN 3 devices should have communication to all VLAN 3 devices only.
6. Test your redundant lines for VLAN 3.
7. Add redundant lines in between the individual room switches and the master VLAN switches.
Whole Enchilada/Crazy Insano Lab #2 (WECIL): Switching

Objective:
To put all or most of the concepts together into one large lab. In this lab we will be simulating a school with 3 rooms using VLANs and STP.

Tools and Materials:
(5) CISCO switches (1900 series)
(8) straight-through cables (st)
(6) Windows PC workstations with Hyperterminal and Ethereal installed
(1) console cable
(10) Crossover cables (xo)
(1) Router

Lab Diagram:

Step-By-Step Instructions:
1. In this lab we will do the same lab but add a router to the mix. How does that change your IP addressing scheme? So the next time you design a switching network that may include routers in the future how would you design the IP scheme. Redraw your network.
2. All VLAN 2 devices should have communication to all devices.
3. Test your redundant lines for VLAN 2.
4. All VLAN 3 devices should have communication to all devices.
5. Test your redundant lines for VLAN 3.
6. Add redundant lines in between the individual room switches and the master VLAN switches.
Whole Enchilada/Crazy Insano Lab #3 (WECIL): Switching

Objective:
To put all or most of the concepts together into one large lab. In this lab we will be simulating a school with 3 rooms using VLANs and STP.

Tools and Materials:
(5) CISCO switches (1900 series)
(8) straight-through cables (st)
(6) Windows PC workstations with Hyperterminal and Ethereal installed
(1) console cable
(10) Crossover cables (xo)
(1) router

Lab Diagram:

Step-By-Step Instructions:
1. How come we don’t need any IP addresses, subnet masks, and gateways on our switches? Try this lab by redesigning your network with IP addresses, subnet masks and gateways on your switches.
2. All VLAN 2 devices should have communication to all devices.
3. Test your redundant lines for VLAN 2.
4. All VLAN 3 devices should have communication to all devices.
5. Test your redundant lines for VLAN 3.
6. Add redundant lines in between the individual room switches and the master VLAN switches.
Whole Enchilada/Crazy Insano Lab #4 (WECIL): Switching

Objective:
To put all or most of the concepts together into one large lab. In this lab we will be simulating a school with 3 rooms using VLANs and STP.

Tools and Materials:
(5) CISCO switches (1900 series)
(6) straight-through cables (st)
(6) Windows PC workstations with Hyperterminal and Ethereal installed
(1) console cable
(10) Crossover cables (xo)
(1) DCE/DTE serial cable
(2) routers

Lab Diagram:

Step-By-Step Instructions:
1. Let’s repeat the last lab but add a web connection.
2. All VLAN 2 devices should have communication to all devices.
3. Test your redundant lines for VLAN 2.
4. All VLAN 3 devices should have communication to all devices.
5. Test your redundant lines for VLAN 3.
6. Each workstation should be able to ping the loopback on the ISP router.
Whole Enchilada/Crazy Insano Lab #5 (WECIL): Switching

Objective:
There is nothing to do here, but I double-dog dare you to try it. The diagram shows the need for a bit of trunking below and I know I didn’t show you how to do it…that’s what makes it so crazy. Go out and look it up! I just wanted to show you the progression of “equipment” in these last wecil’s. The Catalyst 4000/5000 would take the place of the upper-layer stuff. More or less the Core layer. This is a much better design with redundancy built in than in the last WECIL.

Lab Diagram:
Are you enjoying the materials? Well be on the lookout for some other manuals and textbooks on
http://www.lulu.com/learningbydoing and http://www.spcollege.edu/star/cisco

The “Script Kiddie Cookbook”
http://www.lulu.com/learningbydoing

This book title is based upon the Anarchist Cookbook that came out in the 1980’s. In that
tremendously popular book it showed you how to build bombs at home. Similarly the
Script Kiddie Cookbook will show you step by step hacking and hacking methodology.
You will go to jail if you use this information improperly!!

Learn the basics about computer security with this book. This book assumes you know
nothing about security and starts there. All tools and software used are freeware. Other
books will build upon the materials in this book. Learn about the nuts and bolts of
SPAM, port scanning, Knoppix tools, cookies, and bring it all together with exercises in
hacking at legal gaming sites.

GET THE BOOK THE GOVERNMENT DOES NOT WANT YOU TO OWN!!!
Switching Review

Management/Theory
1. What steps enable you to view a switch’s configuration in a web browser?
2. What steps enable you to view a switch’s configuration in a telnet session?
3. What command is used to assign a default gateway address to a 2950 switch?
4. What are three features of full-duplex Ethernet?
5. What are the three layers of Cisco’s hierarchical design?
6. At what layer of the hierarchical design are users and their equipment found? For inter-VLAN routing? For combining traffic from multiple IDF’s?
7. How can collisions be reduced in a network filled with hubs?
8. What is the command for statically adding a MAC address to a 2950 switch table for a specific Ethernet port?
9. On a catalyst switch what do the colors green, amber, and red mean on the SYSTEM LED’s?
10. What are the advantages and disadvantages between cut-through switching and store-and-forward switching? Fragment free?

STP
1. What is spanning tree? What are its features?
2. How can you “force” the election of the STP root bridge if all switches are set to their defaults?
3. What is the first step in the spanning-tree process (meaning the election of root bridges, not the states of the ports)?
4. What is used first when selecting a switch as the root bridge?
5. What are the four states of spanning tree and how can you view the status of a port?
6. Give a brief explanation of what each state in a spanning tree performs.

VLAN
1. What is designated by default as the “management VLAN”? Is it possible to delete the management VLAN?
2. Do multiple VLAN’s on a switch have the same subnet number or different subnet numbers and why?
3. What commands are used on a 2950 switch to configure a port and assign it to a VLAN?
4. What is the minimum sequence of commands to allow interVLAN routing between two VLAN’s on a 2950 switch connected to a router using sub-interfaces on one fast Ethernet port?
5. How many broadcast domains exist on a switch with only one VLAN?
6. Without using VLAN’s what are two advantages of using switches in a network design?

VTP
1. For what is trunking used?
2. For what does VTP use the management VLAN?
Part 3: 
LAN Routing Fundamentals

They said “I was crazy!”

How do you like the idea of free and low-cost textbooks? Well here is how you can help me in the effort to bring more of these types of books to you. They say there is safety in numbers, well I can promote this type of textbook production way better if I hear from you personally. Send me an email at Basham.Matt@spcollege.edu just to tell me you are using the book, you like the idea of cheaper high-quality books, or to just say “hello.” Together we will change education for the better!
Basic Router Commands

Objective:
To become familiar with basic router commands including how to get help.

Background:
In this lab we take you into the mysterious world of the router. You kind of messed around with it before with the Hyperterminal lab, but eventually you knew you would be learning by doing. In this lab you will become familiar with the help commands, the types of prompts you will use, and some basic router commands.

Lab Design:

--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: n

5. If you put in “yes” then you will be able to set up your router using “menu-based” commands. But you didn’t come here to learn how to do anything menu-based. The menu-based commands are severely limited so you need to learn about command line interfaced (CLI) configuration anyways so you might as well dive right in! Put in either “no” or “n” (without the quote marks) and press enter. Also put in “yes” for terminating autoinstall. You should see something similar to:

Would you like to terminate autoinstall? [yes/no] y

Press RETURN to get started!
6. Well the next step should be very obvious…press RETURN to get started. You should see a bunch of messages flashing and scrolling down the hyperterminal session. When it stops, press enter, you should see something like this:

```
00:00:51: %SYS-5-RESTART: System restarted --
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-DS-M), Version 12.0(13),
RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Wed 06-Sep-00 02:30 by linda
Router>
```

This is known as the “user” prompt. You can tell the router prompt is in the user mode because the name (also known as the “host name”) of the router is followed by a carat “>”. This mode allows anyone to see a very limited amount of information about the status of the router. At this prompt you will not be able to change the programming of the router.

7. To see what options are available for us at the user prompt we can “ask” our router for help. Computer devices are like that…if we get stuck, then we can ask it for help. On your workstation if you want some help then you can use your pull-down menus or even use the task bar help option (Start>help). Routers are helpful too. The phrase “easy when you know how” really applies. To get help you should start with the generic “help.” Then press enter.

```
router>help
```

You should see something like this:

```
Help may be requested at any point in a command by entering a question mark '?'. If nothing matches, the help list will be empty and you must backup until entering a '?' shows the available options.
Two styles of help are provided:
1. Full help is available when you are ready to enter a command argument (e.g. 'show ?') and describes each possible argument.
2. Partial help is provided when an abbreviated argument is entered and you want to know what arguments match the input (e.g. 'show pr?'.)

Router>
```

8. Ok…so that didn’t give you much. Most computers or network systems the command “help” works very well. So remember it and use it when appropriate. There is a better way to get help using the question mark on Cisco devices. Try typing this (and press enter):

```
router>?
```
9. Write down what you see on the worksheet entitled “user mode ? options.” Some of the commands you will using more than the others. Which one do you think they are and why? Don’t just quickly turn and start jotting them down from the answers…with routers you should take your time, examine everything twice, and examine the outcome. With router programming speed kills. If you see a line that says:

--------More--------

Then the router is waiting for you to press enter to continue. This just stops what’s on the screen for you to be able to read it. If you hit any other key it will take you back to the prompt without showing the rest of the information.

10. Let’s try using a couple of those commands.

11. Now let’s move on to the next type of prompt: the privileged mode prompt. To get to the privileged prompt you need to type either “enable” or “en” for a shortcut. Many commands can be short-cutted but for now get used to using the entire command. As you progress through these labs and get comfortable with the commands then you can start abbreviating the commands.

    router>enable
    router#

Notice how the prompt changes from a carat to a pound sign. This is a visual cue to you that you are at the privileged mode prompt. To switch back to the user mode prompt simply type “disable.” Actually you can also type “exit” here and it will do the same thing, but “disable” is the technically most correct answer for how to get from the privileged mode prompt to the user mode prompt. Try both and see for yourself. Sounds like a test question don’t it?

12. Now let’s get back to exploring the privileged mode prompt command options. Just like we did at the user mode prompt we can request help for seeing all available command options with a question mark:

    router#$

13. Write down what you see on the worksheet entitled “privileged mode ? options.” Like the user mode prompts some of the commands you will using more than the others. Which one do you think they are and why? In the answers for this lab I have also highlighted the ones you will be more likely to use than the others.

14. Let’s try using a couple of those commands. Type “show run” and look at the output. This is actually the current running configuration script for your router. You will learn more about this in the next couple of labs. Then type “reload” and hit enter. You will need to hit enter one more time and the system will “reload” or in geek terms it will “reboot.” (Sounds Canadian, but its not.)

15. Ok…time to learn about shortcuts with router commands. I know, I know. I said they should not be used because speed kills…these are designed to help you more accurately work with your router. You can use the up and down arrows to view the previous commands. We did this earlier in part 1 with our workstation DOS
prompt and the DOSKEY commands. If you do not see anything when you use
the up arrow it may because you have not used any commands at that specific
prompt mode. Next, lets look at some keystroke shortcuts. Suppose you typed a
command similar to what you need to use next. Ping will be a good example
here…suppose we wanted to ping to destinations 192.168.1.1 and 192.168.1.2.
We could try it this way:

```
router#ping 192.168.1.1 (typed)
router#ping 192.168.1.2 (typed)
```

or we could do it this way:
```
router#ping 192.168.1.1 (typed)
router#ping 192.168.1.1 (used the up arrow)
router#ping 192.168.1.1 (back space one character)
router#ping 192.168.1.2 (typed in a “2”)
```

In this manner we used less keystrokes and we have reduced the possibility of a
typing error on the second ping command. These types of short cuts are ok. You
can use keystroke commands to move back and forth more quickly on the
command line. I use the control+a and control+e with my up arrow quite
frequently. Plus these combinations also sound like some mighty fine fodder for a
certification exam, don’t they? Hint, hint, wink, wink, nudge, nudge, know what I
mean, know what I mean? Fill in the chart below on keystroke shortcuts and what
they do.

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control+a</td>
<td>Completes the entry</td>
</tr>
<tr>
<td>Control+b</td>
<td></td>
</tr>
<tr>
<td>Escape+b</td>
<td></td>
</tr>
<tr>
<td>Control+e</td>
<td></td>
</tr>
<tr>
<td>Control+f</td>
<td></td>
</tr>
<tr>
<td>Escape+f</td>
<td></td>
</tr>
<tr>
<td>Control+n</td>
<td></td>
</tr>
<tr>
<td>Control+p</td>
<td></td>
</tr>
</tbody>
</table>

16. Another way to view the progression of commands is using the “show history”
command. The up arrow will only show you those commands one at a time, but
```
router#show history
```
the show history will show you the last 15 commands (default) you used. Heck,
you can even change how many previous commands will be stored. Lets try that
now:
```
router#terminal history size 5
```
17. Using this command will set the number of commands retained in the history buffer to 5. If you were to “show history” then you would see the previous 5 commands. This number can range from 0 to 256. (Sounds like a good CCNA question doesn’t it?).

18. Ok. We are still moving with our prompts. Before we can make any changes to our router we need to be at the configuration mode prompt.

```
router#config
Configuring from terminal, memory, or network [terminal]?
router#terminal
router(config)#
```

Or we can just by-pass that second statement by combining the two statements:

```
router#config t
router(config)#
```

19. Let’s change the name of our router. We do this from the privileged mode prompt using the command “hostname.” Let’s change it to our name.

```
router(config)#hostname matt
matt(config)#
```

Notice how the prompt changes immediately to our new hostname.
To leave the configuration mode, just type exit.

```
matt(config)#exit
matt#
```

20. It would be a shame if the power were to suddenly get turned off because everything would be erased. We can save our work to the file that is loaded when our router starts. Right now our changes are in a file called “running-configuration.” Here we can type in some changes and see if those changes have the desired effects. If they don’t then we can reverse those changes or even reboot the router (which would load the “start-configuration” file). Suppose we like what we have done. Then we just have to copy our running-configuration file to our start-configuration file. True. It does over write our start-configuration file, but that is what we want to do. Let’s try it.

```
matt#copy running-configuration start-configuration
```

Boy is that a lot to type…just to make it easier you can also type this:

```
matt#copy run start
```

Be very careful to type this in exactly. Sometimes I get typing too quickly and I type copy runs tart and hit enter quickly without looking at what I am doing. Voila poof! I have totally wrecked my files and the operating system needs to be
totally re-loaded. You can see why speed can kill. CISCO has many versions of
its operating system. The one you are using is probably a derivative of version
12. Some of the older commands from previous versions still work with version
12 but do not show up in your help menus. One really helpful command that
duplicates the copy run start is the “write memory” command. All you have to
type is “wr” and the router automatically copies the running configuration file
over the start configuration file. Now you have no chance of messing up the
router operating system with misspelled copy commands.

   matt#wr

21. Thought you were done with prompts? Nope. One other type of prompt is called
the “global mode prompt.” From here we make changes to various parts of the
router. For example, when we want to configure an interface we first must be in
the “interface global mode prompt.” I know, lots of jargon. It really makes more
sense after you have done it a couple of times. Let’s look at the various types of
global mode prompts and the sequence from the user mode prompt we took to get
here (you do not have to type these in…just look at them):

   matt>
   matt>en
   matt#config t
   matt(config)#

   Interface
   matt(config)#interface e0/0
   matt(config-if)#

   Sub-interface
   matt(config)#interface e0/0.1
   matt(config-subif)#

   Router
   matt(config)#router rip
   matt(config-router)#

   Console line
   matt(config)#line vty 0 4
   matt(config-line)#

Your interface name and number can vary with your model. For example the
2500 routers use “e0” for the first Ethernet. The 2610 and 2611’s use “e0/0” and
the 2620 and 2621’s use “fa0/0.” Just learn by doing. You can also use the show
interface command (be sure you are not in config mode) too.

Other modes you may use include: controller, map-list, route-map, ipx router and
map-class. Use your knowledge of help commands to figure out what those
prompts would look like.
**Supplemental Labs or Challenge Activity:**

1. Try going through the initial configuration setup (put in yes instead of no). See how it differs. Its actually very nice but don’t get too attached to it. You will learn more by configuring your router using the command line interface (CLI).
2. What are the other options available with the copy command?
3. If your router is already started then how would you get the router setup script back?

**So What Have I Learned Here?**

In this lab you learned basic router steps including getting help. I guarantee you will be using the help function many more times during your training. In the next lab we will look a bit deeper into how the router boots. I actually had a lot of fun writing that one…there is some information in there you won’t find in any books or documentation anywhere.

**User Mode ? Options**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Privileged Model Options

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Objectives:
To more fully understand how routers hardware and software work together.

Tools and Materials:
CISCO router
Workstation with Hyperterminal program
(1) rollover cable

Background:
All routers essentially have 5 types of logic processors: CPU, FLASH, ROM, RAM, and NVRAM. Finding out information about these devices from CISCO or on the Internet is problematic, to say the least. Here we will discuss the block diagram of generic routers, how to identify the components on 2500/2600 router boards (if you dare to open one up), and how to “see” those processes during the router boot sequence.

Let’s start by looking at a generic block diagram of a router:

```
CPU      FLASH    ROM      RAM       NVRAM
BUS
Network Modules
I/O Port   I/O Port
MSC      MSC
```

CPU-Central Processing Unit (usually Motorola)
FLASH-Holds image of OS (ROM-type) (does not erase when off)
ROM-Holds POST and Bootstrap Programs (does not erase when off)
RAM/DRAM-Holds routing tables, packet buffering, etc. (erases when off)
NVRAM-Holds configuration files (does not erase when off)
MSC-Media Specific Converter

The bus is simply a central transmission point for our bits. The top row of components is physically attached to the motherboard (CPU, NVRAM, etc). The bottom row of components (I/O Port MSC) is the Ethernet, Aux., serial connections, etc. Notice there are no “moving” parts in a router. Computer hard drives have moving parts, which require frequent replacement. Since routers do not have any moving parts they are said to “last longer.” Let’s turn our discussion to the boot sequence and how all of these components inter-relate with the software by looking at a boot sequence block diagram:
Power “on”

POST

Bootstrap

Check for Config file

Locate OS

Load OS

Locate Config

Found

Not found

Load Config

setup mode

Initialize Router config

POST

Bootstrap

Check for Config file

Locate OS

Load OS

Locate Config

ROM

ROM

Confreg (finds the location of the OS)
[Looks in NVRAM>Flash>TFTP]

From ROM into FLASH

From ROM into FLASH
into RAM/DRAM (decompresses)

ROM>NVRAM>RAM/DRAM
This config sets up the interfaces after the OS is loaded

Found

Not found

Load Config

setup mode

Initialize Router config
So let’s look at a boot sequence with Hyperterminal. (My comments appear in cursive writing with italics and highlighting.)

1. **Power on.**
2. **ROM-runs power on diagnostics (you will see some lights on the router blink).**
3. **ROM-runs bootstrap program version 11.3 here (do not confuse this with the IOS version).** You should see the following with your Hyperterminal session:

```
System Bootstrap, Version 11.3(2)XA4, RELEASE SOFTWARE (fc1)
```

Copyright (c) 1999 by cisco Systems, Inc.
TAC:Home:SW:IOS:Specials for info
C2600 platform with 24576 Kbytes of main memory

4. **ROM-directs Flash to load the IOS image from Flash into RAM/DRAM to be de-compressed.**

```
program load complete, entry point: 0x80008000, size: 0x56c7ac
Self decompressing the image:
#################################################
########################################################################
########################################################################
########################################################################
########################################################################
########################################################################
########################################################################
######## [OK]
```

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR sec. 52.227-19 and subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS sec. 252.227-7013.

cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-DS-M), Version 12.0(13), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Wed 06-Sep-00 02:30 by linda
Image text-base: 0x80008088, data-base: 0x80A065AC
cisco 2610 (MPC860) processor (revision 0x203) with 21504K/3072K bytes of memory.
Processor board ID JAD03428529 (932999778)
M860 processor: part number 0, mask 49
Bridging software.
X.25 software, Version 3.0.0.
1 Ethernet/IEEE 802.3 interface(s)
2 Serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read/Write)

5. Active configuration file (startup.cfg) is loaded from NVRAM into RAM/DRAM along with any active network maps or tables into RAM/DRAM. (none here...so system configuration dialog is displayed, but not requested). These include routing tables, ARP caches, fast-switching cache, packet buffering (shared RAM), and packet hold queues.

--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: n

Press RETURN to get started!

00:00:15: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
00:00:15: %LINK-3-UPDOWN: Interface Serial0/0, changed state to down
00:00:15: %LINK-3-UPDOWN: Interface Serial0/1, changed state to down

Lab Diagram:

Step-by-Step Instructions:
1. Hook up a router to a workstation and watch the steps as the router boots.
2. Try the show commands and fill in the description of what each does. Which type of processor (CPU, NVRAM, FLASH, RAM/DRAM, ROM) does each command
reside within? Which one (s) do you think you will be using the most, least, and why? How do you find out what is in ROM?

<table>
<thead>
<tr>
<th>Command</th>
<th>description</th>
<th>processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>sh buf (show buffers)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sh fla (show flash)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sh int (show interface)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sh mem (show memory)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sh pro (show processes)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sh prot (show protocols)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sh ru (show run)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sh start (show start)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sh stacks (show stacks)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sh tech (show tech)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sh ver (show version)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sh ip route</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. Let’s look at a basic router script (use show run). My comments are in **highlighted handwriting**:

Router#sh ru
Building configuration...
Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption

hostname Router
!
memory-size iomem 15
ip subnet-zero
!
interface Ethernet0/0
no ip address
no ip directed-broadcast
shutdown
!
interface Serial0/0
no ip address
no ip directed-broadcast
no ip mroute-cache

interfaces Serial0/1
packets
shutdown
!
interface Serial0/1

Our IOS version
enables time stamping
enables time stamping
disables service
Our router name
sets memory i/o to 15
Let’s us use subnet zero!
interface
ip address for this interface
drops ip directed broadcasts
(good...prevents Denial of Service attacks)
Note: shut down by default
Disables fast switching of IP packets

307
no ip address
no ip directed-broadcast
shutdown
!
ip classless    Disables classless routing behavior
!
line con 0
  transport input none
line aux 0
line vty 0 4
!
no scheduler allocate
end

Router#

Supplemental Lab or Challenge Activity:
1. Repeat this lab with a pre-configured router and watch for changes.
2. How do you think changing the configuration register would affect the boot sequence?
3. Go out to www.cisco.com and try to find configuration register settings to alter the way the boot sequence happens with your router.
4. Several of the explanations above are numbered1-5. Go out to CISCO’s website and find out what they do.
5. Two commands were included by default on your router script above (transport input none and no scheduler allocate). Go out to CISCO’s website and find out what they do.

So What Have I Learned Here?
In this lab you learned about the router boot sequence. Your textbook will also show you some methods for changing how the router boots, loading IOS’s and other stuff. I have also attached some diagrams of motherboards for CISCO 2500 and 2600 routers. You probably won’t find those anywhere…let’s just say a little inside bird told me about this. Here are some really good, but technical books, if you want some more information.


Are you enjoying the materials? Well be on the lookout for some other manuals and textbooks on
http://www.lulu.com/learningbydoing and http://www.spcollege.edu/star/cisco

The “Script Kiddie Cookbook”
http://www.lulu.com/learningbydoing

This book title is based upon the Anarchist Cookbook that came out in the 1980’s. In that tremendously popular book it showed you how to build bombs at home. Similarly the Script Kiddie Cookbook will show you step by step hacking and hacking methodology. You will go to jail if you use this information improperly!!!

Learn the basics about computer security with this book. This book assumes you know nothing about security and starts there. All tools and software used are freeware. Other books will build upon the materials in this book. Learn about the nuts and bolts of SPAM, port scanning, Knoppix tools, cookies, and bring it all together with exercises in hacking at legal gaming sites.

GET THE BOOK THE GOVERNMENT DOES NOT WANT YOU TO OWN!!!
CISCO 2600 series Router Motherboard Configuration
Basic Router Configuration

Objectives:
To learn a method for configuring basic router commands that you will use many times.

Background:
During the course of your CCNA studies you will be setting up many routers with many different router configurations. It is a good idea to learn to set up routers in “steps.”
   Step 1—start with setting up the “router basics”
   Step 2—configure interfaces
   Step 3—configure routing protocols
   Step 4—add any other items (ACL’s, security, routes, etc)
In this lab you will learn about step 1: configuring the router’s name, configuring vty lines, console lines, and setting up passwords.

Tools and Materials:
CISCO router
Workstation with Hyperterminal program
(1) Rollover cable (ro)

Cabling diagram:

Step-by-Step Instructions:
1. Boot up the router and do not use the setup program. Oh sure, setup is easy, but you need to learn it all from the command line. Enter the privileged mode:

   \[\text{Router}>\text{enable}\] (or just “\text{en}”)
   \[\text{Router}#\]
   since no enable password is set yet, the router does not ask for a password

2. Enter configuration mode:

   \[\text{Router}#\text{configure}\] (or just “\text{config t}”)
   \[\text{Router}#\text{terminal}\]
   \[\text{Router(config)}#\]
3. Configure the router’s name to be RouterA:

```
Router(config)##hostname RouterA
RouterA(config)#
```

(note: the name changes immediately)

4. Configure the vty lines with a password “cisco.” These are the available Telnet ports for use from the Internet or from other networking devices on your network. Without a password no one will be able to telnet into the router.

```
RouterA(config)#line vty 0 4
RouterA(config-line)#password cisco
RouterA(config-line)#login
RouterA(config-line)#exit
```

5. Configure the console line so messages will not interrupt what you are typing and so your session does not time out:

```
RouterA(config)#line con 0
RouterA(config-line)#logging synchronous
RouterA(config-line)#exec-timeout 0 0
RouterA(config-line)#exit
```

Feeling frisky? Change `exec-timeout` to 0 1. This will cause your router session to time out every 1 second (it can take up to about 5 minutes to start though). There are only two ways to fix it: router recovery or press the “down” arrow key while you change the exec timeout to a higher number with your other hand at the same time. Doing this generates a continuous interrupt request to the CPU and the session, therefore, does not time out. Logging synchronous is a nice command. When you are configuring a router sometimes messages will interrupt your work. Without this command in your script when you are interrupted you will have to remember exactly what you typed when you were interrupted. With this command the router will “refresh” what you typed on the current line.

6. Configure the secret password “cisco” and the enable password “class.” The enable secret password is used to get from user mode into privileged mode. The enable password is something that shows up from time to time on tests and whether you know how to configure it or not.

```
RouterA(config)#enable secret cisco
RouterA(config)#enable password class
```

7. To see what you have done so far you can always look at the running-configuration file:

```
RouterA(config)#exit (or use control+Z to get all the way out)
RouterA#sh ru (short for show run)
```
8. Once you have determined that your configuration is what you would like on your router you need to save it to your startup-configuration file. Otherwise if your router is re-booted or you loose power then your configuration will be lost.

   RouterA#copy ru start (or wr)

9. Great. Now you know how to save your configuration. But what if someone else saved a configuration and you want to get rid of it? Do this:

   RouterA#erase start (to erase the startup-configuration file)
   RouterA#reload

10. So what if you made a mistake when you are typing something? Some things you can just re-type and they will be changed (like hostname) and some others you can un-do just by typing the word “no” and repeating the errant command.

   RouterA(config)#hostname mark (darn! We wanted “matt”)
   Mark(config)#hostname matt (just type in “matt”)
   Matt(config)#

   matt(config)#line vty 0 4
   matt(config-line)#password csico (darn! We wanted “cisco”)
   matt(config-line)#no password csico
   matt(config-line)#password cisco

Supplemental Lab or Challenge Activity:
1. Don’t have a router to practice this on at home? Just practice writing out this script over and over on paper. Don’t forget to write the prompts…they are important to know too.

   Router>en
   Router#config t
   Router(config)#hostname RouterA
   RouterA(config)#line vty 0 4
   RouterA(config-line)#password cisco
   RouterA(config-line)#login
   RouterA(config-line)#exit
   RouterA(config)#line con 0
   RouterA(config-line)#logging synchronous
   RouterA(config-line)#exec-timeout 0 0
   RouterA(config-line)#exit
   RouterA(config)#enable secret cisco
   RouterA(config)#enable password class

2. Security/Hacking Tip on VTY lines: Port scans (which are legal) on your network can reveal ports 2000, 2001, 4000, 4001, 6000, or 6001 ports in use. These are reserved for CISCO routers. Yup…knowing which type of equipment is in use is beneficial to hackers. Most CISCO network administrators have it “drummed in their heads” that there are only 5 vty lines available (and, for you people studying for the CCNA there are only 5) but, enterprise versions of routers have up to 1000
or so vty lines possible. Knowing a CISCO device exists and knowing most
admins do not know about those “upper” vty lines creates security holes. For
eexample, if I open up 6 simultaneous vty session with Telnet to a CISCO
device...

  Session 1>open vty 0 > password requested
  Session 2>open vty 1 > password requested
  Session 3>open vty 2 > password requested
  Session 4>open vty 3 > password requested
  Session 5>open vty 4 > password requested
  Session 6>open vty 5 > no password required=keys to the kingdom!

To find out how many vty lines you have type this:

```
Router>en
Router#config t
RouterA(config)#line vty 0 ?
```

3. Want to keep people from walking up to your session and making changes? Put a
password on it. Try to figure out how to do that.

4. You can actually cheat this a bit and dump a notepad file into a router. Let’s start
by looking at our prompts and what we typed in:

```
Router>en
Router#config t
Router(config)#hostname RouterA
RouterA(config)#line vty 0 4
RouterA(config-line)#password cisco
RouterA(config-line)#login
RouterA(config-line)#exit
RouterA(config)#line con 0
RouterA(config-line)#logging synchronous
RouterA(config-line)#exec-timeout 0 0
RouterA(config-line)#exit
RouterA(config)#enable secret cisco
RouterA(config)#enable password class
```

Now then, if we remove the prompts we basically have a notepad file:

```
en
config t
hostname RouterA
line vty 0 4
password cisco
login
exit
line con 0
logging synchronous
exec-timeout 0 0
exit
enable secret cisco
enable password class
```
When doing this you generally do not want to abbreviate too many commands…it can really mess things up when you have an inaccurate command or forget to put an exit here or there. Practice makes perfect. For now you should probably forget about this little tip and just get used to typing all of this in by hand.

*So What Have I Learned Here?*

In this lab you have learned how to set up the basics on a router. You will be using this information pretty much for every lab left in this book. After a while this will become automatic to you. In the next lab we will put this to use by learning about our first routing protocol: RIP.

---

**Looking for college courses?**

**We deliver.**

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education.

Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

St. Petersburg College’s eCampus

[http://e.spcollege.edu](http://e.spcollege.edu)

Your time. Your place. Your future.
Basic Rip

Objectives:
To learn about the Routing Information Protocol (RIP version 1).

Background:

Tools and Materials:
(2) PC/workstations
(2) Routers
(2) Switches
(4) Straight-through cables
(1) DCE serial cable
(1) DTE serial cable
(2) rollover cables

Lab Diagram:

Addressing:

<table>
<thead>
<tr>
<th>Routers</th>
<th>Hostnames</th>
<th>E0</th>
<th>S0</th>
<th>S1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>192.168.3.1/24</td>
<td>192.168.30.1/24 (DCE)</td>
<td>n/a</td>
</tr>
<tr>
<td>Randy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ward</td>
<td></td>
<td>192.168.4.1/24</td>
<td>n/a</td>
<td>192.168.30.2/24 (DTE)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workstations</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>192.168.3.2</td>
<td>192.168.4.2</td>
</tr>
<tr>
<td>SM</td>
<td>255.255.255.0</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>GW</td>
<td>192.168.3.1</td>
<td>192.168.4.1</td>
</tr>
</tbody>
</table>
**Step-by-Step Instructions:**
In our last lab on setting up router basics we learned there are four steps to setting up a router: (1) setting up the router basics, (2) configuring the interfaces, (3) configuring the routing protocols, and (4) adding in the fancy stuff. In this lab we will teach you steps two and three.

1. Cable the lab as shown. Be certain your serial cable is plugged in properly...in other words the DCE end goes with the interface command “clockrate.”
2. **STEP 1.** Set up the “router basics” on each router.

   ```
   Router>en
   Router#config t
   Router(config)#hostname Randy     (or hostname Ward)
   Randy(config)#line vty 0 4
   Randy(config-line)#password cisco
   Randy(config-line)#login
   Randy(config-line)#exit
   Randy(config)#line con 0
   Randy(config-line)#logging synchronous
   Randy(config-line)#exec-timeout 0 0
   Randy(config-line)#exit
   Randy(config)#enable secret cisco
   Randy(config)#enable password class
   ```

3. **STEP 2.** Configure the interfaces on each router:

   ```
   Randy(config)#int e0
   Randy(config-if)#ip address 192.168.3.1 255.255.255.0
   Randy(config-if)#no shut
   Randy(config)#int s0
   Randy(config-if)#ip address 192.168.30.1 255.255.255.0
   Randy(config-if)#clockrate 56000
   Randy(config-if)#no shut
   ```

   ```
   Ward(config)#int e0
   Ward(config-if)#ip address 192.168.4.1 255.255.255.0
   Ward(config-if)#no shut
   Ward(config)#int s1
   Ward(config-if)#ip address 192.168.30.2 255.255.255.0
   Ward(config-if)#no shut
   ```

   Now here is an age-old dilemma for Cisco testing. Is the command “clockrate” or “clock rate” (with a space)? Both will work on the routers but only one is correct on the Cisco test. Good luck.

4. **STEP 3.** Configure the routing protocol and advertise/associate/publish (I have seen it written these three different ways on tests) the router’s networks.

   ```
   Randy(config)#router rip
   Randy(config-router)#network 192.168.30.0
   Randy(config-router)#network 192.168.3.0
   ```

   ```
   Ward(config)#router rip
   Ward(config-router)#network 192.168.30.0
   Ward(config-router)#network 192.168.4.0
   ```
5. Setup the workstations with IP address, subnet masks, and gateways addresses. You will need to reboot the workstations. If they ask for a password for network connectivity just put anything in and you should see a message something like “no domain server is available, you may not have some networking functions.” It’s ok if you see it, but you probably will not be able to ping outside of your workstation without seeing that error message. A quirk with Microsoft.

6. Test connectivity from router to router (from the router) by using ping from Randy to Ward. You should see:

   RouterA#ping 192.168.30.2
   Type escape sequence to abort.
   Sending 5, 100-byte ICMP Echos to 192.168.30.2, timeout is 2 seconds:
   !!!!!
   Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms
   RouterA#

7. Test connectivity from workstation to workstation (from DOS) by using ping from workstation A to workstation B. You should see:

   C:\WINDOWS\Desktop>ping 192.168.4.2
   Pinging 192.168.4.2 with 32 bytes of data:
   Reply from 192.168.4.2: bytes=32 time=21ms TTL=126
   Reply from 192.168.4.2: bytes=32 time=20ms TTL=126
   Reply from 192.168.4.2: bytes=32 time=21ms TTL=126
   Reply from 192.168.4.2: bytes=32 time=21ms TTL=126
   Ping statistics for 192.168.4.2:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
   Approximate round trip times in milli-seconds:
   Minimum = 20ms, Maximum = 21ms, Average = 20ms
   C:\WINDOWS\Desktop>

8. Let’s see our route from workstation A to workstation B (from DOS). You should see:

   C:\WINDOWS\Desktop>tracert 192.168.4.2
   Tracing route to STAR10616119 [192.168.4.2]
   over a maximum of 30 hops:
   1  1 ms  1 ms  1 ms 192.168.3.1
   2  25 ms 25 ms 25 ms 192.168.30.2
   3  30 ms 30 ms 30 ms STAR10616119 [192.168.4.2]
   Trace complete.
   C:\WINDOWS\Desktop>

   So does this mean we went three hops? Nope, we only went 2 hops. The last one does not count as a hop…we will see this more in a bit. Be patient…it’s a virtue.
9. All good? Ok…now let’s have some fun with a challenge! Let’s see if we have some neighbors using our CISCO Discovery Protocol, a.k.a. CDP (enabled by default at boot). CDP is a layer 2 protocol (good test question too). You should see:

```
Randy>sh cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                   S - Switch, H - Host, I - IGMP, r - Repeater
Device ID  Local Intrfce    Holdtme Capbility Platform  Port ID
Ward        Ser 0/0      128       R      2610     Ser 0/1
```

Notice with CDP we can see the identification (Ward), address/interface type (Ser 0/0), and platform (2610) of our neighbors. If we do not want to run CDP on all of our interfaces use the “no cdp run” command. Let’s see if we have any CDP traffic being generated. CDP updates every 60 seconds by default. You should see something like this (but the numbers probably won’t match):

```
Randy>sh cdp traffic
CDP counters :
   Packets output: 82, Input: 63
  Hdr syntax: 29, Chksum error: 0, Encaps failed: 9
   No memory: 0, Invalid packet: 0, Fragmented: 0
```

We can see our CDP packets coming and going. We’ll look at that other stuff later. Let’s use the protocols command to see what we have. You should see:

```
Randy>sh protocols
Global values:
   Internet Protocol routing is enabled
   Ethernet0/0 is up, line protocol is up
      Internet address is 192.168.3.1/24
   Serial0/0 is up, line protocol is up
      Internet address is 192.168.30.1/24
   Ethernet0/1 is administratively down, line protocol is down
      Serial0/1 is administratively down, line protocol is down
```

This is good…IP is running and our interfaces are up. E0/1 is down because we didn’t configure it.
10. Let’s look at our path or “route” from one router to another: You should see:

```
Randy>sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP, D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2, E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP, i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default U - per-user static route, o - ODR

Gateway of last resort is not set

C    192.168.30.0/24 is directly connected, Serial0/0
R    192.168.4.0/24 [120/1] via 192.168.30.2, 00:00:07, Serial0/0
C    192.168.3.0/24 is directly connected, Ethernet0/0
Randy>
```

We see our directly connected routes and the one learned via our routing protocol. Getting stuck? Try using this command at the privileged prompt: `clear ip route *` several times on each router to “restart” the routing process (clears the tables, sends updates, receives updates, and re-creates the ip routing table). This is a really good command to remember and keep in your “arsenal.”

11. Let’s watch ICMP packets as they pass from one router to another. Turn on debug, then ping and trace route from the workstation to generate icmp “traffic.” Side note: debug can really chew up resources. Be sure to use just enough debug to get the job done, then turn off debug. Notice how we had to change user modes:

```
Randy#debug ip icmp (use “undebug ip icmp” or “undebug all” to turn off)
```

This is what I sent:

```
C:\WINDOWS\Desktop>ping 192.168.4.2
Pinging 192.168.4.2 with 32 bytes of data:
Reply from 192.168.4.2: bytes=32 time=23ms TTL=126
Reply from 192.168.4.2: bytes=32 time=20ms TTL=126
Reply from 192.168.4.2: bytes=32 time=20ms TTL=126
Reply from 192.168.4.2: bytes=32 time=20ms TTL=126
Ping statistics for 192.168.4.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
        Minimum = 20ms, Maximum = 23ms, Average = 20ms
C:\WINDOWS\Desktop>tracert 192.168.4.2
Tracing route to STAR10616119 [192.168.4.2]
over a maximum of 30 hops:
        1    2 ms   1 ms   1 ms  192.168.3.1
        2    25 ms  25 ms  25 ms  192.168.30.2
        3    30 ms  30 ms  30 ms  STAR10616119 [192.168.4.2]
Trace complete.
C:\WINDOWS\Desktop>
```
You should see on RouterA:

Randy#**debug ip icmp**
ICMP packet debugging is on
Randy#
01:02:29: ICMP: time exceeded (time to live) sent to 192.168.3.2 (dest was 192.168.4.2)
01:02:29: ICMP: time exceeded (time to live) sent to 192.168.3.2 (dest was 192.168.4.2)
01:02:29: ICMP: time exceeded (time to live) sent to 192.168.3.2 (dest was 192.168.4.2)
01:02:29: ICMP: dst (192.168.3.1) port unreachable sent to 192.168.3.2
01:02:31: ICMP: dst (192.168.3.1) port unreachable sent to 192.168.3.2
01:02:32: ICMP: dst (192.168.3.1) port unreachable sent to 192.168.3.2
Randy#

So this is confusing…our times are exceeded, our ports are unreachable, but our icmp’s still worked. Something for you to think about.

12. Let’s see the RIP updates (sent every 30 seconds by default) as they pass through our routers (more on updates and timers in another lab). You should see:

Randy#**debug ip rip**
RIP protocol debugging is on
Randy#
01:05:48: network 192.168.30.0, metric 1
01:05:48: network 192.168.4.0, metric 2
01:05:48: network 192.168.3.0, metric 1
Randy#

See I told you we only went 2 hops maximum! We can see this with our metric 1 and metric 2 statements.

13. We can use hostnames on our routers to make ping-ing a bit easier. Instead of using those long 32-bit IP addresses we can assign names to them. The order of input is important because the router will look at the first ip address, then the next, and so on, depending upon how many ip addresses you associate with a host name. Generally it is a good idea to put them in the order they are most likely to be used. I tend to put serial lines in front of Ethernet lines.

Randy(config)#**ip host ward 192.168.30.2 192.168.4.1**
OR
Randy(config)#**ip host wards0 192.168.30.2**
Randy(config)#**ip host warde0 192.168.4.1**

Ward(config)#**ip host randy 192.168.30.1 192.168.3.1**
OR
Ward(config)#**ip host randys1 192.168.30.1**
Ward(config)#**ip host randye0 192.168.3.1**
14. What does the “description” command do when you are configuring an interface?

Randy(config)#int e0/0
Randy(config-if)#description DCE serial to Ward DTE

Supplemental Lab or Challenge Activity:
1. What would you expect to see on Ward? Try steps 1-6 over again on Ward.
2. Try this with class “A” or “B” private or public IP addresses that you choose.
3. Try this lab with one class “A” private IP address for the Ethernet network on RouterA, a class “B” private IP address over the serial line, and a class “C” private IP address on the Ethernet network on RouterB.
4. Try mixing and matching private and public IP addresses.
5. What are the available commands for router rip? List them and give a brief description of each.

Router(config)#router rip
Router(config-router)#?

So What Have I Learned Here?
Got questions about RIP? Good! Hopefully the next few labs should help provide some clarity about this “eccentric” little routing protocol.

Guest Router Name Derivation

Ward Christensen and Randy Suess are generally attributed as creating the first Bulletin Board System (BBS) in 1978. The BBS site, located in Chicago, Illinois is still supposed to be in operation today.


Take college courses on a new schedule--yours.
Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education. Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

http://e.spcollege.edu
St. Petersburg College’s eCampus. Your time. Your place. Your future.
Basic Troubleshooting: Router-to-Router

Objectives:
To be able to learn the fundamentals of troubleshooting router-to-router connections.

Tools and Materials:
(2) routers
(2) switches
(2) workstations
(4) Straight-through cables
(2) rollover cables
(1) DCE cable
(1) DTE cable

Background:
This lab works with the same configuration from the last lab.

Lab Diagram:

Addressing:

<table>
<thead>
<tr>
<th>Routers</th>
<th>Hostnames</th>
<th>E0 192.168.3.1/24</th>
<th>S0 192.168.30.1/24 (DCE)</th>
<th>S1 n/a</th>
<th>Workstations</th>
<th>IP 192.168.3.2</th>
<th>SM 255.255.255.0</th>
<th>GW 192.168.3.1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Randy</td>
<td>192.168.4.1/24</td>
<td>n/a</td>
<td>192.168.30.2/24 (DTE)</td>
<td>A</td>
<td>192.168.3.2</td>
<td>255.255.255.0</td>
<td>192.168.3.1</td>
</tr>
<tr>
<td></td>
<td>Ward</td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td>192.168.4.2</td>
<td>255.255.255.0</td>
<td>192.168.4.1</td>
</tr>
</tbody>
</table>
Step-by-Step Instructions:

1. Troubleshooting goes along neatly with the OSI model. Just start at the bottom (Physical Layer) and work your way up. Step 1—check for lights on the interfaces. No lights? Then make sure they are plugged in and you have the right type of cable in the right place (DCE/DTE).

2. Let’s go to the data link layer. Check the clockrate, ip/masks, and encapsulation very, very carefully. Look for transposed numbers or incorrect masks. When all else fails...try typing “no shut” on each interface configuration. You would be amazed how many problems “no shut” can fix. Use the `sh int` and `sh run` command to check things.

<table>
<thead>
<tr>
<th>Line</th>
<th>Protocol</th>
<th>What it means</th>
</tr>
</thead>
<tbody>
<tr>
<td>UP</td>
<td>UP</td>
<td>everything is fine.</td>
</tr>
<tr>
<td>UP</td>
<td>DOWN</td>
<td>connection problems (check your cabling)</td>
</tr>
<tr>
<td>DOWN</td>
<td>DOWN</td>
<td>interface problems</td>
</tr>
<tr>
<td>AD.</td>
<td>DOWN</td>
<td>disabled...everything is wrong.</td>
</tr>
</tbody>
</table>

The above chart is a good thing to print out and keep handy near your routers. With `sh int` you should see:

Randy# `sh int`  
**Ethernet0/0** is up, line protocol is up  
  Hardware is AmdP2, address is 0002.fd45.ae60 (bia 0002.fd45.ae60)  
  Internet address is 192.168.3.1/24  
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255  
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)  
  ARP type: ARPA, ARP Timeout 04:00:00  
  Last input 00:04:50, output 00:00:00, output hang never  
  Last clearing of "show interface" counters never  
  Queueing strategy: fifo  
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops  
  5 minute input rate 0 bits/sec, 0 packets/sec  
  5 minute output rate 0 bits/sec, 0 packets/sec  
  235 packets input, 37677 bytes, 0 no buffer  
  Received 147 broadcasts, 0 runts, 0 giants, 0 throttles  
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort  
  0 input packets with dribble condition detected  
  616 packets output, 54789 bytes, 0 underruns  
  70 output errors, 0 collisions, 12 interface resets  
  0 babbles, 0 late collision, 0 deferred  
  70 lost carrier, 0 no carrier  
  0 output buffer failures, 0 output buffers swapped out  

Serial0/0 is up, line protocol is up  
  Hardware is PowerQUICC Serial  
  Internet address is 192.168.30.1/24  
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255

317
Encapsulation HDLC, loopback not set, keepalive set (10 sec)
Last input 00:00:00, output 00:00:03, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  562 packets input, 39641 bytes, 0 no buffer
Received 422 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  576 packets output, 38825 bytes, 0 underruns
  0 output errors, 0 collisions, 27 interface resets
  0 output buffer failures, 0 output buffers swapped out
  8 carrier transitions
DCD=up  DSR=up  DTR=up  RTS=up  CTS=up

Ethernet0/1 is administratively down, line protocol is down
Hardware is AmdP2, address is 0002.fd45.ae61 (bia 0002.fd45.ae61)
MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 252/255, load 1/255
Encapsulation ARPA, loopback not set, keepalive set (10 sec)
ARP type: ARPA, ARP Timeout 04:00:00
Last input never, output 00:52:54, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input packets with dribble condition detected
  69 packets output, 4140 bytes, 0 underruns
  69 output errors, 0 collisions, 0 interface resets
  0 babbles, 0 late collision, 0 deferred
  69 lost carrier, 0 no carrier
  0 output buffer failures, 0 output buffers swapped out
Serial0/1 is administratively down, line protocol is down
Hardware is PowerQUICC Serial
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation HDLC, loopback not set, keepalive set (10 sec)
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
Randy#

Let’s go back over some of those things I highlighted in this example. Note the output from a show interface command. Pay special attention to the contents of the first five lines…this is our “bread and butter” lines. Be sure you know what is on which line and which line is in which order. We see a note about “MTU.” This is the “maximum transmission unit.” If the router is requesting to send a packet larger than the receiving router’s MTU, then the sending router will fragment the outgoing information into allowable sizes. Isn’t that nice? They can get along. Notice the default encapsulation type on serial lines is HDLC. We will be changing this when we get to the WAN part. Finally we see a MAC address per interface (necessary for proper routing to different interfaces). Guess what? We can change this if we want…I wouldn’t worry about it right now. If a hacker gets a request from a device with a MAC address they can determine which company manufactured it. Remember OUI’s? Once I know it is a CISCO device I can port scan to narrow down the devices. Once I know what device it specifically is I can use my knowledge of that device, its security problems, and gain access to it!

3. Time for the network layer. Check to be sure the routing protocol is enabled and that you have the correct routing protocol enabled. Have you advertised/associated/published your networks properly? Test your router-to-router connectivity with ping or an extended ping command. Here is an example of using ping from the Randy console to Ward Ethernet interface:

    Randy#ping 192.168.4.1

You should see:

    Randy#ping 192.168.4.1

    Type escape sequence to abort.
    Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
    !!!!!
    Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms
Possible responses when using ping (from most likely to least likely response):

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>Successful</td>
</tr>
<tr>
<td>.</td>
<td>Timed out</td>
</tr>
<tr>
<td>U</td>
<td>Destination unreachable</td>
</tr>
<tr>
<td>&amp;</td>
<td>Packet Time to Live (TTL) exceeded</td>
</tr>
<tr>
<td>?</td>
<td>Packet type unknown</td>
</tr>
<tr>
<td>C</td>
<td>Congestion experienced during transit</td>
</tr>
<tr>
<td>I</td>
<td>Interruption of Ping packet</td>
</tr>
</tbody>
</table>

You can also do an extended ping. This let’s you “set” the parameters of the ping packet. Here is the same example using an extended ping and what you should see (notice how I switched it to 7 icmp packets being sent):

```plaintext
Randy#ping
Protocol [ip]:
Target IP address: 192.168.4.1
Repeat count [5]: 7
Datagram size [100]: 1000
Timeout in seconds [2]: 4
Extended commands [n]: n
Sweep range of sizes [n]: n
Type escape sequence to abort.
Sending 7, 1000-byte ICMP Echos to 192.168.4.1, timeout is 4 seconds:
!!!!!!!!
Success rate is 100 percent (7/7), round-trip min/avg/max = 288/290/293 ms
```

Notice the “extended commands” section. If I selected “yes” here then I would be able to tell the command from which interface the icmp packets will originate. You will find this very helpful later on when mixing routing protocols on a router. Once you find out if a destination is unreachable you can use the trace route command to “pin-point” where the problem may be. You should see:

```plaintext
Randy#traceroute 192.168.4.1
Type escape sequence to abort.
Tracing the route to 192.168.4.1
  1 192.168.30.2 16 msec 16 msec *
Randy#
```

Possible responses when using traceroute (from most to least likely response):

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Timed out</td>
</tr>
<tr>
<td>U</td>
<td>Port was unreachable</td>
</tr>
<tr>
<td>N</td>
<td>Network was unreachable</td>
</tr>
<tr>
<td>P</td>
<td>Protocol is unreachable</td>
</tr>
<tr>
<td>!H</td>
<td>Received but not forwarded…ACL is set</td>
</tr>
</tbody>
</table>
Another layer 3 tool you can use to look for clues is the **sh ip route** command. Here you can determine if your router is advertising and receiving routes and if they are correctly being advertised and received. Here is an example routing table from Randy in our example:

```
Randy>sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
    D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
    U - per-user static route, o - ODR
Gateway of last resort is not set

C    192.168.30.0/24 is directly connected, Serial0/0
R    192.168.4.0/24 [120/1] via 192.168.30.2, 00:00:07, Serial0/0
C    192.168.3.0/24 is directly connected, Ethernet0/0
Randy>
```

If you feel you have your routing protocol correct but the routes are not showing up in the ip routing table then clear them several times:

```
Randy#clear ip route *
Randy#clear ip route *
Randy#clear ip route *
Randy#
```

Then try to see if your routes are correct. You may even have to clear them on all your routers. As a last resort take out your routing protocol and put it back in. Don’t ask me why…sometimes that is all that is needed. One last layer 3 tool…you can also turn your router into a mini-layer 3 protocol inspector with “debug” commands. Be careful when using these because they save all their information in RAM/DRAM. Too much information can *choke* out the performance of your router so only use debug commands sparingly. To view ping packets (aka ICMP packets) use the **debug ip icmp** command. You should see something like this.

```
Randy#debug ip icmp
ICMP packet debugging is on
Randy#
01:02:29: ICMP: time exceeded (time to live) sent to 192.168.3.2 (dest was 192.168.4.2)
01:02:29: ICMP: time exceeded (time to live) sent to 192.168.3.2 (dest was 192.168.4.2)
01:02:29: ICMP: time exceeded (time to live) sent to 192.168.3.2 (dest was 192.168.4.2)
01:02:29: ICMP: dst (192.168.3.1) port unreachable sent to 192.168.3.2
```
**Don’t forget to use undebug all or undebug ip icmp when you are finished.**

4. Finally Telnet (terminal emulation), an application layer program, tests the functionality of all 7 layers. If you can telnet from one router to another, then everything should be working fine and you won’t need anything from this lab. Here is an example of using telnet from Randy to Ward. You should see:

```
Randy# telnet 192.168.30.2
Trying 192.168.30.2 ... Open

User Access Verification

Password:
Ward>
```

One problem with telnet: if a vty password is not “set” on the other router you will not be able to access the router, even though everything is working fine. Let’s look at what you will see if you do not have a vty password set:

```
Randy# telnet 192.168.30.2
Trying 192.168.30.2 ... Open
Password required, but none set
[Connection to 192.168.30.2 closed by foreign host]
```

5. Finally, do not forget about those workstations out there! Just because you can telnet router to router does not mean all is well…be sure you can ping from workstation A to workstation B.

**Supplemental Lab or Challenge Activity:**

1. In this lab if you were consoled into Ward from workstation B, then what would you expect to see if you typed this command `ping 192.168.3.1`? Assume everything is cabled, programmed, and working correctly.

2. If you typed this command `ping 192.168.3.1` from Ward and received five “U’s” then what would you test? (give several steps)

3. If you typed this DOS command `tracert 192.168.3.1` from workstation B and received a timeout message after the serial interface on Randy then what would you test? (give several steps)

4. If you were having problems with a serial line (DCE) and typed `sh int` on Randy and found out the interface was “UP-DOWN” then what would you test? (give several steps)
So What Have I Learned Here?
In this lab you learned the basics of troubleshooting from one router to the other. As you move up in your studies you will learn more precise troubleshooting methods. I cannot tell you how many times a student told me their network didn’t work and all that was wrong was an unplugged cable. Keep it simple first. Let me introduce you to Murphy’s Law of Computers: It works better when it is plugged in. How true, how true.

Guest Router Name Derivation
Ward Christensen and Randy Suess are generally attributed as creating the first Bulletin Board System (BBS) in 1978. The BBS site, located in Chicago, Illinois is still supposed to be in operation today.


Pro-Cote Inc.
Specialty Coatings Applicators
877-588-4300

Certified Conductive and Static Dissipative Floor applications for every need:
- Facilities with highly sensitive electronic equipment
- Computer facilities
- Electronic manufacturing & assembly
- Facilities containing AGV lines
- Clean rooms
- Manufacturing/processing flammable or combustible materials
- Munitions plants

Pro-Cote Inc. is a proud installer of Valspar Flooring Systems
www.pro-cote-usa.com

Pro-Cote Inc.
P. O. Box 1766
Largo, FL  33779
Toll Free (877) 588-4300
BASIC TROUBLESHOOTING—RIP

look for lights on the interfaces

on        off

check cabling not plugged plug it in

check IP addresses & masks\(^1\), encapsulation, clockrates (sh int and sh run)

right

Try typing “no shut” on the interface worked should be fixed

didn’t work

Check routing protocol, autonomous number, to see if the networks are advertised, or for correct routing info. (sh run, sh ip route clear ip route *) worked should be fixed
didn’t work

Try to telnet worked should be fixed
didn’t work

Time to get help!

\(^1\) I know IP addressing is a layer 3 function, but it uses a layer 2 command sh int to view its status.
Loopback Interfaces

Objectives:
To learn how and when to use loopback interfaces.

Tools and Materials:
(1) router
(1) switch
(2) Straight-through cables
(1) rollover cable
(1) workstation with Hyperterminal program

Background:
Loopback interfaces are used for a variety of situations: for OSPF selection, troubleshooting, and, for us, allowing us to test multiple connections without having to actually have a network set up.

Lab Diagram:

Workstation “A”

Routers
Hostnames      Bell
E0             192.168.3.1/24
S0             192.168.4.1/24
S1             n/a

Workstations
A
IP             192.168.3.2
SM             255.255.255.0
GW             192.168.3.1

Step-by-Step Instructions:
1. Set up the lab as shown. Since there is no cable “physically” connected to serial 0 we should not be able to ping it. We can verify that no cable is present by using
the `sh controller s0` command. This command is especially helpful when doing remote access to routers. With the `show controllers` command you should see:

```
Bell#sh controller e0/0
Interface Ethernet0/0
Hardware is AMD Presidio2
ADDR: 80F3A068, FASTSEND: 800255BC, MCI_INDEX: 0
DIST ROUTE ENABLED: 0
Route Cache Flag: 1
LADR=0x0020 0x0100 0x0000 0x0000
CSR0  =0x00000072, CSR3  =0x00001044, CSR4  =0x0000491D,
CSR15 =0x00000000
CSR80 =0x0000D900, CSR114=0x00000001, CRDA  =0x01D175C0,
CXDA =0x01D17AA0
HW filtering information:
  Promiscuous Mode Disabled, PHY Addr Enabled, Broadcast
Addr Enabled
  PHY Addr=0002.FD45.AE60, Multicast Filter=0x0200 0x0100
0x0000 0x0000
  amdp2_instance=0x80F3B948, registers=0x40000000,
  ib=0x1D17460
  rx ring entries=32, tx ring entries=64
  rxring=0x1D174C0, rxr shadow=0x80F3BB20, rx_head=16,
  rx_tail=0
  txring=0x1D17700, txr shadow=0x80F3BBCC, tx_head=50,
  tx_tail=50, tx_count=0
Software MAC address filter(hash:length/addr/mask/hits):
  0x57:  0 100.5e00.0009 0000.0000.0000         0
  0xC0:  0 100.0ccc.cccc 0000.0000.0000         0
  spurious_idon=0, throttled=0, enabled=0, disabled=0
  rx framing_err=0, rx overflow_err=0, rx buffer_err=0
  rx_bpe_err=0, rx_soft_overflow_err=0, rx_no_enp=0,
  rx_discard=0
  tx_one_col_err=0, tx_more_col_err=0, tx_no_enp=0,
  tx_deferred_err=0
  tx_underrun_err=0, tx late_collision_err=0,
  tx_loss_carrier_err=70
  tx_exc_collision_err=0, tx_buff_err=0, fatal_tx_err=0
  hsrp_conf=0, need_af_check=0
Bell#sh controller s0/0
Interface Serial0/0
Hardware is PowerQUICC MPC860
No serial cable attached
idb at 0x80F4204C, driver data structure at 0x80F47560
SCC Registers:
  General [GSMR]=0x2:0x00000000, Protocol-specific [PSMR]=0x8
...
  0 transmitter CTS losts
  0 aborted short frames
```

2. We can use a loopback interface as a “logical” interface. We can use whatever address we want with loopback addresses (i.e., 1.1.1.1, 240.21.2.2, etc). So let’s configure a loopback interface:
Bell(config)#int loop 0
Bell(config-if)#ip address 1.1.1.1 255.255.255.0
Bell(config-if)#no shut

We really do not have to add the “no shut” since these are logical interfaces, but it's good practice to always “no shut” anytime you are configuring an interface.
Exit from the configuration mode and ping from the router to the loopback interface. You should see:

Bell#ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Bell#

Ping from the workstation to the loopback interface. You should see:

C:\WINDOWS\Desktop>ping 1.1.1.1

Pinging 1.1.1.1 with 32 bytes of data:

Reply from 1.1.1.1: bytes=32 time=2ms TTL=255
Reply from 1.1.1.1: bytes=32 time=1ms TTL=255
Reply from 1.1.1.1: bytes=32 time=1ms TTL=255
Reply from 1.1.1.1: bytes=32 time=1ms TTL=255

Ping statistics for 1.1.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms

C:\WINDOWS\Desktop>
3. We can do more…add these in and try to ping (from DOS) to each loopback interface from your workstation:

<table>
<thead>
<tr>
<th>Loopback</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11.11.11.11/24</td>
</tr>
<tr>
<td>2</td>
<td>22.22.22.22/24</td>
</tr>
<tr>
<td>3</td>
<td>33.33.33.33/24</td>
</tr>
<tr>
<td>4</td>
<td>44.44.44.44/24</td>
</tr>
</tbody>
</table>

Then try to ping them. Here is what you will see from your router:

Bell#**ping 11.11.11.11**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 11.11.11.11, timeout is 2 seconds:
!!!!!
**Success rate is 100 percent (5/5)**, round-trip min/avg/max = 1/2/4 ms

Bell#**ping 22.22.22.22**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 22.22.22.22, timeout is 2 seconds:
!!!!!
**Success rate is 100 percent (5/5)**, round-trip min/avg/max = 1/2/4 ms

Bell#**ping 33.33.33.33**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 33.33.33.33, timeout is 2 seconds:
!!!!!
**Success rate is 100 percent (5/5)**, round-trip min/avg/max = 1/1/4 ms

Bell#**ping 44.44.44.44**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 44.44.44.44, timeout is 2 seconds:
!!!!!
**Success rate is 100 percent (5/5)**, round-trip min/avg/max = 1/1/4 ms

Bell#
Here is what you will see from your workstation:

C:\WINDOWS\Desktop>ping 11.11.11.11
Pinging 11.11.11.11 with 32 bytes of data:
Reply from 11.11.11.11: bytes=32 time=3ms TTL=255
Reply from 11.11.11.11: bytes=32 time=1ms TTL=255
Reply from 11.11.11.11: bytes=32 time=1ms TTL=255
Reply from 11.11.11.11: bytes=32 time=1ms TTL=255
Ping statistics for 11.11.11.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 3ms, Average = 1ms
C:\\WINDOWS\\Desktop>

C:\WINDOWS\Desktop>ping 22.22.22.22
Pinging 22.22.22.22 with 32 bytes of data:
Reply from 22.22.22.22: bytes=32 time=1ms TTL=255
Reply from 22.22.22.22: bytes=32 time=1ms TTL=255
Reply from 22.22.22.22: bytes=32 time=1ms TTL=255
Reply from 22.22.22.22: bytes=32 time=1ms TTL=255
Ping statistics for 22.22.22.22:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
C:\WINDOWS\Desktop>

C:\WINDOWS\Desktop>ping 33.33.33.33
Pinging 33.33.33.33 with 32 bytes of data:
Reply from 33.33.33.33: bytes=32 time=1ms TTL=255
Reply from 33.33.33.33: bytes=32 time=1ms TTL=255
Reply from 33.33.33.33: bytes=32 time=1ms TTL=255
Reply from 33.33.33.33: bytes=32 time=1ms TTL=255
Ping statistics for 33.33.33.33:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
C:\WINDOWS\Desktop>

C:\WINDOWS\Desktop>ping 44.44.44.44
Pinging 44.44.44.44 with 32 bytes of data:
Reply from 44.44.44.44: bytes=32 time=1ms TTL=255
Reply from 44.44.44.44: bytes=32 time=1ms TTL=255
Reply from 44.44.44.44: bytes=32 time=1ms TTL=255
Reply from 44.44.44.44: bytes=32 time=1ms TTL=255
Ping statistics for 44.44.44.44:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
C:\WINDOWS\Desktop>
Let’s do a bit of fudging with our Ethernet interface. We can turn it into, more or less, a pseudo-loopback interface with the addition of a simple command in our interface configuration mode. First I will try to ping the Ethernet interface and see it will not work using the extended ping coming from the serial interface. Then we will fudge the Ethernet interface and try pinging it again.

C:\WINDOWS\Desktop>ping 192.168.3.1
Pinging 192.168.3.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.168.3.1:
    Packets: Sent = 4, Received = 0, Lost = 0 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 3ms, Average = 1ms
C:\WINDOWS\Desktop>

Bell#config t
Bell(config)#interface e0/0
Bell(config-if)#no keepalive

C:\WINDOWS\Desktop>ping 192.168.3.1
Pinging 192.168.3.1 with 32 bytes of data:
Reply from 192.168.3.1: bytes=32 time=1ms TTL=255
Reply from 192.168.3.1: bytes=32 time=1ms TTL=255
Reply from 192.168.3.1: bytes=32 time=1ms TTL=255
Reply from 192.168.3.1: bytes=32 time=1ms TTL=255
Ping statistics for 192.168.3.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 3ms, Average = 1ms
C:\WINDOWS\Desktop>

**Supplemental Lab or Challenge Activity:**
1. When do you think you might use loopback interfaces?
2. How many loopback interfaces can you have on a router?
3. Why do you think you can use any IP address? Can you use 0.0.0.0? What is the upper and lower limit to loopback addresses?

**So What Have I Learned Here?**
In this lab you learned how to configure a loopback adapter. These are actually pretty cool. If we configured an Ethernet interface then we would have to have a cable and a switch or something to be able to ping it. A loopback is a virtual interface so no cable is needed. Later on, when you get up in your studies you will learn many more uses for loopbacks. Let’s add a protocol inspector to our RIP network and look at the packets!

**Guest Router Name Derivation**

In 1876 Alexander Graham Bell invented the telephone. Would you believe the first “hackers” came along a couple of years after that in 1878? Those first “phreakers” played pranks by switching calls to places they were not suppose to go, disconnecting some calls, and other pranks. Yup…it’s been around for a while now.
Earn your Bachelor’s Degree from St. Petersburg College!

Transferring and having access to a 4-year degree has never been easier, more convenient, or cost effective.

A bachelor’s program at St. Petersburg College can get you not only a degree, but a rewarding career in Education, Health Care, Technology Management or Public Safety Administration.

**St. Petersburg College** allows students who have received an Associate degree (Associate in Arts or Associate in Science, depending on the program) to transfer into the following Baccalaureate degrees programs:

- Bachelor of Applied Science in Dental Hygiene
- Bachelor of Applied Science in Nursing
- Bachelor of Applied Science in Technology Management
- Bachelor of Applied Science Orthotics and Prosthetics
- Bachelor of Applied Science in Public Safety Administration
- Bachelor of Applied Science in Veterinary Technology
- Bachelor of Science in Education
  - Elementary Education with infused ESOL (K-6)
  - Exceptional Student Education with infused ESOL (K-12)
  - Secondary Mathematics Education (6-12)
  - Business Technologies Education (6-12)
  - Secondary Science Education (6-12)
  - Technology Education (6-12)
  - Industrial Technical Education (Certificate only)

Opportunities Include:

- **On-line** and in traditional classroom courses
- Day and evening classes
- Full and part-time course schedules

Explore the opportunities waiting for you at St. Petersburg College!

For more information call: (727) 341-4772
E-mail: 4yeardegrees@spcollege.edu
Basic RIP with Protocol Inspector

Objective:
To use a protocol inspector to view network traffic on a RIP version 1 network.

Tools and Materials:
(2) PC/workstations with protocol inspectors (Ethereal used here)
(2) Routers
(2) Switches
(4) Straight-through cables
(1) DCE serial cable
(1) DTE serial cable
(2) rollover cables

Background:
You can get a free protocol inspector at http://www.ethereal.com or packetizer. See the labs in part 1 for downloading instructions if you have not done so already.

Lab Diagram:

Addressing:

<table>
<thead>
<tr>
<th>Routers</th>
<th>Hostnames</th>
<th>IP Address</th>
<th>Subnet Mask</th>
<th>Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>E0</td>
<td>Emmanuel</td>
<td>10.1.3.1/24</td>
<td>255.255.255.0</td>
<td>10.1.3.1</td>
</tr>
<tr>
<td>S0</td>
<td>Goldstein</td>
<td>10.1.4.1/24</td>
<td>255.255.255.0</td>
<td>10.1.4.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workstations</th>
<th>IP Address</th>
<th>Subnet Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10.1.3.2</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>B</td>
<td>10.1.4.2</td>
<td>255.255.255.0</td>
</tr>
</tbody>
</table>
Step-by-Step Instructions:
1. Read all of these instructions carefully…you will be recording times and such so it is important that you are familiar with these steps BEFORE you do them.
2. Cable and set up the lab as shown. Test for complete connectivity by sending icmp packets from one workstation to another. Start `debug ip rip` on a router. We will use this for a benchmark. Remember RIP updates by default every 30 seconds therefore about every 30 seconds or so we would expect to see messages on the hyperterminal session with RIP update information similar to this:

```
emmanual# debug ip rip
RIP protocol debugging is on
emmanual#
01:17:00: RIP: sending v1 update to 255.255.255.255 via Ethernet0/0 (10.1.3.1)
01:17:00: RIP: build update entries
01:17:00: subnet 10.1.4.0 metric 2
01:17:00: subnet 10.1.192.0 metric 1
01:17:00: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (10.1.192.1)
01:17:00: RIP: build update entries
01:17:00: subnet 10.1.3.0 metric 1
emmanual#
01:17:04: RIP: received v1 update from 10.1.192.2 on Serial0/0
01:17:04: 10.1.4.0 in 1 hops
emmanual#
01:17:29: RIP: sending v1 update to 255.255.255.255 via Ethernet0/0 (10.1.3.1)
01:17:29: RIP: build update entries
01:17:29: subnet 10.1.4.0 metric 2
01:17:29: subnet 10.1.192.0 metric 1
01:17:29: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (10.1.192.1)
01:17:29: RIP: build update entries
01:17:29: subnet 10.1.3.0 metric 1
emmanual#
```

Keep track of updates and changes at [http://www.spcollege.edu/star/cisco](http://www.spcollege.edu/star/cisco) Scroll to the bottom of the page and click on the “Lab Manual Edits.”
3. Enable the protocol inspector to begin “capture” of network packets from workstation A by control+k. Change the PPP interface to your NIC. Press “start.” Check your watch and record the time. See figure 1 below.

![Figure 1—Be sure to select the NIC as your capture interface.](image1)

4. From workstation B ping workstation A and then trace the route between the two. We are essentially generating ICMP packets on our network. From our knowledge of CISCO routers we can expect to see RIP updates every 30 seconds ("other"), CDP every 60 seconds ("other"), and our ICMP packets as they are sent and received. See figure 2 below.

![Figure 2—A capture in progress.](image2)

5. Wait two minutes-ish. Remember, you won’t see them until you stop the capture and analyze what has happened.
6. End the capture on the protocol inspector by pressing “stop.”
7. Open the capture file.
8. From “workstation A” you should see something like these pictures (results will vary somewhat):

Figure 3—Here we can see what we were expecting...RIP and ICMP. And those STP packets? You will learn about them in part 3.

Figure 4—We can even look down to the HEX information (bit-by-bit) at our RIP update captures. Notice how our RIP Operation is a “broadcast” (FF FF FF FF FF) on the network.
Here we can actually “see” our rip packets and the two routes it is carrying. Notice our decimal ip address highlighted is 10.1.4.0 and in the bottom frame we see the hexadecimal equivalent 0a-01-04-00. Well? In hex 0a is 10 and the rest should be pretty apparent. Seeing is believing!

Take college courses on a new schedule--yours.

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education. Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

http://e.spcollege.edu

St. Petersburg College’s eCampus. Your time. Your place. Your future.
Supplemental Lab or Challenge Activity:

1. Try this lab again using nothing but specific debug commands. The first time through use specific debug commands. The second time through use the "debug all" command and see how it differs.

2. Try to find out what “promiscuous mode” means as it applies to NIC’s. Why do you think this would be important as it relates to sniffers?

3. If we didn’t want to run CDP on our network, then how do we disable it?

So What Did We Learn Here?
Tools of the trade. Get used to using protocol inspectors to examine your network health. They are really cool. If you want to do more with security they are essential.

Guest Router Name Derivation

Emmanuel Goldstein founded 2600 magazine (a.k.a. The Hacker Quarterly) back in 1984. Every four months a magazine devoted entirely to the sharing of information related to the Internet is published. In many stores you have to ask for it by name because they keep it under the counter or back in the porn section. It has been suggested that purchasing 2600 with a check or credit card or subscribing to the magazine immediately signals the FBI to start a file on you as a “potential hacker.” Want to find out if you have a FBI file started on you? Netmatix0.virtualave.net/FBIFILES.TXT.txt

Also, in the movie “Hackers” one of the names of a main character was “Emmanuel Goldstein.” Coincidence? I don’t think so.
Router Telnet Lab

Objective:
To learn the intricacies of using “telnet” and commands related to telnet to move between CISCO routers.

Tools and Materials:
(2) PC/workstations  
(2) Routers  
(2) Switches  
(4) Straight-through cables  
(1) DCE serial cable  
(1) DTE serial cable  
(2) rollover cables

Background:
Using telnet between routers is similar to using telnet from a DOS or Windows session, except that with routers we have certain keystrokes to suspend, resume disconnect, and end a telnet session. We also have certain show and debug tools that we can use as a “mini” protocol inspector to view telnet features.

Lab Diagram:

Addressing:

<table>
<thead>
<tr>
<th>Routers</th>
<th>Hostnames</th>
<th>IP Address / Subnet Mask</th>
<th>Gateway Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>E0</td>
<td>William</td>
<td>180.11.3.1/24</td>
<td>180.11.3.1</td>
</tr>
<tr>
<td>S0</td>
<td>180.11.12.1/24 (DTE)</td>
<td>n/a</td>
<td>180.11.12.2/24 (DCE)</td>
</tr>
<tr>
<td>S1</td>
<td>n/a</td>
<td>180.11.12.2/24 (DCE)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workstations</th>
<th>Workstation “A”</th>
<th>Workstation “B”</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>180.11.3.2</td>
<td>180.11.3.1</td>
</tr>
<tr>
<td>SM</td>
<td>255.255.255.0</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>GW</td>
<td>180.11.3.1</td>
<td>180.11.4.1</td>
</tr>
</tbody>
</table>
Step-by-Step Instructions:
1. Cable and setup the lab as shown. Test for complete connectivity by sending icmp packets from one workstation to another.
2. From the William router initiate a telnet session into the Gibson router. If vty line passwords were not set on Gibson then you will not be able to telnet into it. If you are successful then you should see something like this:

   william#telnet 180.11.12.2
   Trying 180.11.12.2... Open
   User Access Verification
   Password:
   gibson>

3. Next we will suspend our session between Cleveland and Detroit by using these keys together: Control+Shift+6 and then X (sounds like a good CCNA question). You should see something like this:

   gibson>
   william#

4. To resume the session just hit <enter> or <return> twice. You should see something like this:

   [Resuming connection 1 to 180.11.12.2 ... ]
   gibson>

5. To end a session just type exit. You should see something like this:

   gibson>exit
   [Connection to 180.11.12.2 closed by foreign host]
   william#

6. To see all current sessions type sh sessions. You should see something like this (I started a telnet session so I would have something to show):

   gibson>
   william#sh sessions
   Conn Host Address Byte Idle Conn Name
   * 1 180.11.12.2 180.11.12.2 0 0 180.11.12.2
   william#

7. To view telnet information use the debug telnet command. Enable debug telnet on one Cleveland, then initiate a telnet session Cleveland from Detroit and watch the debug. You should see something like this:

   william#debug telnet
   Incoming Telnet debugging is on
   00:36:59: Telnet66: 1 1 251 1
   00:36:59: TCP66: Telnet sent WILL ECHO (1)
   00:36:59: Telnet66: 2 2 251 3
Supplemental Lab or Challenge Activity:
1. Try using telnet from a workstation into a router.
2. Use the debug telnet with a workstation accessing a router with telnet.
3. Try these commands again later when you have multiple routers set up in a network.
4. Why would you not want to allow anyone to access your router with telnet? Why would you?
5. Look at that telnet debug and figure out what is happening there.
6. Put a different password on each of the vty sessions. Leave one open. Then open multiple sessions into that router. You can use the “terminal monitor” command to see syslog messages as they are generated on that remote telnet session. Be sure to try using show telnet, debug telnet, show users, show sessions, and show debugging.

So What Have I Learned Here?
In Part 1 you learned how to use telnet to get to places on the web. Well…routers are on the web too. So in this lab you learned about using telnet to your routers. In Part 3 you will learn how to telnet into switches. You may use this someday to telnet into the routers at work from your home…just think…no commute. Ahhh wouldn’t it be nice?

Guest Router Name Derivation

In 1982 William Gibson first coined the term “cyber-space” in his novel.
Route Summarization with RIP

Objectives:
To further your understanding of the RIP routing protocol as it applies to subnetting design with classful addresses. You will also view updates sent and received with RIP.

Tools and Materials:
(2) PC/workstations with protocol inspectors
(2) Routers
(2) Switches
(4) Straight-through cables
(1) DCE serial cable
(1) DTE serial cable
(2) rollover cables

Background:
By default, when you enable RIP on a CISCO router you are enabling RIP version 1. There are two versions of RIP which, oddly enough, are called RIP version 1 (a.k.a. RIP) and RIP version 2 (a.k.a. RIPv2). RIP (version 1) is categorized as a “classful” routing protocol. When you enable RIP or RIPv2 the routers pass updates every 30 seconds by default. RIP version 1 does not pass any subnet mask information with its updates. It just “truncates” (cuts-off) any information at the classful boundary (where the network portion stops and the subnet portion starts). In CISCO-speak: RIP uses “auto-summary” by default which cannot be disabled. For example, a class “B” address of 143.46.86.128 with RIP version 1 would be truncated to 143.46.0.0 during its updates. Remember, class B is network-network-host-host. RIP version 2 does pass the subnet information with its updates, but you will learn more about RIPv2 in another lab. Confused? Yeah, me too. Let’s “learn by doing” using a class “B” address in this lab.

Lab Design:
Addressing:

**Routers**

<table>
<thead>
<tr>
<th>Hostnames</th>
<th>Phiber</th>
<th>Optik</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>161.20.4.1/30 (DCE)</td>
<td>n/a</td>
</tr>
<tr>
<td>S1</td>
<td>n/a</td>
<td>161.20.4.2/30 (DTE)</td>
</tr>
<tr>
<td>L0</td>
<td>161.20.3.1/30</td>
<td>161.20.5.1/30</td>
</tr>
<tr>
<td>E0</td>
<td>161.20.2.1/24</td>
<td>161.20.1.1/24</td>
</tr>
</tbody>
</table>

**Workstations**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>161.20.2.2</td>
<td>161.20.1.2</td>
</tr>
<tr>
<td>SM</td>
<td>255.255.255.0</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>GW</td>
<td>161.20.2.1</td>
<td>161.20.1.1</td>
</tr>
</tbody>
</table>

**Step-by-Step Instructions:**

1. Cable and set up the lab as shown. Test for connectivity from workstation A to its gateway and workstation B to its gateway. Test ping from workstation A to workstation B. This should NOT work. Test ping from workstation A to Loopback 0. Test ping from workstation B to Loopback 0. These should work (virtual ports). Test ping from workstation B to its gateway and to workstation A. This one also should NOT work because of route summarization with RIP version 1. In short…route summarization “chopped” the network off at 161.20.0.0/16 and did not advertise the /24 routes. Follow the logic flow charts at the end of the lab.

2. Let’s look a little deeper at what is happening. Since this is a routing issue let’s issue the `sh ip route` command on Phiber. You should see that the loopback is directly connected on Phiber:

```
phiber#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
        U - per-user static route, o - ODR

Gateway of last resort is not set

161.20.0.0/16 is variably subnetted, 4 subnets, 2 masks
 R 161.20.5.0/30 [120/1] via 161.20.4.2, 00:00:02, Serial10/0
 C 161.20.4.0/30 is directly connected, Serial10/0
 C 161.20.3.0/30 is directly connected, Loopback0
 C 161.20.2.0/24 is directly connected, Ethernet0/0
```

We can see routes 161.20.2.0, 161.20.3.0, and 161.20.4.0 are directly connected with 161.20.5.0 being learned over the serial line but the 161.20.1.0 network is not listed because it was summarized. Likewise we similar things on Optik:
optik#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
       U - per-user static route, o - ODR
Gateway of last resort is not set

161.20.0.0/16 is variably subnetted, 4 subnets, 2 masks
C  161.20.5.0/30 is directly connected, Loopback0
C  161.20.4.0/30 is directly connected, Serial0/1
C  161.20.1.0/24 is directly connected, Ethernet0/0
R  161.20.3.0/30 [120/1] via 161.20.4.1, 00:00:06, Serial0/1

3. Let’s also see what is happening with RIP. Turn on `debug ip rip` on both routers to view the updates sent and received. You should see on Phiber:

    phiber#debug ip rip
    RIP protocol debugging is on
    phiber#
    01:26:15: RIP: received v1 update from 161.20.4.2 on Serial0/0
    01:26:15: 161.20.5.0 in 1 hops
    01:26:19: RIP:sending v1 update to 255.255.255.255 via Ethernet0/0 (161.20.2.1)
           - suppressing null update
    01:26:19: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (161.20.4.1)
    01:26:19: subnet 161.20.3.0, metric 1
    01:26:19: RIP: sending v1 update to 255.255.255.255 via Loopback0 (161.20.3.1)
    01:26:19: subnet 161.20.5.0, metric 2
    01:26:19: subnet 161.20.4.0, metric 1
    01:26:41: RIP: received v1 update from 161.20.4.2 on Serial0/0
    01:26:41: 161.20.5.0 in 1 hops
    01:26:49: RIP: sending v1 update to 255.255.255.255 via Ethernet0/0 (161.20.2.1) - suppressing null update
    01:26:49: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (161.20.4.1)
    01:26:49: subnet 161.20.3.0, metric 1
    01:26:49: RIP: sending v1 update to 255.255.255.255 via Loopback0 (161.20.3.1)
    01:26:49: subnet 161.20.5.0, metric 2
    phiber#undebug ip rip
    RIP protocol debugging is off
On Optik you should see:

```
optik#debug ip rip
RIP protocol debugging is on
01:26:35: RIP: sending v1 update to 255.255.255.255 via Ethernet0/0 (161.20.1.1) - suppressing null update
01:26:35: RIP: sending v1 update to 255.255.255.255 via Serial0/1 (161.20.4.2)
01:26:35:      subnet 161.20.5.0, metric 1
01:26:35: RIP: sending v1 update to 255.255.255.255 via Loopback0 (161.20.5.1)
01:26:35:      subnet 161.20.4.0, metric 1
01:26:35:      subnet 161.20.3.0, metric 2
01:26:42: RIP: received v1 update from 161.20.4.1 on Serial0/1
01:26:42: 161.20.3.0 in 1 hops
optik#undebug ip rip
RIP protocol debugging is off
optik#
```

Compare this to these logic flow charts:

How RIP *sends* updates (for example from Phiber to Optik):

<table>
<thead>
<tr>
<th>Are you part of my major network?</th>
<th>yes</th>
<th>Do you have the same subnet mask?</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td></td>
<td>Router “Phiber” summarizes at the major net boundary and advertises the network.</td>
<td></td>
</tr>
<tr>
<td>yes</td>
<td></td>
<td>Router “Phiber” drops the network and does not advertise it.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router “Phiber” advertises the Subnet.</td>
<td></td>
</tr>
</tbody>
</table>

How RIP *receives* updates (for example on Phiber to Optik):

<table>
<thead>
<tr>
<th>Is the subnet received on the same major net as the intf. that received the update?</th>
<th>no</th>
<th>Do any subnets of this major network already exist in the routing table?</th>
<th>yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td></td>
<td>Router “Optik” applies the mask of the interface that received the update.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ignore the update</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router “Optik” applies a classful mask.</td>
<td></td>
</tr>
</tbody>
</table>
Supplemental Lab or Challenge Activity:
1. You are the network administrator for a small real estate company in Tulsa, Oklahoma. You have to set up a network with two CISCO 2611 routers, 4 1924 switches, and 18 workstations and 4 printers per subnet. For security purposes you have decided that you do not want to advertise subnet information for one subnet on each router. Therefore you have decided to use discontiguous subnets so your routers will summarize routes. You will need to design and set up 4 subnets in the company. When you are finished designing it you will need to build it and be able to ping from each workstation to each other workstation where possible.

2. A good command to remember is to use *clear ip route* . Sometimes you want to force your IP table to update and change and this is one good way to make that happen.

3. Let’s look at some designs and have you determine whether all workstations could ping all other workstations before implementing it. In other words do you think that given the IP addressing design that the routers will summarize the networks or not?

Scenario 1:

Routers

<table>
<thead>
<tr>
<th>Hostnames</th>
<th>Phiber</th>
<th>Optik</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>10.2.4.1/30 (DCE)</td>
<td>n/a</td>
</tr>
<tr>
<td>S1</td>
<td>n/a</td>
<td>10.2.4.2/30 (DTE)</td>
</tr>
<tr>
<td>E0</td>
<td>192.168.1.1/24</td>
<td>192.168.5.1/24</td>
</tr>
<tr>
<td>E1</td>
<td>192.168.2.1/24</td>
<td>192.168.4.1/24</td>
</tr>
</tbody>
</table>

Workstations

<table>
<thead>
<tr>
<th>A-E0</th>
<th>192.168.1.2</th>
<th>192.168.5.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-E0</td>
<td>255.255.255.0</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>A-E1</td>
<td>192.168.1.1</td>
<td>192.168.5.1</td>
</tr>
<tr>
<td>B-E1</td>
<td>255.255.255.0</td>
<td>255.255.255.0</td>
</tr>
</tbody>
</table>

Scenario 2:

Routers

<table>
<thead>
<tr>
<th>Hostnames</th>
<th>Phiber</th>
<th>Optik</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>10.2.4.1/24 (DCE)</td>
<td>n/a</td>
</tr>
<tr>
<td>S1</td>
<td>n/a</td>
<td>10.2.4.2/24 (DTE)</td>
</tr>
<tr>
<td>E0</td>
<td>192.168.1.1/24</td>
<td>192.168.5.1/24</td>
</tr>
<tr>
<td>E1</td>
<td>192.168.2.1/24</td>
<td>192.168.4.1/24</td>
</tr>
</tbody>
</table>

Workstations

<table>
<thead>
<tr>
<th>A-E0</th>
<th>192.168.1.2</th>
<th>192.168.5.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-E0</td>
<td>255.255.255.0</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>A-E1</td>
<td>192.168.1.1</td>
<td>192.168.5.1</td>
</tr>
<tr>
<td>B-E1</td>
<td>255.255.255.0</td>
<td>255.255.255.0</td>
</tr>
</tbody>
</table>
Scenario 3:

<table>
<thead>
<tr>
<th>Hostnames</th>
<th>Phiber</th>
<th>Optik</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>1.0.0.1/8 (DCE)</td>
<td>n/a</td>
</tr>
<tr>
<td>S1</td>
<td>n/a</td>
<td>1.0.0.2/8 (DTE)</td>
</tr>
<tr>
<td>E0</td>
<td>2.0.0.1/8</td>
<td>4.0.0.1/8</td>
</tr>
<tr>
<td>E1</td>
<td>3.0.0.1/8</td>
<td>5.0.0.1/8</td>
</tr>
<tr>
<td>Workstations</td>
<td>A-E0</td>
<td>B-E0</td>
</tr>
<tr>
<td>IP</td>
<td>2.0.0.2/8</td>
<td>4.0.0.2/8</td>
</tr>
<tr>
<td>SM</td>
<td>255.0.0.0</td>
<td>255.0.0.0</td>
</tr>
<tr>
<td>GW</td>
<td>2.0.0.1/8</td>
<td>4.0.0.1/8</td>
</tr>
<tr>
<td>Workstations</td>
<td>A-E1</td>
<td>B-E1</td>
</tr>
<tr>
<td>IP</td>
<td>3.0.0.1/8</td>
<td>5.0.0.2/8</td>
</tr>
<tr>
<td>SM</td>
<td>255.0.0.0</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>GW</td>
<td>3.0.0.1/8</td>
<td>5.0.0.1/8</td>
</tr>
</tbody>
</table>

Use this for a lab design:

So What Did I Learn Here?

In this lab you learned about a geek-speak term “summarization.” Think back to the subnetting lab…network.network.host.host for a class B address. Between the network and host is the “classful boundary” which is where a router will summarize an address if it uses a protocol like RIP that does not pass subnet mask information. In the next lab we start introducing more routers into our network. It’s about time, right?

Guest Router Name Derivation

Phiber Optik was the leader of the Master’s of Deception (MoD) hackers ring in New York City in the 1980’s/early 1990’s. Allegedly he master-minded the Martin Luther King day crash of AT&T’s national phone service in 1990. Known for his daring actions and media stunts he appeared or was interviewed in many publications including Harper’s, Esquire, and the New York Times. Don’t worry…he got busted. Turk 182!
Intermediate RIP with 3 routers

Objectives:
To learn how to implement networking schemes with more than 2 routers.

Tools and Materials:
(3) PC/workstations
(3) Routers
(3) Switches
(6) Straight-through cables
(2) DCE serial cable
(2) DTE serial cable
(3) rollover cables

Lab Diagram:

Addressing:

Routers
Hostnames  acid  phreak  scorpion
E0  192.168.1.1/24  192.168.2.1/24  192.168.3.1/24
S0  10.1.1.1/24 (DCE)  10.2.1.1/24 (DCE)  n/a
S1  n/a  10.1.1.2/24 (DTE)  10.2.1.2/24 (DTE)

Workstations  a  b  c
IP  192.168.1.2  192.168.2.2  192.168.3.2
SM  255.255.255.0  255.255.255.0  255.255.255.0
GW  192.168.1.1  192.168.2.1  192.168.3.1

Step-by-Step Instructions:
3. Cable the lab as shown.
4. Complete the basic router setup on each router.
5. Configure the interfaces on each router.
6. Configure the routing protocol and advertise the router’s networks.
7. Setup the workstations with IP address, subnet masks, and gateways addresses.
   You will need to reboot the workstations. If they ask for a password for network
connectivity just put anything in and you should see a message something like “no domain server is available, you may not have some networking functions.” It’s ok if you see it, but you probably will not be able to ping outside of your workstation without seeing that error message. A quirk with Microsoft.

8. Test connectivity from router to router (from the router) by using ping from alpha to gamma. You should see:

```
acid# ping 10.2.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms
acid#
```

9. Test connectivity from workstation to workstation (from DOS) by using ping from workstation a to workstation c. You should see:

```
C:\WINDOWS\Desktop> ping 192.168.3.2

Pinging 192.168.3.2 with 32 bytes of data:

Reply from 192.168.3.2: bytes=32 time=21ms TTL=126
Reply from 192.168.3.2: bytes=32 time=20ms TTL=126
Reply from 192.168.3.2: bytes=32 time=21ms TTL=126
Reply from 192.168.3.2: bytes=32 time=21ms TTL=126

Ping statistics for 192.168.3.2:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 20ms, Maximum = 21ms, Average = 20ms

C:\WINDOWS\Desktop>
```

10. Let’s see our route from workstation a to workstation c (from DOS). You should see:

```
C:\WINDOWS\Desktop> tracert 192.168.3.2
Tracing route to STAR10616119 [192.168.3.2] over a maximum of 30 hops:
  1    1 ms    1 ms    1 ms  192.168.1.1
  2   25 ms   25 ms   25 ms  10.1.1.2
  3   30 ms   30 ms   30 ms  10.2.1.2
  4   45 ms   45 ms   45 ms STAR10616119 [192.168.3.2]
Trace complete.
```
Supplemental Lab or Challenge Activity:
5. What would you expect to see if you used these commands?

acid>sh cdp neighbors
acid>sh cdp traffic
acid>sh protocols
acid>sh ip route
acid#debug ip icmp
acid#debug ip rip

6. What would you expect to see on phreak? Try steps 1-6 over again on phreak.
7. Try this with class “B” private IP addresses that you choose.
8. Try this with class “A” private IP address that you choose.
9. Try this lab with one class “A” private IP address for the Ethernet network on acid, a class “B” private IP address over the serial line, and a class “C” private IP address on the Ethernet network on phreak.
10. Try this with class “C” public IP addresses that you choose.
11. Try this with class “B” public IP addresses that you choose.
12. Try this with class “A” public IP address that you choose.
13. Try mixing and matching private and public IP addresses.
14. Try adding a fourth router either before acid or after scorpion. Use it to simulate an ISP with a loopback interface. Obviously you do not want to broadcast your routing tables to the ISP so use a derivative of the “passive interface” command to stop those broadcasts out the serial interface. Oh, know don’t be so snotty…sooner or later you have to learn how to figure out things like this without exact instructions.

So What Have I Learned Here?
After thoroughly drenching ourselves in all things RIP with two routers we decided to tack on another router and bring our total to three. This actually introduces you to routing protocol issues. For example, even though the middle router may be able to ping everywhere in the network, the workstations or other routers may not be able to ping through the middle router, which is evidence of a routing problem on the middle router. This is actually a quite common scenario. The first thing I would do is clear the IP routes out of the tables and check the routes again. So why don’t we have any labs with four or five routers? Simple. If you can do three then four or five is easy. Since most classes are short it is actually a waste of time to set up four or five routers. By the time you get them set up for class it is time to go home. Well now off the soapbox and on to the next lab!

Guest Router Name Derivation

More members of the Master’s of Deception (MoD) hackers ring in New York City in the 1980’s/early 1990’s. They were instrumental in starting the Great Hacker War against the Legion of Doom (LoD) hackers ring (also from NYC). Eventually the LoD were persuaded to cooperate with the police and helped to bust the MoD.
RIP metrics and the Limitations of RIP

Objectives:
To learn how about the limitations of RIP and its metrics.

Background:
In this lab we will explore 3 of the 4 “features” of RIP (version 1). First, we will test the maximum hop count metric (15 is ok, 16 is unreachable). Next we will view the update broadcast of RIP (every 30 seconds) and then change the timer. Finally we look at the timers associated with RIP: route-timeout and flush timer. We will not look at the “feature of RIP” that RIP maintains only the best routes.

Tools and Materials:
(3) PC/workstations with protocol inspectors
(3) Routers
(3) Switches
(6) Straight-through cables
(2) DCE serial cable
(2) DTE serial cable
(3) rollover cables

Lab Diagram:

Addressing:

<table>
<thead>
<tr>
<th>Routers</th>
<th>Hostnames</th>
<th>acid</th>
<th>phreak</th>
<th>scorpion</th>
</tr>
</thead>
<tbody>
<tr>
<td>E0</td>
<td>192.168.1.1/24</td>
<td>192.168.2.1/24</td>
<td>192.168.3.1/24</td>
<td></td>
</tr>
<tr>
<td>S0</td>
<td>10.1.1.1/24 (DCE)</td>
<td>10.2.1.1/24 (DCE)</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>n/a</td>
<td>10.1.1.2/24 (DTE)</td>
<td>10.2.1.2/24 (DTE)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workstations</th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>192.168.1.2</td>
<td>192.168.2.2</td>
<td>192.168.3.2</td>
</tr>
<tr>
<td>SM</td>
<td>255.255.255.0</td>
<td>255.255.255.0</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>GW</td>
<td>192.168.1.1</td>
<td>192.168.2.1</td>
<td>192.168.3.1</td>
</tr>
</tbody>
</table>

Step-by-Step Instructions:
7. Cable the lab as shown.
8. Complete the basic router setup on each router.
9. Configure the interfaces on each router.
10. Configure the routing protocol and advertise/associate/publish the router’s networks.
11. If you have enough routers then test the maximum hop count by adding in enough routers…I would suggest a total of 17. Ping from end to end and from the middle out. Test that 16 hop count limit to the max! Otherwise skip this step.
12. Change the rip timer using these commands. Notice how I show you what the previous (default) RIP timers are before I changed them. The format is updates-invalid timer-hold down timer-flush timer. Then I show you what they were after I changed them:

**Before (30-180-180-240):**

```bash
scorpion#sh ip protocols
Routing Protocol is "rip"
    Sending updates every 30 seconds, next due in 1 seconds
    Invalid after 180 seconds, hold down 180, flushed after 240
    Outgoing update filter list for all interfaces is
    Incoming update filter list for all interfaces is
    Redistributing: rip
    Default version control: send version 1, receive any
    version
    Interface     Send  Recv   Key-chain
    Ethernet0/0   1     1  2
    Serial0/1     1     1  2

Routing for Networks:
    10.0.0.0
    192.168.3.0
```

**During:**

```bash
scorpion#conf t
scorpion(config-router)#timers basic 15 30 60 90
scorpion(config-router)#^Z
```

**After (15-30-60-90):**

```bash
scorpion#sh ip protocols
Routing Protocol is "rip"
    Sending updates every 15 seconds, next due in 20 seconds
    Invalid after 30 seconds, hold down 60, flushed after 90
    Outgoing update filter list for all interfaces is
    Incoming update filter list for all interfaces is
    Redistributing: rip
    Default version control: send version 1, receive any
    version
    Interface     Send  Recv   Key-chain
    Ethernet0/0   1     1  2
    Serial0/1     1     1  2
```

Routing for Networks:

- 10.0.0.0
- 192.168.3.0
Routing Information Sources:

<table>
<thead>
<tr>
<th>Gateway</th>
<th>Distance</th>
<th>Last Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.2.1.1</td>
<td>120</td>
<td>00:00:07</td>
</tr>
</tbody>
</table>

Distance: (default is 120)

scorpion#

**Supplemental Lab or Challenge Activity:**

1. Why would you want to be able to change the timers? Come on now and think hard.
2. Fill in the table below with the *default* timers for RIP to give you some perspective on the RIP metrics. They sound like nice CCNA questions don’t they?

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hop count</td>
<td></td>
</tr>
<tr>
<td>Update timer</td>
<td></td>
</tr>
<tr>
<td>Invalid timer</td>
<td></td>
</tr>
<tr>
<td>Hold-down timer</td>
<td></td>
</tr>
<tr>
<td>Flush timer</td>
<td></td>
</tr>
</tbody>
</table>

**So What Have I Learned Here?**

In this lab you learned about the metrics involved with RIP. Ok. So there aren’t that many, but other protocols will have many metrics so this is a good introduction. Next you will be turning your router into a DHCP server. Gosh that is fun!

**Guest Router Name Derivation**

Acid Phreak and Scorpion are more members of the Master’s of Deception (MoD) hackers ring in New York City in the 1980’s/early 1990’s. They were instrumental in starting the Great Hacker War against the Legion of Doom (LoD) hackers ring (also from NYC). Eventually the LoD were persuaded to cooperate with the police and helped to bust the MoD.

*Keep track of updates and changes at [http://www.spcollege.edu/star/cisco](http://www.spcollege.edu/star/cisco) Scroll to the bottom of the page and click on the “Lab Manual Edits.”*
Dynamic Host Configuration Protocol (DHCP) Lab

Objective:
To learn how to implement DHCP using CISCO routers in networks.

Background:
Although it is preferable to use an actual DHCP server for addressing in a network
CISCO routers can be used to serve that purpose. The command you will need to be
more familiar with is the ip helper address to point your subnets to the DHCP server. As
you see below you must use at least one 2620 router as the DHCP server. This router has
the memory and operating system capable of supporting DHCP. Sorry those 2500’s and
2610/2611’s just won’t work.

Tools and Materials:
(2) PC/workstations
(2) Routers (one must be at least a 2620).
(2) Switches
(4) Straight-through cables
(1) DCE serial cable
(1) DTE serial cable
(2) rollover cables

Lab Diagram:

Addressing:

<table>
<thead>
<tr>
<th></th>
<th>Hostnames</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Routers</td>
<td>kevin</td>
<td>mitnik</td>
<td></td>
</tr>
<tr>
<td>E0</td>
<td>10.0.0.1/8</td>
<td></td>
<td>192.168.3.1/24</td>
</tr>
<tr>
<td>S0</td>
<td>192.168.1.2/24 (DCE)</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>n/a</td>
<td></td>
<td>192.168.1.1/24 (DTE)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workstations</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>IP</td>
<td>10.0.0.2</td>
</tr>
<tr>
<td></td>
<td>SM</td>
<td>255.0.0.0</td>
</tr>
<tr>
<td></td>
<td>GW</td>
<td>10.0.0.1</td>
</tr>
</tbody>
</table>
Step-by-Step Instructions:
8. Cable the lab as shown.
9. Complete the basic router setup on each router.
10. Configure the interfaces on each router.
11. Configure the routing protocol and advertise/associate/publish the router’s networks.
12. Setup the workstations with IP address, subnet masks, and gateways addresses. You will need to reboot the workstations.
13. Test connectivity from router to router.
14. Test connectivity from workstation A to workstation B from DOS.
15. Verify your RIP routes are being advertised.
16. Remove the IP address and gateway from workstation A and set it to obtain its address automatically. You will need to reboot.
17. Program the dhcp router to start dhcp services with the 10.0.0.0/8 network. We will use the name for our dhcp pool as “pool 10-net.” Note how the prompt changes modes below. The last command establishes the default router address.

kevin# config t
kevin(config)# ip dhcp pool 10-net
kevin(config-dhcp)# network 10.0.0.0 255.0.0.0
kevin(config-dhcp)# default-router 10.0.0.1

18. You should be able to release and renew the ip address. You should get an address of 10.0.0.2 on the workstation. (Use Start>run>winipcfg then press the release and renew buttons). Every now and then the ip addressing may seemingly “skip” an IP address. If you have x.x.x.1 on the interface and are expecting x.x.x.2 for the first address and you end up with x.x.x.3 because sometimes the switch may grab one of those numbers…go check your switch and don’t sweat it. Just be sure to plan for it.
19. Test your connectivity between the two workstations.
20. Remove the IP address and gateway from workstation B and set it to obtain its address automatically. You will need to reboot.
21. Set up the class “C” pool on the dhcp router/server. The only difference with this IP pool is we know the interface on mitnik requires the first ip address in the pool so we need to exclude it (try it without the exclude command and you will see the error message).

```
kevin#config t
kevin(config)#ip dhcp pool 192-net
kevin(config-dhcp)#network 192.168.3.0 255.255.255.0
kevin(config-dhcp)#exit
kevin(config)#ip dhcp excluded-address 192.168.3.1
```

22. Program mitnik to pass DHCP requests to the DHCP router/server. It “helps” the router request from a workstation (from e0) for a dhcp address and directs the request to the dhcp server/router down the serial line.

```
mitnik(config)#interface e0/0
mitnik(config-int)#ip helper-address 192.168.1.2
```

23. Do a release and renew on workstation B’s IP address.
24. Test ping from workstation A to B and B to A. Everything should work just fine.

**Supplemental Lab or Challenge Activity:**
1. Go to [www.cisco.com](http://www.cisco.com) or use the help functions of your router to find out more ways to use the commands available with dhcp.
2. How many DHCP address pools can you set up on one router?
3. How does a DHCP server differ from a DNS server? What command could you use to enable a router to use a domain server?

**So What Have I Learned Here?**
In the first part you learned how to renew and release IP addresses from a workstation. In this lab you learned how to set up a router as a DHCP server. I really wouldn’t recommend using your router as a DHCP server if you could at all help it…why spends several thousand dollars for a router to act as a DHCP server when you could just set up an old workstation to do the same? Your call not mine.

---

**Guest Router Name Derivation**

To *some*, Kevin Mitnik is an icon in the hacking community. In 1986 he was arrested for breaking into the Digital Equipment Corporation network. He was arrested in 1995 again for allegedly stealing 20,000 credit card numbers, but was actually convicted for illegal use of cellular numbers. He was a regular contributing writer and guest lecturer at hacking conventions like Defcon. Too bad his last conviction prohibits him from ever using a computer, a telephone, or receiving monetary compensation from appearances and articles. Bummer, all that knowledge and he has to give it away for free…but that is what hackers are about anyways.
Subnetting with DHCP

Objectives:
To learn how to design a network with a router used as a DHCP server.

Tools and Materials:
(3) PC/workstations
(3) Routers (one must be a 2620)
(3) Switches
(6) Straight-through cables
(2) DCE serial cable
(2) DTE serial cable
(3) rollover cables

Lab Diagram:

Addressing:

<table>
<thead>
<tr>
<th>Routers</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hostnames</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E0/0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E0/1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S0/0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S0/1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workstations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GW</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Step-by-Step Instructions:
You are the network administrator for a small hospital in Vermont. The changeover from one hospital management group to another has caused massive cut-backs throughout the hospital staff and resources. You have been told to reduce or eliminate your expenditures and that all purchases are on hold. The only problem is you really, really needed that new server to set up a website/intranet for the staff. Heck, it would have really made your life so much better, but you just do not have the funds now. Since you remembered you can set up routers to act as DHCP servers you came up with an ingenious plan to cannibalize your DHCP server (and make it your new web server) and decided to “make it so.” Geeze, your boss will be wondering how you still managed to make it all work even with out the new server, right? No, they will probably cut you back more. Welcome to life on Dilbert’s side. So your task is to:

1. Design a network addressing scheme using private IP addresses. The router on the far left of the diagram above should be the dhcp router/server. Each Ethernet interface should have a different private IP address class.
2. Cable the lab as shown.
3. Complete the basic router setup on each router.
4. Configure the interfaces on each router.
5. Configure the routing protocol and advertise the router’s networks.
6. Configure DHCP and IP helper.
7. Hang a loopback interface off the dhcp router with an address of 1.1.1.1 to “simulate” web access.
8. Setup the workstations with IP address, subnet masks, and gateways addresses. You will need to reboot the workstations.
9. Test connectivity from all workstations to the others.

So What Have I Learned Here?
In this lab you learned how to apply your knowledge of routing and DHCP. In those “other” lab books you never really get a change to think for yourself. Here, instead of mindlessly cranking out commands step-by-step you have to use your brain. Let’s just hope you have enough Dew to stay awake.

Take college courses on a new schedule--yours.
Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education. Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

http://e.spcollege.edu

St. Petersburg College’s eCampus. Your time. Your place. Your future.
Are you enjoying the materials? Well be on the lookout for some other manuals and textbooks on
http://www.lulu.com/learningbydoing and http://www.spcollege.edu/star/cisco

The “Script Kiddie Cookbook” Available Mid-August 2004 at http://www.lulu.com/learningbydoing

This book title is based upon the Anarchist Cookbook that came out in the 1980’s. In that tremendously popular book it showed you how to build bombs at home. Similarly the Script Kiddie Cookbook will show you step by step hacking and hacking methodology. You will go to jail if you use this information improperly!!!

Learn the basics about computer security with this book. This book assumes you know nothing about security and starts there. All tools and software used are freeware. Other books will build upon the materials in this book. Learn about the nuts and bolts of SPAM, port scanning, Knoppix tools, cookies, and bring it all together with exercises in hacking at legal gaming sites.

GET THE BOOK THE GOVERNMENT DOES NOT WANT YOU TO OWN!!!

In preparation for release in 12-18 months:
Basham, M.J. Learning by Doing: Acceptable Use Policy's and their implementation in networking.
Basham, M.J. Learning by Doing: CCNP Switching Essentials.
**Static and Dynamic Routing**

**Objective:**
Internet Service Providers (ISP’s) route information to customers one of two ways, either by running a dynamic routing protocol, such as Open Shortest Path First (OSPF) or Border Gateway Protocol (BGP), or by pointing a static route to the customer’s networks. In this case we will configure the dynamic routing protocol RIP on our internal network and use a default static route to the ISP on our border router. The ISP will be able to reach our web servers, represented by loopbacks on Router1 and Router2, by using a static route. The ISP’s loopbacks represent networks somewhere on the Internet and will be reachable by the end of the lab.

**Tools and Materials:**
(4) 2500 or 2600 series routers
(1) Rollover cable
(2) V.35 Serial Cable
(1) Cross Connect Cable

**Background:**
Routing Information Protocol (RIP) is the dynamic routing protocol that will be used. A lot of small networks run this routing protocol because of its simplicity. RIP only has one metric, hops, and as a result makes no allocations for link speed. One of the things we network administrators like to do is use a private ip addressing scheme for our internal network because it allows us a great deal of versatility and virtually an endless supply of ip addresses (more than enough 16,777,216 ish), not to mention added security by using ranges of ip addresses that are not publicly routable. All configurations where tested and verified on four Cisco 2610 Access layer Routers running version 12.0(3)T3.

**Lab Diagram:**

```
ISP
  +-------------------+  +-------------------+
  | Loopback 0 1.1.1.1/8 |
  | Loopback 1 2.2.2.2/8 |
  +-------------------+  +-------------------+
  | Ethernet 0/0 64.96.12.2/30 |
  | Ethernet 0/0 64.96.12.1/30 |
  +-------------------+  +-------------------+
  | Border |
  +-------------------+  +-------------------+
  | Serial 0/0 192.168.1.9/30 |
  | Serial 0/0 192.168.1.10/30 |
  +-------------------+  +-------------------+
  | Router1 |
  +-------------------+  +-------------------+
  | Loopback 0 11.0.0.1/8 |
  | Serial 0/1 192.168.1.6/30 |
  | Loopback 0 12.0.0.1/8 |
  +-------------------+  +-------------------+
  | Router2 |
  +-------------------+  +-------------------+
  | 192.168.1.8/30 |
  | 192.168.1.10/30 |
  +-------------------+  +-------------------+
  | RIP Networks |
```
Step-by-step instructions:

1.) The first thing you must do is set up the lab in accordance with the diagram provided.
2.) Configure ISP, Border, Router1 and Router2 routers with serial, Ethernet and loopback interface addresses as specified.
3.) Verify all necessary interfaces are in a state of “up and up” by issuing the following command: (Remember Serial links must have one side DCE (Clock rate) with the other side DTE)

<table>
<thead>
<tr>
<th>ISP</th>
<th>sh ip int bri</th>
<th>Interface</th>
<th>IP-Address</th>
<th>OK? Method</th>
<th>Status</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Ethernet0/0</td>
<td>64.96.12.2</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Serial0/0</td>
<td>unassigned</td>
<td>YES NVRAM</td>
<td>administratively down down</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Serial1/1</td>
<td>unassigned</td>
<td>YES NVRAM</td>
<td>administratively down down</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Loopback0</td>
<td>1.1.1.1</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Loopback1</td>
<td>2.2.2.2</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>border</th>
<th>sh ip int bri</th>
<th>Interface</th>
<th>IP-Address</th>
<th>OK? Method</th>
<th>Status</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Ethernet0/0</td>
<td>64.96.12.1</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Serial0/0</td>
<td>unassigned</td>
<td>YES NVRAM</td>
<td>administratively down down</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Serial1/0</td>
<td>192.168.1.5</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Router1</th>
<th>sh ip int bri</th>
<th>Interface</th>
<th>IP-Address</th>
<th>OK? Method</th>
<th>Status</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Ethernet0/0</td>
<td>unassigned</td>
<td>YES NVRAM</td>
<td>administratively down down</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Serial0/0</td>
<td>192.168.1.9</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Serial1/0</td>
<td>192.168.1.6</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Loopback0</td>
<td>11.0.0.1</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Router2</th>
<th>sh ip int bri</th>
<th>Interface</th>
<th>IP-Address</th>
<th>OK? Method</th>
<th>Status</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Ethernet0/0</td>
<td>unassigned</td>
<td>YES NVRAM</td>
<td>administratively down down</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Serial0/0</td>
<td>192.168.1.10</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Serial1/0</td>
<td>unassigned</td>
<td>YES NVRAM</td>
<td>down</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Loopback0</td>
<td>12.0.0.1</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
</tbody>
</table>

4.) Once all necessary interfaces are in a state of “up and up” we can program our dynamic protocol RIP and our static routes.
5.) Enter the following commands on Router2 to configure RIP:

```
Router2#conf t
Router2(config)#router rip
Router2(config-router)#network 12.0.0.0
Router2(config-router)#network 192.168.1.8
```
6.) Enter the following commands on Router1 to configure RIP:

```
Router1#conf t
Router1(config)#router rip
Router1(config-router)#network 192.168.1.8
Router1(config-router)#network 192.168.1.4
Router1(config-router)#network 11.0.0.0
```

7.) Enter the following commands on border to configure RIP:

```
border#conf t
Enter configuration commands, one per line. End with CNTL/Z.
border(config)#router rip
border(config-router)#network 192.168.1.4
border(config-router)#^Z
border#
```

8.) Now let's check to make sure RIP is converged (all routers know about all RIP networks) an operating properly before we continue. Issue the following commands on border:

```
border#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
   D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
   N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
   E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
   i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
   U - per-user static route, o - ODR, P - periodic downloaded static route
   T - traffic engineered route

Gateway of last resort is not set

64.0.0.0/30 is subnetted, 1 subnets
C  64.96.12.0 is directly connected, Ethernet0/0
R  11.0.0.0/8 [120/1] via 192.168.1.6, Serial0/1
R  12.0.0.0/8 [120/2] via 192.168.1.6, Serial0/1
   192.168.1.0/30 is subnetted, 2 subnets
R  192.168.1.8 [120/1] via 192.168.1.6, Serial0/1
C  192.168.1.4 is directly connected, Serial0/1
```

Examine the routing table. We can see our two loopbacks from Router1 (11.0.0.0/8) and Router2 (12.0.0.0/8), we also see the network connection between Router1 and Router2 (192.168.1.8/30). If your routing table does not look similar to this, start back at step 3 and make sure all commands have been entered.
I want to take a minute to decipher the different entries in the routing table and explain what they all mean. Let’s examine the entry for the 11.0.0.0/8 network:

```
R  11.0.0.0/8 [120/1] via 192.168.1.6, Serial0/1
```

- **R**: Stands for RIP
- **11.0.0.0/8**: Network Learned from RIP
- **[120/1]**: RIP’s Administrative Distance/Metric (Hops. RIP’s only metric)
- **via 192.168.1.6, Serial0/1**: Information was learned from this ip address
- **Information was received through this interface (local)**

Take a look at the other entries and see how the border router learned about those respective networks.

Generally speaking, if we one of our routing tables is totally converged, we can assume all routing tables have the necessary information, but we all know what happens when we assume! So let’s look at the other routing tables:

**Router1**

```
Router1#sh ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR, P - periodic downloaded static route
T - traffic engineered route

Gateway of last resort is not set

C    11.0.0.0/8 is directly connected, Loopback0
R    12.0.0.0/8 [120/1] via 192.168.1.10, Serial0/0
    192.168.1.0/30 is subnetted, 2 subnets
    C       192.168.1.8 is directly connected, Serial0/0
    C       192.168.1.4 is directly connected, Serial0/1
```

**Router2**

```
Router2#sh ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR, P - periodic downloaded static route
T - traffic engineered route

Gateway of last resort is not set
ISP#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
     D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
     i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
     U - per-user static route, o - ODR, P - periodic downloaded static route
     T - traffic engineered route

Gateway of last resort is not set

C  1.0.0.0/8 is directly connected, Loopback0
C  2.0.0.0/8 is directly connected, Loopback1
     64.0.0.0/30 is subnetted, 1 subnets
     64.96.12.0 is directly connected, Ethernet0/0

9.) Now we need to configure a static route on the ISP to reach our web servers on
Router1 and Router2. Enter the following commands on ISP:

ISP#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ISP(config)#ip route 12.0.0.0 255.0.0.0 64.96.12.1
ISP(config)#ip route 11.0.0.0 255.0.0.0 ethernet 0/0
ISP(config)#^z
ISP#

The format for entering static routes is described below.
From global configuration mode we enter:

ISP(config)# ip route N.N.N.N S.S.S.S [Next-Hop Ip address OR Exit Interface]

Where N.N.N.N is the destination Network and S.S.S.S is the destination
Network’s Subnet Mask. The last entry specifies how to reach the previously
entered Network. We can either specify Next-Hop Ip address OR the interface
through which we want traffic destined for the described Network to Exit. With
the first entry I chose to use Next-Hop Ip address:
ISP(config)#ip route 12.0.0.0 255.0.0.0 64.96.12.1 (Where is this ip address?)
N.N.N.N  S.S.S.S  Next-Hop Ip address
The second entry I chose to use Exit Interface:
ISP(config)#ip route 11.0.0.0 255.0.0.0 Ethernet 0/0
N.N.N.N  S.S.S.S  Exit Interface

Both will accomplish the same task which is to let the ISP router know how to reach the web servers on Router1 and Router2. Static routes can be entered on any router running any routing protocol (or running no routing protocol). Be careful because static routes using Next-Hop Ip address have an administrative distance of 1 and static routes using Exit Interface have and administrative distance of 0 and show up as directly connected. I say be careful because those entries supercede all dynamic routing protocols. Static routes are only recommended when there is only one way in and one way out of a network.

Verify the entries in the routing table:

ISP>sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR, P - periodic downloaded static route
T - traffic engineered route

Gateway of last resort is not set

C   1.0.0.0/8 is directly connected, Loopback0
C   2.0.0.0/8 is directly connected, Loopback1
   64.0.0.0/30 is subnetworked, 1 subnets
C   64.96.12.0 is directly connected, Ethernet0/0
S   11.0.0.0/8 is directly connected, Ethernet0/0
S   12.0.0.0/8 [1/0] via 64.96.12.1

So we should be able to ping our web servers located at 12.0.0.1 and 11.0.0.1, right? Let’s try:
ISP>ping 12.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
ISP>ping 11.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 11.0.0.1, timeout is 2 seconds:
....
Success rate is 0 percent (0/5)
ISP>
Both pings were unsuccessful? What happened? We told the ISP how to get to the web servers, so why was the ping unsuccessful? To answer this question we need to go to the border router and take a look at the routing table:

```
border#sh ip route
Codes:  C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR, P - periodic downloaded static rout
       T - traffic engineered route

Gateway of last resort is not set

   64.0.0.0/30 is subnetted, 1 subnets
C       64.96.12.0 is directly connected, Ethernet0/0
R    11.0.0.0/8 [120/1] via 192.168.1.6, Serial0/1
R    12.0.0.0/8 [120/2] via 192.168.1.6, Serial0/1

192.168.1.0/30 is subnetted, 2 subnets
R    192.168.1.8 [120/1] via 192.168.1.6, Serial0/1
C    192.168.1.4 is directly connected, Serial0/1
```

The entries are all there for the RIP networks and we are directly connected to the service provider, so what’s the problem? We need to go back to the OSI model to answer that question.

Layer 3 of the OSI can be described as our “map”. This is where routing takes place and the router’s routing table is maintained. ALL packets (layer 3 Protocol Data Unit (PDU)) must have a source and destination address. When we generate a ping there are two main components; Echo-Request and Echo-Reply. The request is generated when we type the word “ping” and then an Ip address or www.cisco.com. So we were in the ISP router and typed “ping 12.0.0.1” and “ping 11.0.0.1”, this was the Echo-Request. This packet has a destination ip address of 12.0.0.1 and 11.0.0.1 respectively. The router looks in the routing table to see how to reach those networks. (Remember we told the router how to get to those networks by entering the static routes from step 9). So why didn’t the ping succeed? Remember a packet has both a source and destination ip address, well it knows how to get to the destination, but once the packet gets there it does not know how to get back. Router1’s web server (11.0.0.1) and Router2’s web server (12.0.0.1) generate an Echo-Reply to 64.96.12.2 (the source of the Echo-Request). There is one problem, both Router1 and Router2 do not have an entry
in their routing table for the 64.96.12.0 network, so the Echo-Reply never gets there, it is in effect unroutable. 
So how do we fix this? Remember I said static routes can be used in conjunction with dynamic routing protocols and “Static routes are only recommended when there is only one way in and one way out of a network.” To make this work we must go to the border router. Let’s first look at the border router’s routing table:

border#sh ip route
Codes:  C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP 
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area 
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2 
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP 
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default 
       U - per-user static route, o - ODR, P - periodic downloaded static rout 
       T - traffic engineered route 

Gateway of last resort is not set

64.0.0.0/30 is subnetted, 1 subnets 
  C       64.96.12.0 is directly connected, Ethernet0/0 
  R    11.0.0.0/8 [120/1] via 192.168.1.6, Serial0/1 
  R    12.0.0.0/8 [120/2] via 192.168.1.6, Serial0/1 
  192.168.1.0/30 is subnetted, 2 subnets 
  C       192.168.1.8 [120/1] via 192.168.1.6, Serial0/1 
  C       192.168.1.4 is directly connected, Serial0/1 

We have no routes to the ISP’s loopbacks, this can be fixed by putting a default static route on the border router to the service provider (Remember there is only one way in and one way out of our network). Notice the “Gateway of last resort is not set”. This will change after we enter a default static route in the border router. Issue the following commands on the border router:

border#conf t 
Enter configuration commands, one per line. End with CNTL/Z. 
border(config)#ip route 0.0.0.0 0.0.0.0 64.96.12.2 
border(config)#^Z 
border# 

The eight zeros we entered after “ip route” tell the router if it can’t find the destination network in it’s routing table, then send it to this ip address. This is referred to as the “Gateway of last resort”. Basically saying, “if all else fails, send the packet here, they will know what to do with it”. (We used Next-Hop ip address, but you could have also used Exit-Interface).

Now let’s look at the routing table on the border router: 
border#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR, P - periodic downloaded static route
T - traffic engineered route

Gateway of last resort is 64.96.12.2 to network 0.0.0.0

64.0.0.0/30 is subnetted, 1 subnets
C  64.96.12.0 is directly connected, Ethernet0/0
R  11.0.0.0/8 [120/1] via 192.168.1.6, Serial0/1
R  12.0.0.0/8 [120/2] via 192.168.1.6, Serial0/1
192.168.1.0/30 is subnetted, 2 subnets
R  192.168.1.8 [120/1] via 192.168.1.6, Serial0/1
C  192.168.1.4 is directly connected, Serial0/1
S*  0.0.0.0/0 [1/0] via 64.96.12.2

Most of the time setting the Gateway of last resort on the border router is all we need to do. This change in the border router should trickle down to Router1 and Router2. Let’s take a look:

**Router1#sh ip route**

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR, P - periodic downloaded static route
T - traffic engineered route

Gateway of last resort is 192.168.1.5 to network 0.0.0.0

C  11.0.0.0/8 is directly connected, Loopback0
R  12.0.0.0/8 [120/1] via 192.168.1.10, Serial0/0
192.168.1.0/30 is subnetted, 2 subnets
C  192.168.1.8 is directly connected, Serial0/0
C  192.168.1.4 is directly connected, Serial0/1
R*  0.0.0.0/0 [120/1] via 192.168.1.5, Serial0/1
```

Notice how the Gateway of last resort was learned through RIP. Let’s look at Router2:

**Router2#sh ip route**

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR, P - periodic downloaded static route
T - traffic engineered route

Gateway of last resort is 192.168.1.5 to network 0.0.0.0

C  11.0.0.0/8 is directly connected, Loopback0
R  12.0.0.0/8 [120/1] via 192.168.1.10, Serial0/0
192.168.1.0/30 is subnetted, 2 subnets
C  192.168.1.8 is directly connected, Serial0/0
C  192.168.1.4 is directly connected, Serial0/1
R*  0.0.0.0/0 [120/1] via 192.168.1.5, Serial0/1
```
Gateway of last resort is 192.168.1.9 to network 0.0.0.0

R    11.0.0.0/8 [120/1] via 192.168.1.9, Serial0/0
C    12.0.0.0/8 is directly connected, Loopback0
     192.168.1.0/30 is subnetted, 2 subnets
C    192.168.1.8 is directly connected, Serial0/0
R    192.168.1.4 [120/1] via 192.168.1.9, Serial0/0
R*   0.0.0.0/0 [120/2] via 192.168.1.9, Serial0/0

Notice how the Gateway of last resort was learned through RIP.

Remember I said “Most of the time setting the Gateway of last resort on the border router is all we need to do.” With RIP this is true, most of the time. If your routing tables do not look like Router1 and Router2 shown above, you will have to enter and additional command.

Go back to the border router and enter the following command:

```
border#conf t
Enter configuration commands, one per line. End with CNTL/Z.
border(config)#router rip
border(config-router)#default-information originate
border(config-router)#^Z
border#clear ip route *
border#
```

Now check Router1 and Router2 to verify the routing tables have the Gateway of Last resort set. The routing tables should look like the ones above.

Let’s go to the ISP and try our pings again:

```
ISP>ping 11.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 11.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/7/8 ms
ISP>ping 12.0.0.1
```

368
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/10/12 ms

1.) Now go to Router1 and try to ping 1.1.1.1 and ping 2.2.2.2. Did it work?
   Why not?
2.) Go to Router2 and try to ping 1.1.1.1 and ping 2.2.2.2. Did it work? Why not?
3.) Try an extended ping from Router2:

   ```
   Router2#ping
   Protocol [ip]:
   Target IP address: 1.1.1.1
   Repeat count [5]:
   Datagram size [100]:
   Timeout in seconds [2]:
   Extended commands [n]: y
   Source address or interface: 12.0.0.1
   Type of service [0]:
   Set DF bit in IP header? [no]:
   Validate reply data? [no]:
   Data pattern [0xABCD]:
   Loose, Strict, Record, Timestamp, Verbose[none]:
   Sweep range of sizes [n]:
   Type escape sequence to abort.
   Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
   !!!!!
   Success rate is 100 percent (5/5), round-trip min/avg/max = 4/7/8 ms
   ```

4.) Try and extended ping from Router1:

   ```
   Router1#ping
   Protocol [ip]:
   Target IP address: 1.1.1.1
   Repeat count [5]:
   Datagram size [100]:
   Timeout in seconds [2]:
   Extended commands [n]: y
   Source address or interface: 11.0.0.1
   Type of service [0]:
   Set DF bit in IP header? [no]:
   Validate reply data? [no]:
   Data pattern [0xABCD]:
   Loose, Strict, Record, Timestamp, Verbose[none]:
   Sweep range of sizes [n]:
   ```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms
Router1#

5.) Why does an extended ping work and a regular ping not work? Think about
Echo Request and Echo-Reply.

Looking for college courses?  
We deliver.

Choose from over 400 online courses in dozens of majors, including our new bachelor’s
degree program in Technology Management. St. Petersburg College’s eCampus
delivers Internet convenience and the quality instruction that has made SPC an
international leader in online education.

Visit our web site for a full listing of courses and programs, along with support services
like online registration, academic advising and financial aid.

St. Petersburg College’s eCampus
http://e.spcollege.edu

Your time. Your place. Your future.
Static and Dynamic Routes with Discontiguous RIP Networks

Objective:
To learn how static routes can be used in network design to overcome the problems encountered with discontiguous networks.

Background:
In our earlier lab you learned about route summarization. In that lab you learned what routes are passed with RIP and which ones are not. Just suppose we inherited our network with some given IP addresses and re-assigning IP addresses was not an option. We could use a static route to be able to “route” between what was once “un-routable.”

Tools and Materials:
(2) PC/workstations
(2) Routers
(2) Switches
(4) Straight-through cables
(1) DCE serial cable
(1) DTE serial cable
(2) rollover cables

Lab Design:

Addressing:

<table>
<thead>
<tr>
<th></th>
<th>Phiber</th>
<th>Optik</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>161.20.4.1/30 (DCE)</td>
<td>n/a</td>
</tr>
<tr>
<td>S1</td>
<td>n/a</td>
<td>161.20.4.2/30 (DTE)</td>
</tr>
<tr>
<td>L0</td>
<td>161.20.3.1/30</td>
<td>161.20.5.1/30</td>
</tr>
<tr>
<td>E0</td>
<td>161.20.2.1/24</td>
<td>161.20.1.1/24</td>
</tr>
</tbody>
</table>
Workstations  | A             | B             
-------------|---------------|---------------
IP           | 161.20.2.2    | 161.20.1.2    
SM           | 255.255.255.0 | 255.255.255.0 
GW           | 161.20.2.1    | 161.20.1.1    

Step-by-Step Instructions:
1. Cable and set up the lab as shown.
2. Complete the basic router setup on each router.
3. Configure the interfaces on each router.
4. Configure the routing protocol and advertise/associate/publish the router’s networks. Configure the workstations. You should NOT be able to ping from workstation A to workstation B or vice versa. You should be able to ping from workstation A or B to either loopback. And then try showing the route from ...you should see the loopback interface for Phiber (learned via RIP) in the routing table for Optik:

    optik# sh ip route
    Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - EGP
          D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
          N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
          E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
          i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
          default U - per-user static route, o - ODR
    Gateway of last resort is not set
        161.20.0.0/16 is variably subnetted, 4 subnets, 2 masks
        C   161.20.5.0/30 is directly connected, Loopback0
        C   161.20.4.0/30 is directly connected, Serial0/1
        C   161.20.1.0/24 is directly connected, Ethernet0/0
        R   161.20.3.0/30 [120/1] via 161.20.4.1, 00:00:06, Serial0/1
    optik#

5. So let’s fix that little problem here:

    optik(config)# ip route 161.20.2.0 255.255.255.0 161.20.4.1

What this line says to the router is “to get to the network 161.20.2.0/24 use the interface with the address of 161.20.4.1.” (note: it’s the address on the far side of the serial line…more on that in a bit). Now a request from workstation B to the Ethernet interface has directions on how to get to that address. We have provided them to the router with manual (static) instructions. Our router has summarized our networks because of the addresses we used but, ha-ha!, we are one step ahead of that router because we let it know who’s the boss by slapping a static route in there…take that!

6. Now you should be able to ping from workstation B to the Ethernet interface on Phiber and to workstation A. Now try to ping from workstation A to B. You should not be able to ping. This is because Phiber has no way to direct traffic, even though we did it on Optik. We must add another static route from Phiber to
Optik to allow workstation A to be able to ping workstation B. Go ahead and add the route. (Can’t tell you everything step-by-step, otherwise you wouldn’t learn much…ok…if you get stuck you can check the answers.)

7. Static routes are really good for troubleshooting. Later on when you learn about setting up routers with multiple routes to a destination you will learn to use static routes to “force” communication over one path in particular to test that specific path. Suppose the route given by the “source router” in the picture above was chosen by the source router to be the “best path” to the destination router. But we wanted to test the capabilities of a “lesser path” (given as “destination router”) to the destination router. We could force the route with a static route.

8. We can actually specify the interface, rather than using the IP address for setting up a static route (told you we’d come back to it!). So instead of this:

```
optik(config)#ip route 192.168.1.0 255.255.255.0 179.40.6.1
```

For the same thing we could use this:

```
optik(config)#ip route 192.168.1.0 255.255.255.0 serial0/0
```

If you forget your options then use your help command:

```
optik(config)#ip route 192.168.1.0 255.255.255.0 ?
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.B.C.D</td>
<td>Forwarding router's address</td>
</tr>
<tr>
<td>FastEthernet</td>
<td>FastEthernet IEEE 802.3</td>
</tr>
<tr>
<td>Loopback</td>
<td>Loopback interface</td>
</tr>
<tr>
<td>Null</td>
<td>Null interface</td>
</tr>
<tr>
<td>Serial</td>
<td>Serial</td>
</tr>
</tbody>
</table>

Now let’s explore some of the other options for static routes:

```
optik(config)#ip route 1.0.0.0 255.0.0.0 serial0/1 ?
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;1-255&gt;</td>
<td>Distance metric for this route</td>
</tr>
<tr>
<td>A.B.C.D</td>
<td>Forwarding router's address</td>
</tr>
<tr>
<td>name</td>
<td>Specify name of the next hop</td>
</tr>
<tr>
<td>permanent</td>
<td>permanent route</td>
</tr>
<tr>
<td>tag</td>
<td>Set tag for this route</td>
</tr>
</tbody>
</table>
The first option we see a distance metric for this route. Each routing protocol has a different default distance metric assigned to it. RIP has a default static route distance of 120. So actually we already put that in our command, even though it does not appear in our ip route command. What this is used for is when we want to put in more than one static route on our router. The router will automatically select the static route with the lowest distance metric first then, if that route is not available, go to the route with the second lowest distance metric and then so on. Distance metrics, as you can see, vary from 1 to 255. Here are some common metrics for you to know about here at this time:

- Connected interface 0
- Static route 1
- RIP 120
- Unknown 255

If we were to add another router in then we would need to add in another static route. Using that methodology if we had a network with many routers we could bury ourselves in static routes which has the possibility of causing major problems. In our example we just did instead of setting a static route between the two routers we could set a default network route on optik. This will essentially allow us to add routers at will without all those static routes. Setting many static routes essentially defeats the purposes of having routers make decisions anyways. So there. In the next couple of labs you will learn more about different types of routes and their uses. In the meantime let’s try to do some more exercises and learn by doing!

9. Ok. Let’s try putting a loop back into our network. Connect another serial line from s0/1 (DTE) on phiber to s0/0 (DCE) on optik. Use 56000 for the clock rate. We know from our routing loop labs that our split-horizon is set by default to prevent routing loops, but if we have two paths wouldn’t we want to take advantage of that? Absolutely! If all of our metrics are equal, then our routers will perform load-balancing across the equal lines. Now, of course, you know we can change that. The command to change load-balancing is “variance.” Use your knowledge of the CISCO technical support site and router help features to find out more about this command and how to use it. What we are more concerned with in this lab is static routing. Set your new serial connection to have a different administrative distance than the main line so it will act as a backup line.

10. Ping and trace the route between workstation A and B.

11. Take the main line down (just unplug one end of the serial cable) and ping and trace the route again. Remember RIP may take a few seconds to “catch” up. Your traffic should now be re-routed across the back up line.

12. Bring the main line back up and re-ping and re-trace the route. Unless you used the “permanent” suffix to the ip route command the back up line should still be the preferred line. But…you know how to fix that too.

**Supplemental Labs or Challenge Activities:**
1. Set a whole network with 4-5 routers with routes that are summarized and use static lines to enable “routing.” Now you can see why we don’t always prefer using them.
2. Find out what the other administrative distances are for the other routing protocols. Hint: look on CISCO’s website.

So What Have I Learned Here?
In this lab you learned that, while useful, static routes can become a pain in the admin. It is best to do dynamic routing only when absolutely necessary. Later, as you progress in your studies you will better learn when and where to use static routes. But for now just forget about them.

Guest Router Name Derivation
Phiber Optik was the leader of the Master’s of Deception (MoD) hackers ring in New York City in the 1980’s/early 1990’s. Allegedly he master-minded the Martin Luther King day crash of AT&T’s national phone service in 1990. Known for his daring actions and media stunts he appeared or was interviewed in many publications including Harper’s, Esquire, and the New York Times. Don’t worry…he got busted. Turk 182!

The “Script Kiddie Cookbook” Available Mid-September 2004 at http://www.lulu.com/learningbydoing

This book title is based upon the Anarchist Cookbook that came out in the 1980’s. In that tremendously popular book it showed you how to build bombs at home. Similarly the Script Kiddie Cookbook will show you step by step hacking and hacking methodology. You will go to jail if you use this information improperly!!!

Learn the basics about computer security with this book. This book assumes you know nothing about security and starts there. All tools and software used are freeware. Other books will build upon the materials in this book. Learn about the nuts and bolts of SPAM, port scanning, Knoppix tools, cookies, and bring it all together with exercises in hacking at legal gaming sites.

GET THE BOOK THE GOVERNMENT DOES NOT WANT YOU TO OWN!!!
Overcoming Problems with Routing Loops

Objective:
To understand the problems routing loops can cause in a network and how to overcome those problems.

Tools and Materials:
(3) PC/workstations
(3) Routers
(3) Switches
(6) Straight-through cables
(3) DCE serial cable
(3) DTE serial cable
(3) rollover cables

Lab Diagram:

Addressing:

<table>
<thead>
<tr>
<th>Routers</th>
<th>Hostnames</th>
<th>Prophet</th>
<th>Knight</th>
<th>Lightning</th>
</tr>
</thead>
<tbody>
<tr>
<td>E0</td>
<td>172.16.1.1/16</td>
<td>172.16.2.1/16</td>
<td>172.16.3.1/16</td>
<td></td>
</tr>
<tr>
<td>S0</td>
<td>10.1.1.1/24 (DCE)</td>
<td>10.2.1.1/24 (DCE)</td>
<td>10.3.1.2/24 (DTE)</td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>10.3.1.1/24 (DTE)</td>
<td>10.1.1.2/24 (DTE)</td>
<td>10.2.1.2/24 (DTE)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workstations</th>
<th>a</th>
<th>b</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>172.16.1.2/16</td>
<td>172.16.2.2/16</td>
<td>172.16.3.2/16</td>
</tr>
<tr>
<td>SM</td>
<td>255.255.0.0</td>
<td>255.255.0.0</td>
<td>255.255.0.0</td>
</tr>
<tr>
<td>GW</td>
<td>172.16.1.1</td>
<td>172.16.2.1</td>
<td>172.16.3.1</td>
</tr>
</tbody>
</table>

Step-by-Step Instructions:
2. Cable the lab as shown except for the serial line between Prophet and Lightning.
3. Complete the basic router setup on each router.
4. Configure the interfaces on each router.
5. Configure the routing protocol and advertise the router’s networks.
6. Setup the workstations with IP address, subnet masks, and gateways addresses. You will need to reboot the workstations. If they ask for a password for network connectivity just put anything in and you should see a message something like “no domain server is available, you may not have some networking functions.” It’s ok if you see it, but you probably will not be able to ping outside of your workstation without seeing that error message.

7. Test connectivity from workstation A to workstation C.

8. Turn on `debug ip rip` on each router.

```
00:14:56: RIP: sending v1 update to 255.255.255.255 via Ethernet0/0 (172.16.1.1)
00:14:56: RIP: build update entries
00:14:56: network 10.0.0.0 metric 1
00:14:56: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (10.1.1.1)
00:14:56: RIP: build update entries
00:14:56: network 10.0.0.0 metric 1
00:14:56: subnet 10.1.1.0 metric 1
00:14:56: subnet 10.2.1.0 metric 2
00:14:56: network 172.16.0.0 metric 1
00:15:08: RIP: received v1 update from 10.1.1.2 on Serial0/0
00:15:08: 10.2.1.0 in 1 hops
00:15:08: 172.16.0.0 in 1 hops
00:15:26: RIP: sending v1 update to 255.255.255.255 via Ethernet0/0 (172.16.1.1)
00:15:26: RIP: build update entries
00:15:26: network 10.0.0.0 metric 1
00:15:26: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (10.1.1.1)
00:15:26: RIP: build update entries
00:15:26: network 10.0.0.0 metric 1
00:15:26: subnet 10.1.1.0 metric 1
00:15:26: subnet 10.2.1.0 metric 2
00:15:26: network 172.16.0.0 metric 1
00:15:26: RIP: received v1 update from 10.1.1.2 on Serial0/0
00:15:26: 10.2.1.0 in 1 hops
00:15:26: 172.16.0.0 in 1 hops
```

We can clearly see the 30 seconds between the updates here (56 and 26 seconds). We can also see we have a couple of routes with 1 or 2 hops away.

9. Now let’s add in that other serial line between Prophet and Lightning. This will create a routing loop in our network. By default CISCO routers are prepared for routing loops. To create a problem with a routing loop use this command:

```
prophet(config)#interface s0/1
prophet(config-if)#no ip split-horizon
prophet(config-if)#no auto-summary

lightning(config)#interface s0/0
lightning(config-if)#no ip split-horizon
lightning(config-if)#no auto-summary
```
10. You should see lots of debug messages about routing loops now. To stop those routing loop problems type “ip split-horizon” again on the serial interfaces or just disconnect that serial line. This problem is known as “counting to infinity.” Again, the proof is in the pudding:

```
00:16:50:       RIP: sending v1 update to 255.255.255.255
                via Ethernet0/0 (172.16.1.1)
00:16:50:       RIP: build update entries
00:16:50:       network 10.0.0.0 metric 1
00:16:50:       RIP: sending v1 update to 255.255.255.255
                via Serial0/0 (10.1.1.1)
00:16:50:       RIP: build update entries
00:16:50:       network 10.0.0.0 metric 1
00:16:50:       subnet 10.1.1.0 metric 1
00:16:50:       subnet 10.2.1.0 metric 2
00:16:50:       subnet 10.3.1.0 metric 16
00:16:50:       network 172.16.0.0 metric 1
00:17:20:       RIP: sending v1 update to 255.255.255.255
                via Ethernet0/0 (172.16.1.1)
00:17:20:       RIP: build update entries
00:17:20:       network 10.0.0.0 metric 1
00:17:20:       RIP: sending v1 update to 255.255.255.255
                via Serial0/0 (10.1.1.1)
00:17:20:       RIP: build update entries
00:17:20:       network 10.0.0.0 metric 1
00:17:20:       subnet 10.1.1.0 metric 1
00:17:20:       subnet 10.2.1.0 metric 2
00:17:20:       subnet 10.3.1.0 metric 16
00:17:20:       network 172.16.0.0 metric 1
```

Ok...here we can see an addition to our routing information since we installed a routing loop. The maximum hops with RIP is 16 and I think we pegged it pretty good. Heck, let’s even try pinging that interface…it should work, right?

```
lighning#ping 10.3.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
lighning#
```

**Supplemental Lab or Challenge Activity:**
1. You can also solve the problem of routing loops by changing the metrics for the routing protocol. Having just completed the lab on RIP metrics, try this lab again changing the metrics for RIP from 16 hops to 3 and see what happens.
2. Define and differentiate between split-horizon, poison reverse update and count to infinity. More than likely you will see a question about these on your test.
So What Have I Learned Here?
In this lab you learned that problems with routing loops are automatically taken care of by the `ip split-horizon` command in your router. Why did we bother learning about it? Well no network is pure and chances are you will have routers from other vendors in your network. No all of them automatically eliminate routing loop problems so you need to be aware of them.

Guest Router Name Derivation

In September 1998 a hacker known as “Prophet” (Robert Riggs) cracked the BellSouth network and downloaded copies of operating manuals to his own computer and copied them to a BBS. He also sent them to another hacker “Knight Lightning” (Craig Neidorf) who published the information in his underground electronic magazine “Phrack.” Prophet pled guilty to wire fraud. Knight Lightning fought his case because he had only taken a copy of the document and “didn’t hurt anything.” It turned out the document was also available for sale from BellSouth, but Knight Lightning was still left with a six-figure legal bill for a document he could have purchased legally for $13.00 and Prophet has a criminal record.
RIP Version 2 and Redistribution with RIP

**Objective:**
To learn about RIP version 2.

**Background:**
In our earlier lab you learned about route summarization. In that lab you learned what routes are passed with RIP and which ones are not. We learned that we could use a static route to be able to “route” between what was once “un-routable.” This was known as “auto-summarization” and, by default it is enabled with RIP (and cannot be disabled). We also learned that too many static routes can cause problems for us as administrators. Another way to solve that problem would have been to switch to a routing protocol that allowed subnet masks to be passed. One such protocol that does it is RIP version 2.

**Tools and Materials:**
(2) PC/workstations  
(2) Routers  
(2) Switches  
(4) Straight-through cables  
(1) DCE serial cable  
(1) DTE serial cable  
(2) rollover cables

**Lab Design:**

![Lab Diagram]

**Addressing:**

<table>
<thead>
<tr>
<th></th>
<th>Workstation “A”</th>
<th>Workstation “B”</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hostnames</strong></td>
<td><strong>Phiber</strong></td>
<td><strong>Optik</strong></td>
</tr>
<tr>
<td>S0</td>
<td>161.20.4.1/30 (DCE)</td>
<td>n/a</td>
</tr>
<tr>
<td>S1</td>
<td>n/a</td>
<td>161.20.4.2/30 (DTE)</td>
</tr>
<tr>
<td>L0</td>
<td>161.20.3.1/30</td>
<td>161.20.5.1/30</td>
</tr>
<tr>
<td>E0</td>
<td>161.20.2.1/24</td>
<td>161.20.1.1/24</td>
</tr>
</tbody>
</table>
**Step-by-Step Instructions:**

1. Cable and set up the lab as shown.
2. Complete the basic router setup on each router.
3. Configure the interfaces on each router.
4. Configure the workstations. You should NOT be able to ping from workstation A to anywhere. Silly billy...we haven’t configured a routing protocol yet.
5. So let’s fix that little problem here using RIP version 2. Configure the routing protocol and advertise the router’s networks using RIP version 2 by doing this:

   ```
   phiber(config)#router rip
   phiber(config-router)#network 161.20.0.0
   phiber(config-router)#version 2
   ```

   And on the other router:

   ```
   optik(config)#router rip
   optik(config-router)#network 161.20.0.0
   optik(config-router)#version 2
   ```

6. Now you should be able to ping from each workstation to the other workstation, to the loopbacks on both routers and everywhere. BAM! Problem solved much easier than with static routes. Yeah...it’s that easy. 😊

7. Now let’s take it up another level and add some more routers to our network (look for the lab diagram at the end of this lab). One router will act as an ISP and the other will be a new company we just acquired. They are running RIP on their network. The boss their likes RIP because she is familiar with it so you decide to leave it intact. But you need to be able to pass your routing information over your network so you use your knowledge of the CISCO website and find out information about two commands you can use to “redistribute” your routing protocol:

   ```
   ip rip send version 1
   ip rip receive version 1
   ```

8. Also you do not want your ISP to have the information about your network so you decided to stop all routing table broadcasts out the serial interface on phiber. You enter this command:

   ```
   phiber(config)#router rip
   phiber(config-router)#passive-interface serial0/1
   ```

9. Use your knowledge of debug commands, both before and after implementing the passive interface command, to verify it is working properly. Heck, even a show ip route would work too.
10. Did you remember to statically connect your network to the ISP? Tsk, tsk.

ISP

a.k.a

“Border Router”

Workstation “A”

Workstation “B”

Workstation “C”

Addressing:

<table>
<thead>
<tr>
<th>Routers</th>
<th>Hostnames</th>
<th>ISP</th>
<th>RIPv1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E0</td>
<td>n/a</td>
<td>192.168.1.1/24</td>
</tr>
<tr>
<td></td>
<td>L0</td>
<td>172.16.1.1/16</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>S0</td>
<td>220.221.222.253/30 (DCE)</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>S1</td>
<td>n/a</td>
<td>161.20.6.2/24 (DTE)</td>
</tr>
</tbody>
</table>

*So What Have I Learned Here?*

In this lab you learned about RIP version 2. It does pass the subnet masks so we can use that VLSM that we learned about a couple of labs ago...See, a place for everything and everything in its place. Isn’t that nice? I think that will about do it for part 2. In the next couple of parts you will learn about switching and then start picking up more routing protocols now that you have mastered the basics of your router, RIP, and troubleshooting.

**Guest Router Name Derivation**

Phiber Optik was the leader of the Master’s of Deception (MoD) hackers ring in New York City in the 1980’s/early 1990’s. Allegedly he master-minded the Martin Luther King day crash of AT&T’s national phone service in 1990. Known for his daring actions and media stunts he appeared or was interviewed in many publications including Harper’s, Esquire, and the New York Times. Don’t worry…he got busted. Turk 182!
Electronic technology continues to infiltrate industry, electrostatic discharge threatens operations causing costly damage, injury and downtime. Pro-Cote Inc. can protect your work environment against destructive static shock with Valspar ESD Epoxy Flooring Systems.

Certified Conductive and Static Dissipative applications for every need:

- Facilities with highly sensitive electronic equipment
- Computer facilities
- Electronic manufacturing & assembly
- Facilities containing AGV lines
- Clean rooms
- Manufacturing/processing flammable or combustible materials
- Munitions plants

“On Site Certification”

Specialty Applications:

- Non-skid coatings
- Chemical resistant urethanes
- Epoxy coatings
- Moisture cured urethane
- Seamless epoxy floor systems
- Waterborne systems
- VOC compliant systems
- Concrete leveling
- Concrete resurfacing
- Control joint repair

The Management at Pro-Cote Inc. wants to put its expertise to work for you. With over 55 years of combined experience, our goal is to be the best specialty coatings applicator company, period.

Application Experience

- Industrial floor coatings
- Manufacturing production areas
- Distribution and warehouse floors
- Loading docks and ramps
- Safety zones and work areas
- Battery charging areas
- UPS rooms
- Parking garages

Let us help with your specification needs.

“Safety Solutions”

Pro-Cote Inc. is a proud installer of Valspar Flooring Systems

www.valsparflooring.com

Pro-Cote Inc.
P. O. Box 1766
Largo, FL 33779
Toll Free (877) 588-4300
To be able to discern between RIP and RIPv2 and when to use each.

*Lab Design:*

![Diagram of network topology](image)

Workstation “A”        Workstation “B”
Router name: robert  morris  worm

*Step-by-Step Instructions for RIP:*
1. Set up the network shown using a 24-bit mask with a Class “C” private address and use RIP as the routing protocol. Don’t forget to advertise the routes.
2. Test ping from workstation A to workstation B.
3. Do a trace route from workstation A to workstation B.
4. On each router view and record its routing table.
5. Turn on all debug on Robert and Worm.

*Hungry for more???

You are **almost there** for the *CompTIA Network+* and *Security+* Certifications too!

For more information:

**http://www.comptia.org/certification/default.aspx**

**Building successful careers for students everywhere…**

6. Test ping from workstation A to workstation B and view the ICMP messages on Robert and Worm.
7. Change the serial lines to a 30-bit mask. (hint: the IP numbers will also need to be changed).
8. Repeat steps 2-6. About 60% of the time you will not be able to ping from workstation A to workstation B. A known quirk with RIP. Don’t sweat it if it works.
9. Switch to using RIP version 2 on all routers.
10. Repeat steps 2-6. So why do you think it works with RIPv2 and not RIP? Why or when would you use RIPv2 instead of RIP? Why or when would you use RIP instead of RIPv2?
11. Sometimes there are some bugs and quirks with the routing protocols. For example, if you use “contiguous” subnet numbers, or all in a row (for example 192.168.1.0/24, 192.168.2.0/24, 192.168.3.0/24…) the “normal” rules or theories do not apply and things that should not work, do. Crazy, I know. This is what they look for on the CCIE exams, your knowledge of the quirks and eccentricities with the IOS’s and commands. The sooner you learn to question the materials and “push” them to the limit the more you will learn. I call this playing “let’s see what happens if…” Once you have completed labs go back and re-do them, pushing them…finding their limits…how many vty lines does my router have? What are the address ranges for loopback interfaces? And that sort of question. They also make jim dandy fine test questions, especially default settings.

Guest Router Name Derivation
In 1991 Robert Morris became the first individual convicted for violating the 1986 Federal Computer Fraud and Abuse Act. He created an internet worm as part of a graduate school project whose sole purpose was to expose security vulnerabilities in networks so that network administrators could pro-actively fix any security holes. Unfortunately the project went amiss and computer networks crashed left and right when it was released errantly on the Internet. In hind-sight he should have kept it a little better under control. It just goes to show you that good intentions also get punished… “ignorance of the law is no excuse.”

Take college courses on a new schedule--yours.
Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education. Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

http://e.spcollege.edu
St. Petersburg College’s eCampus. Your time. Your place. Your future.
Objective:
To learn about the basics of the Interior Gateway Routing Protocol (IGRP) by making a small network.

Tools and Materials:
(2) PC/workstations
(2) Routers
(2) Switches
(4) Straight-through cables
(1) DCE serial cable
(1) DTE serial cable
(2) rollover cables

Lab Diagram:

Addressing:
Routers
Hostnames | John | Draper
---|---|---
E0 | 184.34.67.1/16 | 184.36.67.1/16
S0 | 184.35.67.1 (DCE) | n/a
S1 | n/a | 184.35.67.2/16 (DTE)

Workstations
A
IP | 184.34.67.3 | 184.36.67.3
SM | 255.255.0.0 | 255.255.0.0
GW | 184.34.67.1 | 184.36.67.1
B

Background:
IGRP is proprietary distance-vector routing protocol created by CISCO in the later 1980’s to overcome some of the limitations of RIP. It uses bandwidth and delay by default as its metrics. It can, however, use other metrics such as reliability, load, and MTU. IGRP uses autonomous numbers for setting up its routing protocol. An autonomous system number is used to set up many different IGRP networks within our
company and control access between them. There are three types of routes that are advertised with IGRP: internal, system and external.

Like RIP we must first enable IGRP and then advertise, publish or associate our networks with IGRP (all three things are the same…I have seen it many different ways on tests—hint-hint). IGRP shares characteristics of RIP that we saw in Part 2: it does not pass subnet mask information (geek speak: “it truncates at the classful boundary”).

**Step-By-Step Instructions:**

1. Set up and cable the lab as shown. Then set up the basics and interfaces on each router.

2. Add in IGRP as the routing protocol and advertise, publish or associate the networks like this:

```
john(config)#router igrp 38
john(config-router)#network 184.34.0.0
john(config-router)#network 184.35.0.0
```

```
draper(config)#router igrp 38
draper(config-router)#network 184.35.0.0
draper(config-router)#network 184.36.0.0
```

Notice how I picked (out of thin air) to use #38 as my autonomous system number. It really does not matter which one I use just as long as I use the same one on both sides. Notice how I advertised (published/associated) my networks at the classful boundary…a limitation of IGRP. I used “38” for my AS number…is there an upper limit to them?

3. Test by pinging from one workstation to the other. It should work just fine. Do a show ip route. You should see something like this:

```
draper#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B- BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR
Gateway of last resort is not set
I    184.34.0.0/16 [100/8576] via 184.35.67.1, 00:00:06, Serial0/1
C    184.35.0.0/16 is directly connected, Serial0/1
C    184.36.0.0/16 is directly connected, Ethernet0/0
```

```
draper#
```
Let’s test IGRP’s classful routing capability by changing the serial cable addresses to 192.168.1.1/24 and 192.168.1.2/24. Then try to ping again. Sometimes it might work, but most times it won’t work. Remember we want reliability for our networks too.

4. **Let’s test** IGRP’s classful routing capability by changing the serial cable addresses to 192.168.1.1/24 and 192.168.1.2/24. Then try to ping again. Sometimes it might work, but most times it won’t work. Remember we want reliability for our networks too.

**Supplemental Lab or Challenge Activity:**

1. Change the autonomous number on router “john” to 39. Can the two workstations still ping each other?
2. Repeat this lab using a class “A” IP addressing scheme.
3. Repeat this lab using a class “C” IP addressing scheme.
So What Did I Learn Here?
In this lab you learned about a new routing protocol called IGRP. Over the next few labs we will learn more about this protocol and several other ones too. This will help build your repertoire of routing protocols and look pretty darned cool on a resume too.

Guest Router Name Derivation

John Draper, a.k.a. “Captain Crunch,” gained notoriety in the 1970’s as a “phreaker” (phone hacker) when he figured out how pay phones work. He discovered when you put a dime in a payphone (calls in the 1970’s used to be 10 cents) the telephone had an electromechanical converter that sent a 2600-hertz tone to the phone company as a “signal” that a dime had been inserted into the telephone. About the same time he discovered that a whistle given out in boxes of Captain Crunch cereal emitted a frequency of 2600 hertz. Aha! He then could make telephone calls essentially for free. Shortly thereafter he also discovered the “Oscar Meyer Wiener” whistles also emitted a 2600-hertz frequency. Today’s pay phones still work on the same premises. The 2600-hertz frequency was also used to derive the name for “2600” magazine, better known as “The Hacker Quarterly” started by Emmanuel Goldstein in 1984.

Pro-Cote Inc.
Specialty Coatings Applicators
877-588-4300

Certified Conducive and Static Dissipative Floor applications for every need:
- Facilities with highly sensitive electronic equipment
- Computer facilities
- Electronic manufacturing & assembly
- Facilities containing AGV lines
- Clean rooms
- Manufacturing/processing flammable or combustible materials
- Munitions plants

Pro-Cote Inc. is a proud installer of Valspar Flooring Systems
www.pro-cote-usa.com

Pro-Cote Inc.
P. O. Box 1766
Largo, FL 33779
Toll Free (877) 588-4300
Basic IGRP with Protocol Inspector

Objective:
To learn how to capture and dissect IGRP packets over a simple two-router network.

Tools and Materials:
(3) PC/workstations
(2) Routers
(3) Switches
(7) Straight-through cables
(2) rollover cables

Lab Diagram:

Workstation “A”       Workstation “C”   Workstation “B”

Addressing:

Routers

<table>
<thead>
<tr>
<th>Hostnames</th>
<th>Kevin</th>
<th>Paulsen</th>
</tr>
</thead>
<tbody>
<tr>
<td>E0</td>
<td>38.12.245.1/8</td>
<td>40.12.245.1/8</td>
</tr>
<tr>
<td>E1</td>
<td>39.12.245.1</td>
<td>39.12.245.2/8</td>
</tr>
</tbody>
</table>

Workstations

<table>
<thead>
<tr>
<th>Workstations</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>38.12.245.2</td>
<td>40.12.245.2</td>
<td>39.12.245.3/8</td>
</tr>
<tr>
<td>SM</td>
<td>255.0.0.0</td>
<td>255.0.0.0</td>
<td>255.0.0.0</td>
</tr>
<tr>
<td>GW 1</td>
<td>38.12.245.1</td>
<td>40.12.245.1</td>
<td>39.12.245.1</td>
</tr>
<tr>
<td>GW 2</td>
<td>n/a</td>
<td>n/a</td>
<td>39.12.245.2</td>
</tr>
</tbody>
</table>

Background:
One of the disadvantages of using the Ethereal protocol inspector is that it will only capture packets on the subnet to which it is attached. In order to grab those IGRP packets we must set up a network that will allow us to do so. In the last lab we used a serial line between the two routers. Let’s change that to an Ethernet line (as well as using dual Ethernet routers) and try to capture IGRP packets with our Ethereal.
Step-By-Step Instructions:
1. Set up and cable the lab as shown. Notice how we need two gateway addresses on workstation C. Since the packets can travel either way we need to account for both gateways.
2. Test ping from each workstation to each other. This should be just fine and jimdandy.
3. Do a show ip route. It should look like this:

kevin#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR
Gateway of last resort is not set
C 38.0.0.0/8 is directly connected, Ethernet0/0
C 39.0.0.0/8 is directly connected, Ethernet0/1
I 40.0.0.0/8 [100/1200] via 39.12.245.2, 00:01:15, Ethernet0/1
kevin#

4. Now start Ethereal on workstations B and C. Let it run for 2-3 minutes. Then stop and analyze it. On workstation C you should see something like this:

You will have to expand the tree [+] buttons to see all of this information. Notice how we can see the metrics and their values here. Hmm…looks good.
5. Then open up one for workstation B. When Ethereal first comes up everything is sequentially ordered by time. Let’s change to ordering by “protocol.” Just click on the protocol button near the headers to sort them alphabetically by protocol from A to Z. Clicking “protocol” again will sort them descending from Z to A. Notice here how we have two entry routes into the network.

**Supplemental Lab or Challenge Activity:**

1. Try using workstation C without the second gateway. What happens when you try to ping both A and B?
2. Since Ethereal is showing us our IGRP responses then where are the requests (queries)?
3. We also see that we are using IGRP version 1…is there an IGRP version 2? We know RIP has a version 2.
4. Why do we have two entries into our network on workstation B?
5. What would you expect to see on workstation A?
6. How could we force a IGRP routing update?
7. Repeat this lab using a class “B” IP addressing scheme.
8. Repeat this lab using a class “C” IP addressing scheme.

**So What Did I Learn Here?**

In this lab you learned how to capture IGRP packets with Ethereal. In our next lab you will expand upon this by changing the metrics over our Ethernet lines and “watch” the routing in action. Having fun yet? This stuff is just so much fun!
Guest Router Name Derivation

In 1990 Kevin Paulsen, a.k.a. “dark dante,” used his knowledge of the phone company and their operations to seize control of all telephone lines into KIIS-FM in Los Angeles. Then it was easy for him to be the 102nd caller and win the shiny Porche. He also has been photographed picking locks to phone company property and admitted to hacking into the FBI to obtain lists of companies that are owned and operated by the FBI.

Looking for college courses?
We deliver.

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education.

Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

St. Petersburg College’s eCampus
http://e.spcollege.edu

Your time. Your place. Your future.
Intermediate IGRP: Metrics

Objective:
To learn about the metrics used with IGRP.

Tools and Materials:
(4) PC/workstations
(4) Routers
(4) Switches
(7) Straight-through cables
(2) rollover cables

Lab Diagram:

Addressing:

Routers

<table>
<thead>
<tr>
<th>Hostnames</th>
<th>E0</th>
<th>E1</th>
<th>S0 (DCE)</th>
<th>S1</th>
</tr>
</thead>
<tbody>
<tr>
<td>dennis</td>
<td>200.150.100.1/24</td>
<td>n/a</td>
<td>201.150.100.1/24</td>
<td>n/a</td>
</tr>
<tr>
<td>ritchie</td>
<td>202.150.100.1/24</td>
<td>203.150.100.1/24</td>
<td>n/a</td>
<td>201.150.100.2/24</td>
</tr>
</tbody>
</table>

Routers

<table>
<thead>
<tr>
<th>Hostnames</th>
<th>E0</th>
<th>E1</th>
<th>S0</th>
<th>S1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ken</td>
<td>202.150.100.2/24</td>
<td>203.150.100.2/24</td>
<td>n/a</td>
<td>201.150.101.2/24</td>
</tr>
<tr>
<td>Thompson</td>
<td>200.150.101.1/24</td>
<td>n/a</td>
<td>201.150.101.1/24</td>
<td>n/a</td>
</tr>
</tbody>
</table>
### Workstations A and B

<table>
<thead>
<tr>
<th></th>
<th>IP</th>
<th>SM</th>
<th>GW 1</th>
<th>GW 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workstations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>200.150.100.2</td>
<td>255.255.255.0</td>
<td>200.150.101.1</td>
<td>n/a</td>
</tr>
<tr>
<td>B</td>
<td>200.150.101.2</td>
<td>255.255.255.0</td>
<td>200.150.101.1</td>
<td>n/a</td>
</tr>
</tbody>
</table>

### Workstations C and D

<table>
<thead>
<tr>
<th></th>
<th>IP</th>
<th>SM</th>
<th>GW 1</th>
<th>GW 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workstations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>202.150.100.3</td>
<td>255.255.255.0</td>
<td>202.150.100.1</td>
<td>202.150.100.2</td>
</tr>
<tr>
<td>D</td>
<td>203.150.100.3</td>
<td>255.255.255.0</td>
<td>203.150.100.1</td>
<td>203.150.100.2</td>
</tr>
</tbody>
</table>

**Background:**

In part 2 you learned that RIP uses “Hop Count” as its routing metric. IGRP uses “bandwidth” (BW) and “delay” (DLY), by default as its routing metrics. Unlike RIP, IGRP has other metrics that can be used for its routing process. Those other metrics include maximum transmission unit (MTU), reliability (RLY), and load. In this lab you will learn how to manipulate these metrics to suit your network needs. You will be “statically” configuring load balancing by changing the metrics to make one of two routes more desirable than the other. Finally you will learn how to set up “dynamic” load balancing so each route gets a nearly equal amount of the work.

### Step-By-Step Instructions:

1. Set up and cable the lab as shown. Give yourself enough time to do this. Don’t rush through it otherwise your typos will cause headaches.
2. Test ping from each workstation to each other. This should be just fine.
3. Do a show ip route on each router. They should look like this:

    dennis#sh ip route

    Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
    D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
    U - per-user static route, o - ODR

    Gateway of last resort is not set

    I  202.150.100.0/24 [100/8576] via 201.150.100.2, 00:00:19, Serial0/0
    I  203.150.100.0/24 [100/8576] via 201.150.100.2, 00:00:19, Serial0/0
    I  201.150.101.0/24 [100/10576] via 201.150.101.1, 00:00:19, S0/0
    C  200.150.100.0/24 is directly connected, Ethernet0/0
    C  201.150.100.0/24 is directly connected, Serial0/0
    I  200.150.101.0/24 [100/10676] via 201.150.101.1, 00:00:19, S0/0
    dennis#
ritchie#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
U - per-user static route, o - ODR

Gateway of last resort is not set
C 202.150.100.0/24 is directly connected, Ethernet0/0
C 203.150.100.0/24 is directly connected, Ethernet0/1
I 201.150.101.0/24 [100/8576] via 202.150.100.2, 00:00:58, E0/0
    [100/8576] via 203.150.100.2, 00:00:58, E0/1
I 200.150.100.0/24 [100/8576] via 201.150.100.1, 00:00:27, Serial0/1
C 201.150.100.0/24 is directly connected, Serial0/1
I 200.150.101.0/24 [100/8676] via 202.150.100.1, 00:00:58, E0/0
    [100/8676] via 203.150.100.1, 00:00:58, E0/1

ken#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
U - per-user static route, o - ODR

Gateway of last resort is not set
C 202.150.100.0/24 is directly connected, Ethernet0/0
C 203.150.100.0/24 is directly connected, Ethernet0/1
C 201.150.100.0/24 is directly connected, Serial0/1
I 200.150.100.0/24 [100/8676] via 202.150.100.1, 00:00:41, E0/0
    [100/8676] via 203.150.100.1, 00:00:41, E0/1
I 201.150.100.0/24 [100/8576] via 202.150.100.1, 00:00:41, E0/0
    [100/8576] via 203.150.100.1, 00:00:41, E0/1
I 200.150.101.0/24 [100/8576] via 201.150.101.1, 00:00:04, Serial0/1

thompson#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
U - per-user static route, o - ODR

Gateway of last resort is not set
I 202.150.100.0/24 [100/8576] via 201.150.101.2, 00:00:08, Serial0/0
I 203.150.100.0/24 [100/8576] via 201.150.101.2, 00:00:08, Serial0/0
C 201.150.101.0/24 is directly connected, Serial0/0
I 200.150.100.0/24 [100/10676] via 201.150.101.2, 00:00:08, Serial0/0
I 201.150.100.0/24 [100/10576] via 201.150.101.2, 00:00:08, Serial0/0
C 200.150.101.0/24 is directly connected, Ethernet0/0
4. Now start Ethereal on workstations C or D. Let it run for 2-3 minutes. Then stop and analyze it. On workstation C or D you should see something like this:

You will have to expand the tree [+] buttons to see all of this information. Notice how we can see the metrics and their values here. Hmm...looks good. You can see our default metrics with IGRP are: delay set to 100, bandwidth set to 1000, MTU of 1500 bytes, reliability set to 255, and load set to 1. Hop count here is not a metric, per se, but a device to measure how far it is from here to the “entry point for the network.” Another way to view our default metrics is with the show interface command. Here is an example of the first five lines of output:

```
ritchie# sh int ethernet0/0
Ethernet0/0 is up, line protocol is up
   Hardware is AmdP2, address is 0002.fd45.ac80 (bia 0002.fd45.ac80)
   Internet address is 202.150.100.1/24
   MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
   Encapsulation ARPA, loopback not set, keepalive set (10 sec)
   ARP type: ARPA, ARP Timeout 04:00:00
```

5. Now let’s try to see how workstation “A” is routed to workstation “B” by using trace route from the DOS prompt:

```
C:\WINDOWS\Desktop> tracert 200.150.101.2
Tracing route to STAR10616125 [200.150.101.2] over a maximum of 30 hops:
    1    2 ms    1 ms    1 ms  200.150.100.1
```
The “crucial” step in our trace is in bold above. We can see the path is through the lower Ethernet path in our diagram. We can actually statically configure the Ethernet 1 interface (on ritchie) to pass the packets through Ethernet 1 by lowering (or raising) the specific metrics to make the 203.x.x.x route more desirable. Likewise we could also raise (or raise) the metrics on Ethernet 0 to make it less desirable.

6. Let’s start by making the 203.x.x.x more desirable by increasing the delay on Ethernet 0 from 1000 to 10000. Since there is a longer delay on Ethernet 0 (which we statically set) then the 203.x.x.x network would become the preferred route (with all other metrics being equal).

```
ritchie(config)#int e0/0
ritchie(config-if)#delay 10000
```

7. Now we can repeat our trace and see if it works the way we want (to force the path over the 203.x.x.x network):

```
C:\WINDOWS\Desktop>tracert 200.150.101.2
Tracing route to STAR10616125 [200.150.101.2] over a maximum of 30 hops:
  1     1 ms     1 ms     1 ms  200.150.100.1
  2    68 ms    25 ms    25 ms  201.150.100.2
  3    25 ms    26 ms    26 ms  203.150.100.2
  4    49 ms    49 ms    49 ms  201.150.101.1
  5    59 ms    59 ms    59 ms  STAR10616125
[200.150.101.2]
Trace complete.
```

8. Take college courses on a new schedule--yours.

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education. Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

```
http://e.spcollege.edu
```

St. Petersburg College’s eCampus.  Your time. Your place. Your future.
Bingo! Just what we had hoped for…Let’s check this with Ethereal.

Here we can see the entry for network 202.x.x.x now has a delay of 10000. Don’t you just love it when things work nicely?

From the default settings you can decrease the delay from 1000 down to 500 on the Ethernet 1 interface and get the same effect. To force the trace from A to B on ritchie to always use the 203 route (it will always use the 202 route with default settings on each:

<table>
<thead>
<tr>
<th>change</th>
<th>from</th>
<th>to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth</td>
<td>1000</td>
<td>10000</td>
</tr>
<tr>
<td>OR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay</td>
<td>1000</td>
<td>10000</td>
</tr>
<tr>
<td>OR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTU</td>
<td>1500</td>
<td>50</td>
</tr>
<tr>
<td>OR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RLY</td>
<td>255</td>
<td>255**</td>
</tr>
<tr>
<td>OR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load</td>
<td>1</td>
<td>255**</td>
</tr>
<tr>
<td>OR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* You wouldn’t want to go higher than 1500 if you are using Ethernet (max. size of 1518)
** Minimum/Maximum size is already set.
Supplemental Lab or Challenge Activity:
1. What is the "Variance" command and how does it relate to IGRP?
2. Repeat this lab using a class “B” IP addressing scheme.
3. Repeat this lab using a class “C” IP addressing scheme.

So What Did I Learn Here?
In this lab you have learned how to statically and dynamically manipulate your metrics to achieve traffic flow in the manner you desire. Watch out for the “variance” command when you are studying for your test. This is a good “one-line wonder” question—the information only appears once, but you are still expected to know it anyway. In the next lab you will learn more about that autonomous number thing-a-ma-jigie. Don’t erase your configurations…we will use the same one for the next lab.

Guest Router Name Derivation
In 1969 Dennis Ritchie and Ken Thompson invented the UNIX operating System. If they only knew then what they were doing…creating software that would help put a man on the moon, transmit pictures back from Mars, and the solar system…oh, yeah…and give a green light to hackers everywhere. Nobody said anything was perfect.

The “Script Kiddie Cookbook” Available Mid-September 2004 at http://www.lulu.com/learningbydoing

This book title is based upon the Anarchist Cookbook that came out in the 1980’s. In that tremendously popular book it showed you how to build bombs at home. Similarly the Script Kiddie Cookbook will show you step by step hacking and hacking methodology. You will go to jail if you use this information improperly!!!

Learn the basics about computer security with this book. This book assumes you know nothing about security and starts there. All tools and software used are freeware. Other books will build upon the materials in this book. Learn about the nuts and bolts of SPAM, port scanning, Knoppix tools, cookies, and bring it all together with exercises in hacking at legal gaming sites.

GET THE BOOK THE GOVERNMENT DOES NOT WANT YOU TO OWN!!!
Objective:
To learn how to redistribute IGRP networks with IGRP networks and IGRP networks with RIP networks.

Tools and Materials:
(4) PC/workstations
(4) Routers
(4) Switches
(7) Straight-through cables
(2) rollover cables

Addressing:
Routers

<table>
<thead>
<tr>
<th>Hostnames</th>
<th>E0 IP Address /24</th>
<th>E1 IP Address /24</th>
<th>S0 IP Address /24</th>
<th>S1 IP Address /24</th>
</tr>
</thead>
<tbody>
<tr>
<td>dennis</td>
<td>200.150.100.1/24</td>
<td>n/a</td>
<td>201.150.100.1/24</td>
<td>n/a</td>
</tr>
<tr>
<td>ritchie</td>
<td>202.150.100.1/24</td>
<td>203.150.100.1/24</td>
<td>n/a</td>
<td>201.150.100.2/24</td>
</tr>
</tbody>
</table>

Routers

<table>
<thead>
<tr>
<th>Hostnames</th>
<th>E0 IP Address /24</th>
<th>E1 IP Address /24</th>
<th>S0 IP Address /24</th>
<th>S1 IP Address /24</th>
</tr>
</thead>
<tbody>
<tr>
<td>ken</td>
<td>202.150.100.2/24</td>
<td>203.150.100.2/24</td>
<td>n/a</td>
<td>201.150.101.2/24</td>
</tr>
<tr>
<td>Thompson</td>
<td>200.150.101.1/24</td>
<td>n/a</td>
<td>201.150.101.1/24</td>
<td>n/a</td>
</tr>
</tbody>
</table>
Workstations A   B
   IP   200.150.100.2  200.150.101.2
   SM  255.255.255.0  255.255.255.0
   GW 1  200.150.100.1  200.150.101.1
   GW 2 n/a         n/a

Workstations C   D
   IP   202.150.100.3  203.150.100.3
   SM  255.255.255.0  255.255.255.0
   GW 1  202.150.100.1  203.150.100.1
   GW 2 202.150.100.2  203.150.100.2

Background:
Picture this…your company is running IGRP with an autonomous system number of 38.
You have 17 routers in your network spread out over 4 states. Your company buys out
another company with IGRP and an autonomous system number of 18 and 15 routers
spread out over 2 other states. It would literally take you several days to convert the new
network over to work with your network but your boss wants it up and running yesterday.
No problem. You can redistribute those other autonomous system numbers into your
own on only the “border router” with several simple commands. You can be done in
minutes! In this lab you will learn how to redistribute IGRP with IGRP and IGRP with
RIP.

Step-By-Step Instructions:
1. Since the last lab was so extensive to set up and this lab only modifies it a bit I
   thought I would save you some time.
2. Now let’s set up a “brief version” of the scenario above:

   ritchie(config)#router igrp 38
   ritchie(config-router)#no network 201.150.100.0
   ritchie(config-router)#redistribute igrp 18

   ritchie(config-router)#router igrp 18
   ritchie(config-router)#network 201.150.100.0
   ritchie(config-router)#redistribute igrp 38

   dennis(config)#no router igrp 38
   dennis(config)#router igrp 18
   dennis(config-router)#network 201.150.100.0
   dennis(config-router)#network 200.150.100.0
3. Now we can see how this affects our ip routes. On each router you will see:

dennis#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
Gateway of last resort is not set
I 202.150.100.0/24 [100/8576] via 201.150.100.2, 00:00:29, Serial0/0
I 203.150.100.0/24 [100/8576] via 201.150.100.2, 00:00:30, Serial0/0
I 201.150.101.0/24 [100/10576] via 201.150.100.2, 00:00:30, S0/0
C 200.150.100.0/24 is directly connected, Ethernet0/0
C 201.150.100.0/24 is directly connected, Serial0/0
I 200.150.101.0/24 [100/10676] via 201.150.100.2, 00:00:30, S0/0

ritchie#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
Gateway of last resort is not set
C 202.150.100.0/24 is directly connected, Ethernet0/0
C 203.150.100.0/24 is directly connected, Ethernet0/1
I 201.150.101.0/24 [100/8576] via 203.150.100.2, 00:00:20, E0/1
[100/8576] via 202.150.100.2, 00:00:20, E0/0
I 200.150.100.0/24 [100/8576] via 201.150.100.1, 00:00:40, Serial0/1
C 201.150.100.0/24 is directly connected, Serial0/1
I 200.150.101.0/24 [100/8676] via 203.150.100.2, 00:00:20, E0/1
[100/8676] via 202.150.100.2, 00:00:20, E0/0

ken#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
Gateway of last resort is not set
C 202.150.100.0/24 is directly connected, Ethernet0/0
C 203.150.100.0/24 is directly connected, Ethernet0/1
C 201.150.101.0/24 is directly connected, Serial0/1
I 200.150.100.0/24 [100/8676] via 202.150.100.1, 00:00:44, E0/0
[100/8676] via 203.150.100.1, 00:00:44, E0/1
I 201.150.100.0/24 [100/8576] via 202.150.100.1, 00:00:44, E0/0
[100/8576] via 203.150.100.1, 00:00:44, E0/1
I 200.150.101.0/24 [100/8576] via 201.150.101.1, 00:00:25, Serial0/1
thompson#sh ip route

Codes:  C - connected,  S - static,  I - IGRP,  R - RIP,  M - mobile,  B - BGP
       D - EIGRP,  EX - EIGRP external,  O - OSPF,  IA - OSPF inter area
       N1 - OSPF NSSA external type 1,  N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1,  E2 - OSPF external type 2,  E - EGP
       i - IS-IS,  L1 - IS-IS level-1,  L2 - IS-IS level-2,  * - candidate default
       U - per-user static route,  o - ODR

Gateway of last resort is not set

I    202.150.100.0/24 [100/8576] via 201.150.101.2, 00:00:38, Serial0/0
I    203.150.100.0/24 [100/8576] via 201.150.101.2, 00:00:38, Serial0/0
C    201.150.101.0/24 is directly connected, Serial0/0
I    200.150.100.0/24 [100/10676] via 201.150.101.2, 00:00:38, S0/0
I    201.150.100.0/24 [100/10576] via 201.150.101.2, 00:00:38, S0/0
C    200.150.101.0/24 is directly connected, Ethernet0/0

4. Now let’s change our igrp 18 network over to a RIP network. First let’s get rid of the igrp 18 information:

ritchie(config)#router igrp 38
ritchie(config-router)#no redistribute igrp 18
ritchie(config)#no router igrp 18

dennis(config)#no router igrp 18

5. Now let’s change over to RIP and redistribute it in our network with IGRP:

ritchie(config)#router igrp 38
ritchie(config-router)#redistribute rip 1

ritchie(config-router)#router rip
ritchie(config-router)#network 201.150.100.0
ritchie(config-router)#redistribute igrp 38

dennis(config)#router rip
dennis(config-router)#network 201.150.100.0
dennis(config-router)#network 200.150.100.0

You should be able to ping from router to router without too much problem. (Different IOS’s may have to use additional commands) However, from workstation A to B will not work because the Time To Live will be exceeded. This is a known problem when redistributing RIP into IGRP where the potential for a routing loop exists. For now just disconnect the straight through cables on Ethernet 0 on both Ritchie and ken. This will eliminate the routing loop problem. Relax. Remember RIP takes a while to converge so you might not see the routes or be able to ping for a few minutes. Also, clearing the ip routes a few times couldn’t hurt either:

dennis#clear ip route *
dennis#clear ip route *
dennis#clear ip route *
ritchie#clear ip route *
ritchie#clear ip route *
ritchie#clear ip route *
ritchie#clear ip route *

ken#clear ip route *
ken#clear ip route *
ken#clear ip route *
ken#clear ip route *

thompson#clear ip route *
thompson#clear ip route *
thompson#clear ip route *

6. Once we have done this then now we can see how this affects our ip routes. On each router you will see:

**dennis#sh ip route**

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
U - per-user static route, o - ODR

Gateway of last resort is not set

- **R 203.150.100.0/24 [120/1] via 201.150.100.2, 00:00:11, Serial0/0**
- **R 201.150.101.0/24 [120/1] via 201.150.100.2, 00:00:12, Serial0/0**
- **C 200.150.100.0/24 is directly connected, Ethernet0/0**
- **C 201.150.100.0/24 is directly connected, Serial0/1**
- **R 200.150.101.0/24 [120/1] via 201.150.100.2, 00:00:12, Serial0/0**

**ritchie#sh ip route**

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
U - per-user static route, o - ODR

Gateway of last resort is not set

- **C 203.150.100.0/24 is directly connected, Ethernet0/1**
- **I 201.150.101.0/24 [100/8576] via 203.150.100.2, 00:01:00, E0/1**
- **R 200.150.100.0/24 [120/1] via 201.150.100.1, 00:00:22, Serial0/1**
- **C 201.150.100.0/24 is directly connected, Serial0/1**
- **I 200.150.101.0/24 [100/8676] via 203.150.100.2, 00:01:00, E0/1**
Supplemental Lab or Challenge Activity:
1. When redistributing IGRP with IGRP what happens if you only redistribute on one side (redistribute igrp 38 within 18 but not redistributing igrp 18 within 38)?
2. Repeat this lab with a 26 bit subnet mask. Why does it or doesn’t it work very well now?

So What Did I Learn Here?
In this lab you started to learn the basics about redistribution with routing protocols. Sorry to tell you this is just the tip of the iceberg. Very few networks use the exact same routing protocol throughout the entire network (more likely in large networks). In fact later when you redistribute other protocols you will also have to put metrics in as well. Whew! RIP…done. IGRP…done. There are three other routing protocols we need to discuss in the next few labs: EIGRP, OSPF, and BGP. These three are covered in-depth in the upper-level CISCO courses but you should be aware of the basics regarding these protocols and for what they are used.
Guest Router Name Derivation

In 1969 Dennis Ritchie and Ken Thompson invented the UNIX operating System. If they only knew then what they were doing…creating software that would help put a man on the moon, transmit pictures back from Mars, and the solar system…oh, yeah…and give a green light to hackers everywhere. Nobody said anything was perfect.
Enhanced IGRP

Objective:
To learn the basics about the EIGRP routing protocol and how to configure EIGRP in a small network.

Tools and Materials:
(2) PC/workstations
(2) Routers
(2) Switches
(4) Straight-through cables
(1) DCE serial cable
(1) DTE serial cable
(2) rollover cables

Lab Diagram:

Addressing:
Routers
Hostnames    war games
E0    138.74.16.1/20 220.34.98.17/28
S0    14.32.0.1/12 (DCE) n/a
S1    n/a    14.32.0.2/12 (DTE)

Workstations
A    B
IP    138.74.16.2 220.34.98.18
SM    255.255.240.0 255.255.255.240
GW    138.74.16.1 220.34.98.17

Background:
The Enhanced Interior Gateway Routing Protocol (EIGRP) is a proprietary hybrid (distance vector) routing protocol developed by CISCO to exceed the capabilities of IGRP. In a nutshell EIGRP is similar to IGRP except that its metrics are 256 times that of IGRP (sounds like a good test question). In fact, in most cases EIGRP and IGRP are interchangeable. We just talked about redistribution of IGRP and RIP. There is no need
to add the extra metrics statements like with did with those (except for certain IOS versions). EIGRP and IGRP can be redistributed without those extra metric statements. How easy is that? Unlike IGRP, EIGRP supports Variable Length Subnet Masking (VLSM) so we do not have to be so concerned about the classful boundaries like we had to with IGRP (and RIP too). Instead of sending updates every 30 seconds like RIP and IGRP EIGRP sends out periodic “hello… I am still here” packets and will only send the entire routing table when a change is made. This helps to reduce the overhead traffic—another perk with EIGRP.

**Step-By-Step Instructions:**

1. Cable the lab as shown and configure the interfaces.
2. Enable EIGRP as a routing protocol and advertise/publish/associate your networks. Like IGRP EIGRP requires an autonomous system number too:

   ```
   war(config)# router eigrp 88
   war(config-router)# network 138.74.16.0
   war(config-router)# network 14.32.0.0
   games(config)# router eigrp 88
   games(config-router)# network 14.32.0.0
   games(config-router)# network 220.34.98.0
   ```

One cool thing about EIGRP is you can really “force” things to work better. Everything under the sun seems to need a mask of some sort so why not the routing protocol advertisement? EIGRP lets you do this. For example we could have done the above statements this way too:

```
war(config)# router eigrp 88
war(config-router)# network 138.74.16.0 255.255.240.0
war(config-router)# network 14.32.0.0 255.240.0.0

games(config)# router eigrp 88
games(config-router)# network 14.32.0.0 255.240.0.0
games(config-router)# network 220.34.98.0 255.255.240.0
```

3. Try to ping from A to B. It should work just fine.
4. Start Ethereal on workstation A. After about 30 packets disconnect the serial line, wait a few seconds, and then plug it back in. Remember EIGRP will only send the tables when a change occurs, otherwise it just sends “hello” packets. We should now see both:
Do you see anything unusual here? How about our destination address of 224.0.0.10? (you cannot see it on mine but you can see it on yours.) How about those metrics? Yeah…I know. Something to look up. You can also see the autonomous system number too.

5. So how come you do not see any ‘updates” from when our line went down? Remember we have to be on the subnet too. Our workstations do not receive the update broadcasts. We can fudge it a bit by adding another router into our switch. Then we should be able to see the changes.

New Lab Diagram:

Workstation “A”  
New Router: “WOPR”  
E0/0  138.74.16.20/20  
L0  1.1.1.1/8  

Workstation “B”
6. Don’t forget to update your route advertisements with EIGRP. Now we should be able to see those changes when we take down the serial line:

7. Notice the reachable/not reachable routes and how our metrics changed from those “K” numbers to those like IGRP metrics. Neat!

8. Let’s compare the protocol inspector output put to a debug eigrp packets:

```plaintext
wopr#debug eigrp packets
EIGRP Packets debugging is on
  (UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK)
00:08:10: EIGRP: Sending HELLO on Ethernet0/0
00:08:10:   AS 88, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rELY 0/0
00:08:10: EIGRP: Received HELLO on Ethernet0/0 nbr 138.74.16.1
00:08:10:   AS 88, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rELY 0/0 peerQ un/rELY 0/0
00:08:10: EIGRP: Sending HELLO on Loopback0
00:08:10:   AS 88, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rELY 0/0
00:08:10: EIGRP: Received HELLO on Loopback0 nbr 1.1.1.1
00:08:10:   AS 88, Flags 0x0, Seq 0/0 idbQ 0/0
00:08:10: EIGRP: Packet from ourselves ignored
00:08:17: EIGRP: Received QUERY on Ethernet0/0 nbr 138.74.16.1
00:08:17:   AS 88, Flags 0x0, Seq 16/0 idbQ 0/0 iidbQ un/rELY 0/0 peerQ un/rELY 0/0
00:08:17: EIGRP: Enqueueing ACK on Ethernet0/0 nbr 138.74.16.1
00:08:17:   Ack seq 16 iidbQ un/rELY 0/0 peerQ un/rELY 1/0
00:08:17: EIGRP: Sending ACK on Ethernet0/0 nbr 138.74.16.1
00:08:17: EIGRP: Received ACK on Ethernet0/0 nbr 138.74.16.1
00:08:17:   AS 88, Flags 0x0, Seq 0/16 idbQ 0/0 iidbQ un/rELY 0/0 peerQ un/rELY 1/0
00:08:17: EIGRP: Enqueueing REPLY on Ethernet0/0 nbr 138.74.16.1 iidbQ un/rELY 0
```
9. Let’s reconnect it and look at our ip routes:

wopr#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
       U - per-user static route, o - ODR

Gateway of last resort is not set
C  1.0.0.0/8 is directly connected, Loopback0
D  220.34.98.0/24 [90/2221056] via 138.74.16.1, 00:00:59, Ethernet0/0
D  14.0.0.0/8 [90/2195456] via 138.74.16.1, 00:01:04, Ethernet0/0
C  138.64.0.0/12 is directly connected, Ethernet0/0

war#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
       U - per-user static route, o - ODR

Gateway of last resort is not set
D  1.0.0.0/8 [90/409600] via 138.74.16.3, 00:05:46, Ethernet0/0
138.74.0.0/16 is variably subnetted, 2 subnets, 2 masks
 138.74.0.0/16 is a summary, 00:23:55, Null0
D  138.74.16.0/20 is directly connected, Ethernet0/0
D  220.34.98.0/24 [90/2195456] via 14.32.0.2, 00:01:10, Serial0/0
14.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D  14.0.0.0/8 is a summary, 00:01:15, Null0
C  14.32.0.0/12 is directly connected, Serial0/0

games#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP

412
Notice how our EIGRP routes are noted with a “D” not an “E.”

**Supplemental Lab or Challenge Activity:**

1. How would you redistribute IGRP and EIGRP? RIP and EIGRP?
2. Go out to CISCO and look up EIGRP on their technical documentation site. What is DUAL and RTP?
3. How often are “hello” packets sent?

**So What Did I Learn Here?**

You learned about the hybrid CISCO-proprietary routing protocol EIGRP.

---

**Guest Router Name**

War Games is the first “great” hacker movie from 1984 starring Matthew Broderick, Alley Sheedy, and Dabney Coleman. In a roundabout way it created “idols” for young disenchanted computer geeks to become hackers. In the movie Matthew Broderick “hacked” into a military computer called “WOPR.” Most of the little geeks (me included) got the message loud and clear: if you become a hacker you get to date a very pretty girl and visit exotic locations without the permission of your parents. They got the “exotic locations” right…most ended up in jail. Not too many “girls” in there (the men’s facilities). I will leave it to your imagination though.

---

Keep track of updates and changes at [http://www.spcollege.edu/star/cisco](http://www.spcollege.edu/star/cisco) Scroll to the bottom of the page and click on the “Lab Manual Edits.”
Open Shortest Path First (OSPF)

Objective:
To learn how to configure a very basic OSPF network with two routers and to learn about wildcard masks.

Tools and Materials:
(2) PC/workstations
(2) Routers  (OSPF is a memory hog…the more memory the better here)
(2) Switches
(4) Straight-through cables
(1) DCE/DTE serial cable
(2) rollover cables

Lab Diagram:

Addressing:

Routers

Hostnames    | wash | leung
-------------|------|------
E0           | 172.16.1.1/24 | 172.16.3.1/24
S0           | 172.16.2.1/24 (DCE) | n/a
S1           | n/a | 172.16.2.2/24 (DTE)

Workstations

A

IP          | 172.16.1.2 | 172.16.3.2/24
SM          | 255.255.255.0 | 255.255.255.0
GW          | 172.16.1.1 | 172.16.3.1

B

Background:
OSPF was developed in the late 1980’s as an alternative to the distance vector routing protocols (RIP, IGRP, etc). OSPF is link-state protocol that uses the Dijstra’s algorithm (Shortest Path First-SPF). OSPF does what it sounds like: it calculates the shortest route to a destination, but not necessarily the quickest one. Unlike IGRP and EIGRP the OSPF protocol is not proprietary to CISCO equipment. Unlike IGRP and RIP (version 1) OSPF can accommodate passing various lengths of subnets with data information (VLSM and/or CIDR). OSPF on a wider scale is better left to upper-level courses. You are only getting a brief overview here.
Quick overview: Wildcard Masks

A while back you learned about subnet masks. We use wildcard masks to instruct our devices to “only pay attention” to certain information. The easiest way I know to explain how to set up a wildcard mask is: a wildcard mask is usually the exact opposite of a subnet mask (in terms of binary one’s and zero’s). One last note: a wildcard mask, unlike a subnet mask, does not have to contain contiguous one’s...more on this later). Let’s look at an example:

If we had a network 172.16.1.0/24 and wanted to use a routing protocol:

- With RIP, IGRP, EIGRP, BGP (with subnet mask):
  - network 172.16.1.0 255.255.255.0
  - let’s see that subnet mask in binary:
    - 11111111.11111111.11111111.00000000

- With OSPF (with wildcard mask):
  - network 172.16.1.0 0.0.0.255
  - let’s see that wildcard mask in binary:
    - 00000000.00000000.00000000.11111111

Step-By-Step Instructions:

1. Set up and cable the lab as shown. Do not use any routing protocol. Notice how our addresses extend beyond our address class boundary. OSPF will pass subnet information.

2. Now let’s add in our OSPF routing protocol. We use the number 0 because OSPF requires at least one “area” be numbered 0. Yes...the number “1” is an autonomous system number too.

   wash(config)#router ospf 1
   wash(config-router)#network 172.16.1.0 0.0.0.255 area 0
   wash(config-router)#network 172.16.2.0 0.0.0.255 area 0

   leung(config)#router ospf 1
   leung(config-router)#network 172.16.2.0 0.0.0.255 area 0
   leung(config-router)#network 172.16.3.0 0.0.0.255 area 0

3. We can use some show commands too:

   wash#sh ip route
   Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
   D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
   N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
   E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
   i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
   default
   U - per-user static route, o - ODR
Gateway of last resort is not set

    172.16.0.0/24 is subnetted, 3 subnets
C       172.16.1.0 is directly connected, Ethernet0/0
C       172.16.2.0 is directly connected, Serial0/0
O       172.16.3.0 [110/74] via 172.16.2.2, 00:01:25, Serial0/0

leung#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
U - per-user static route, o - ODR

Gateway of last resort is not set

    172.16.0.0/24 is subnetted, 3 subnets
O       172.16.1.0 [110/74] via 172.16.2.1, 00:01:29, Serial0/1
C       172.16.2.0 is directly connected, Serial0/1
C       172.16.3.0 is directly connected, Ethernet0/0

wash#sh ip ospf
Routing Process "ospf 1" with ID 172.16.2.1
Supports only single TOS(TOS0) routes
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x0
Number of DCbitless external LSA 0
Number of DoNotAge external LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0
nssa
Area BACKBONE(0)
    Number of interfaces in this area is 2
    Area has no authentication
    SPF algorithm executed 3 times
    Area ranges are
    Number of LSA 2. Checksum Sum 0x848F
    Number of DCbitless LSA 0
    Number of indication LSA 0
    Number of DoNotAge LSA 0

leung#sh ip ospf
Routing Process "ospf 1" with ID 172.16.3.1
Supports only single TOS(TOS0) routes
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x0
Number of DCbitless external LSA 0
Number of DoNotAge external LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0
nssa
    Area BACKBONE(0)
Number of interfaces in this area is 2
Area has no authentication
SPF algorithm executed 2 times
Area ranges are
Number of LSA 2. Checksum Sum 0x848F
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0

```
wash#sh ip ospf neighbor
Neighbor ID     Pri  State           Dead Time   Address     Interface
172.16.3.1        1   FULL/  -        00:00:32    172.16.2.2  Serial0/0
```

```
leung#sh ip ospf neighbor
Neighbor ID     Pri  State           Dead Time   Address     Interface
172.16.2.1        1   FULL/  -        00:00:31    172.16.2.1  Serial0/1
```

```
leung#debug ip ospf events
OSPF events debugging is on
00:10:09: OSPF: Rcv hello from 172.16.2.1 area 0 from Serial0/1 172.16.2.1
00:10:09: OSPF: End of hello processing
00:10:19: OSPF: Rcv hello from 172.16.2.1 area 0 from Serial0/1 172.16.2.1
00:10:19: OSPF: End of hello processing
00:10:29: OSPF: Rcv hello from 172.16.2.1 area 0 from Serial0/1 172.16.2.1
00:10:29: OSPF: End of hello processing
```

Supplemental Lab or Challenge Activities:
1. Go out to CISCO and find out how Designated Routers and Backup Designated Routers are elected.
2. Find out why we use loopback address with OSPF.
3. Capture and analyze the OSPF packet structure.
4. What is a “hello” packet in OSPF?

So What Did I Just Learn Here?
In this lab you learned the basics of the OSPF routing protocol. Trust me...there is a lot more to this routing protocol. You also learned about the basics of wildcard masks. We will be using these in a couple more labs on Access Control Lists so now was a good time to bring this up.
Washington Leung was sentenced in early 2002 to 18 months in Federal prison and $92,000 in restitution for illegally accessing and deleting records at his former place of employment using the computers of his new place of employment (I will bet the new place of employment is now another former place of employment). Apparently he made unwanted advances to a female at his first company and was fired for it. He worked in the Human Resources Department on employment records, compensation, payroll, and passwords of accounts. After he was terminated from his first company he landed a job at a new company. Guess what? The first company never changed those passwords. So Leung copied and then deleted about 1000 records from the first company over the Internet using computers at his second job. He also gave that woman’s file a makeover: a $40,000 a year RAISE and a $100,000 bonus. Then he created a Hotmail account in the woman’s name and sent an email to the executives of the first company from “her” with an attachment of her original file. Don’t try this at home boys and girls: Forensic images of the computer he used at the second company revealed the hotmail account was created with that computer. Boo-ya! Busted prison style!

Take college courses on a new schedule--yours.

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education. Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

http://e.spcollege.edu

St. Petersburg College’s eCampus. Your time. Your place. Your future.
**Multi Area OSPF**
By A. David Vasquez

**Objective:**
To configure the routing protocol OSPF over multiple areas, implement address summarization via “summary address” and “area range” commands as well as static default routes to the ISP from the multiple areas.

**Tools and Materials:**
(3) 2500 or 2600 series routers
(1) Rollover cable
(2) V.35 Serial cables

**Background:**
You are the network administrator for a small network with three different sites. The CCNP in charge of the NOC has a brilliant idea to convert the existing network from RIP v2 into an OSPF network because they just learned it in school. Since this person is your boss, it then becomes your job to make the migration possible. Currently the routing tables look sloppy with all of the different departments hanging off of them, you need to summarize the loopback addresses into one advertisement per router.

All configurations where tested and verified on three Cisco 2610 Access layer Routers running IOS version 12.0(3)T3.

**Lab Diagram:**

![Lab Diagram](image-url)
Step-by-step instructions:
1. The first thing you must do is set up the lab in accordance with the diagram provided.
2. Configure TOP, MIDDLE and BOTTOM routers with serial and loopback interface addresses.
3. Verify all necessary interfaces are in a state of “up and up” by issuing the following command:

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP-Address</th>
<th>OK?</th>
<th>Method</th>
<th>Status</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet0/0</td>
<td>unassigned</td>
<td>YES</td>
<td>NURAM</td>
<td>administratively down</td>
<td>down</td>
</tr>
<tr>
<td>Serial0/0</td>
<td>unassigned</td>
<td>YES</td>
<td>NURAM</td>
<td>administratively down</td>
<td>down</td>
</tr>
<tr>
<td>Serial0/1</td>
<td>10.0.0.5</td>
<td>YES</td>
<td>NURAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>Loopback0</td>
<td>1.1.0.1</td>
<td>YES</td>
<td>NURAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>Loopback1</td>
<td>1.1.1.1</td>
<td>YES</td>
<td>NURAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>Loopback2</td>
<td>1.1.2.1</td>
<td>YES</td>
<td>NURAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>Loopback3</td>
<td>1.1.3.1</td>
<td>YES</td>
<td>NURAM</td>
<td>up</td>
<td>up</td>
</tr>
</tbody>
</table>

If all necessary interfaces are not “up and up” troubleshoot as needed. (Hint: Don’t forget serial interfaces have DCE and DTE connections and the DCE side needs a Clock rate. It does not matter if the DCE is on serial 0/0 or serial 0/1 as long as you supply it a clock rate, furthermore, if there is no clock rate on the DCE side, the DTE will never be “up and up” it will remain in a state of “up and down”.) If you are unsure which end of the cable you are programming use the following command to determine DCE or DTE:
bottom# show controllers s0/0
Interface Serial0/0
Hardware is PowerQUICC MPC860
DTE V.35 TX and RX clocks detected.
(There is a lot more information below that has been left out)

middle# show controllers s0/0
Interface Serial0/0
Hardware is PowerQUICC MPC860
DCE V.35, clock rate 2000000
(There is a lot more information below that has been left out)

The above information shows the serial link between bottom and middle router. If you see “DTE V.35 TX and RX clocks not detected” check the DCE side and you will probably see “DCE V.35, no clock rate”

4. All of the interfaces should be “up and up”. Now we can configure OSPF over multiple areas:

First we will advertise the loopback networks on top router in area 64000, then the serial network in area 0.

```
top# config t
top(config)# router ospf 1
top(config-router)# network 1.1.0.0 0.0.0.255 area 64000
top(config-router)# network 1.1.1.0 0.0.0.255 area 64000
top(config-router)# network 1.1.2.0 0.0.0.255 area 64000
top(config-router)# network 1.1.3.0 0.0.0.255 area 64000
top(config-router)# network 10.0.0.4 0.0.0.3 area 0
```

Do a “show ip protocols” to verify you entered things correctly:

```
top# sh ip protocols
Routing Protocol is "ospf 1"
Sending updates every 0 seconds
Invalid after 0 seconds, hold down 0, flushed after 0
Outgoing update filter list for all interfaces is
Incoming update filter list for all interfaces is
Redirecting: ospf 1
Routing for Networks:
  1.1.0.0/24
  1.1.1.0/24
  1.1.2.0/24
  1.1.3.0/24
  10.0.0.4/30
```

Second, we will advertise the loopback and serial networks on middle router in area 0:

```
middle# config t
middle(config)# router ospf 1
middle(config-router)# network 172.16.1.32 0.0.0.31 area 0
middle(config-router)# network 172.16.1.64 0.0.0.31 area 0
middle(config-router)# network 172.16.1.96 0.0.0.31 area 0
```
Do a “show ip protocols” to verify you entered things correctly:

```
middle#sh ip protocols
Routing Protocol is "ospf 1"
Sending updates every 0 seconds
Invalid after 0 seconds, hold down 0, flushed after 0
Outgoing update filter list for all interfaces is
Incoming update filter list for all interfaces is
Redistributing: ospf 1
Routing for Networks:
  10.0.0.4/30
  10.0.0.8/30
  172.16.1.32/27
  172.16.1.64/27
  172.16.1.96/27
  172.16.1.128/27
```

Third, we will advertise the loopback networks on the bottom router in area 65000:

```
bottom#config t
bottom(config)#router ospf 1
bottom(config-router)#network 192.168.1.4 0.0.0.3 area 65000
bottom(config-router)#network 192.168.1.8 0.0.0.3 area 65000
bottom(config-router)#network 192.168.1.12 0.0.0.3 area 65000
bottom(config-router)#network 192.168.1.16 0.0.0.3 area 65000
```

Do a “show ip protocols” to verify you entered things correctly:

```
bottom#sh ip protocols
Routing Protocol is "ospf 1"
Sending updates every 0 seconds
Invalid after 0 seconds, hold down 0, flushed after 0
Outgoing update filter list for all interfaces is
Incoming update filter list for all interfaces is
Redistributing: ospf 1
Routing for Networks:
  10.0.0.8/30
  192.168.1.4/30
  192.168.1.8/30
  192.168.1.12/30
  192.168.1.16/30
```

5. Since you should be in your bottom router at this point lets go ahead and issue some commands to see what OSPF is doing:

```bash
bottom#sh ip ospf
Routing Process "ospf 1" with ID 192.168.1.17
Supports only single TOS(TOS0) routes
It is an area border router
```

Notice this is your highest loopback interface

Also known as ABR
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x0
Number of DCbitless external LSA 0
Number of DoNotAge external LSA 0
Number of areas in this router is 2. 2 normal 0 stub 0 nssa
External flood list length 0
Area BACKBONE(0)
Number of interfaces in this area is 1
Area has no authentication
SPF algorithm executed 6 times
Area ranges are
Number of LSA 11. Checksum Sum 0x4FF7C
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0
Area 65000
Number of interfaces in this area is
Area has no authentication
SPF algorithm executed 5 times
Area ranges are
Number of LSA 11. Checksum Sum 0x74712
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0

Verify your bottom router has established a neighbor adjacency with the middle router:

```
bottom#sh ip ospf neighbor
Neighbor ID     Pri   State    Dead Time   Address      Interface
172.16.1.129      1   FULL/  - 00:00:30    10.0.0.9     Serial0/0
```

Take a look at the Neighbor ID. Where is this ip address taken from? The address listed under the “Address” field is the ip address of the neighbor and the “Interface” is the local interface of the router and represents how the local router is connected to its neighbor.

Now let’s look at the routing table on the bottom router:

```
bottom#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR, P - periodic downloaded static route
T - traffic engineered route
```

Authentication? OSPF allows us to set up authentication so our routing updates are secured. We will have labs in the future showing how to configure this feature!
Gateway of last resort is not set

1.0.0.0/32 is subnetted, 4 subnets
O IA   1.1.1.1 [110/129] via 10.0.0.9, Serial0/0
O IA   1.1.0.1 [110/129] via 10.0.0.9, Serial0/0
O IA   1.1.3.1 [110/129] via 10.0.0.9, Serial0/0
O IA   1.1.2.1 [110/129] via 10.0.0.9, Serial0/0
172.16.0.0/32 is subnetted, 4 subnets
O   172.16.1.129 [110/65] via 10.0.0.9, Serial0/0
O   172.16.1.33 [110/65] via 10.0.0.9, Serial0/0
O   172.16.1.97 [110/65] via 10.0.0.9, Serial0/0
O   172.16.1.65 [110/65] via 10.0.0.9, Serial0/0
10.0.0.0/30 is subnetted, 2 subnets
C   10.0.0.8 is directly connected, Serial0/0
O   10.0.0.4 [110/128] via 10.0.0.9, Serial0/0
192.168.1.0/30 is subnetted, 4 subnets
C   192.168.1.8 is directly connected, Loopback1
C   192.168.1.12 is directly connected, Loopback2
C   192.168.1.14 is directly connected, Loopback0
C   192.168.1.16 is directly connected, Loopback3

6. Now let’s take a look at OSPF in the middle router:

```
middle# sh ip ospf
Routing Process "ospf 1" with ID 172.16.1.129
Supports only single TOS(TOS0) routes
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x0
Number of DCbitless external LSA 0
Number of DoNotAge external LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
External flood list length 0
Area BACKBONE(0)

Number of interfaces in this area is 6
Area has no authentication
SPF algorithm executed 8 times
Area ranges are
Number of LSA 11. Checksum Sum 0x4E987
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0
```

Verify your middle router has established a neighbor adjacency with both the top and bottom router:

```
middle# sh ip ospf neig
Neighbor ID       Pri State Dead Time   Address       Interface
192.168.1.17      1 FULL/  - 00:00:36 10.0.0.10       Serial0/0
1.1.3.1           1 FULL/  - 00:00:33 10.0.0.5        Serial0/1
```

Notice how these routes show up as “OSPF Inter area” routes. This is because they came from a different autonomous system (area 64000).

Since the bottom router’s serial 0/0 is a part of area 0, all of the area 0 routes from the middle router are seen as just “OSPF” routes. Notice 10.0.0.4. What area was that advertised in?

Take a look at the Neighbor ID. Where is this ip address taken from? The address listed under the “Address” field is the ip address of the neighbor and the “Interface” is the local interface of the router and represents how the local router is connected to its neighbor.
Now let's look at the routing table on the middle router:

```
middle#sh_ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
U - per-user static route, o - ODR, P - periodic downloaded
static route, T - traffic engineered route

Gateway of last resort is not set

1.0.0.0/32 is subnetted, 4 subnets
O IA  1.1.1.1 [110/65] via 10.0.0.5, Serial0/1
O IA  1.1.0.1 [110/65] via 10.0.0.5, Serial0/1
O IA  1.1.3.1 [110/65] via 10.0.0.5, Serial0/1
O IA  1.1.2.1 [110/65] via 10.0.0.5, Serial0/1
172.16.0.0/27 is subnetted, 4 subnets
C    172.16.1.128 is directly connected, Loopback3
C    172.16.1.32 is directly connected, Loopback0
C    172.16.1.96 is directly connected, Loopback2
C    172.16.1.64 is directly connected, Loopback1
10.0.0.0/30 is subnetted, 2 subnets
C    10.0.0.0.8 is directly connected, Serial0/0
C    10.0.0.4 is directly connected, Serial0/1
192.168.1.0/32 is subnetted, 4 subnets
O IA  192.168.1.9 [110/65] via 10.0.0.10, Serial0/0
O IA  192.168.1.13 [110/65] via 10.0.0.10, Serial0/0
O IA  192.168.1.15 [110/65] via 10.0.0.10, Serial0/0
O IA  192.168.1.17 [110/65] via 10.0.0.10, Serial0/0
```

Notice how these routes show up as “OSPF Inter area” routes. This is because they came from a different autonomous system (area 64000 and 65000).

7. Now let’s take a look at OSPF in the top router:

```

```
top#sh_ip ospf
Routing Process "ospf 1" with ID 1.1.3.1
Supports only single TOS(TOS0) routes
It is an area border router
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x0
Number of DCbitless external LSA 0
Number of DoNotAge external LSA 0
Number of areas in this router is 2. 2 normal 0 stub 0 nssa
External flood list length 0
Area BACKBONE(0)
  Number of interfaces in this area is 1
  Area has no authentication
  SPF algorithm executed 15 times
  Area ranges are
  Number of LSA 11. Checksum Sum 0x4DF8C
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
```

What are these? Stub? Not So Stubby Area (NSSA)? You can look up these terms online, I have attached a few links at the end of this lab to assist you with OSPF terms.
Verify your top router has established a neighbor adjacency with the middle router:

```
Verify your top router has established a neighbor adjacency with the middle router:

top# sh ip ospf nei
Neighbor ID   Pri   State    Dead Time       Address      Interface
172.16.1.129  1   FULL/   - 00:00:33   10.0.0.6     Serial0/1
```

Now let’s look at the routing table on the top router:

```
Now let’s look at the routing table on the top router:

top# sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
       U - per-user static route, o - ODR, P - periodic downloaded
       static route,T - traffic engineered route

Gateway of last resort is not set

1.0.0.0/24 is subnetted, 4 subnets
 C   1.1.0.0 is directly connected, Loopback0
 C   1.1.1.0 is directly connected, Loopback1
 C   1.1.2.0 is directly connected, Loopback2
 C   1.1.3.0 is directly connected, Loopback3

172.16.0.0/32 is subnetted, 4 subnets
 O   172.16.1.129 [110/65] via 10.0.0.6, Serial0/1
 O   172.16.1.33 [110/65] via 10.0.0.6, Serial0/1
 O   172.16.1.97 [110/65] via 10.0.0.6, Serial0/1
 O   172.16.1.65 [110/65] via 10.0.0.6, Serial0/1

10.0.0.0/30 is subnetted, 2 subnets
 O   10.0.0.8 [110/128] via 10.0.0.6, Serial0/1
 C   10.0.0.4 is directly connected, Serial0/1

192.168.1.0/32 is subnetted, 4 subnets
 O IA 192.168.1.9 [110/129] via 10.0.0.6, Serial0/1
 O IA 192.168.1.13 [110/129] via 10.0.0.6, Serial0/1
 O IA 192.168.1.15 [110/129] via 10.0.0.6, Serial0/1
 O IA 192.168.1.17 [110/129] via 10.0.0.6, Serial0/1
```

8. OK. Now OSPF is working great, but our routing tables are kind of messy, so let’s clean them up a bit. Since we are on the top router at this point let’s start here. Enter the following commands on the top router to summarize these routes:
top#conf t
top(config)#router ospf 1
top(config-router)#area 64000 range 1.1.0.0 255.255.252.0

When trying to figure out the appropriate ip address and subnet mask to be used in the “area range” or “summary address” command it is best to get out a pencil and paper. No matter how many times I have done this, I still use a pencil and paper to calculate the bits. Here is an example:

Remember the loopback ip addresses? Starting from the left, which of the four octets begins to show differences? (Hint: I have highlighted them for you 🙌)

- Loopback 0 1.1.0.1 /24
- Loopback 1 1.1.1.1 /24
- Loopback 2 1.1.2.1 /24
- Loopback 3 1.1.3.1 /24

The first two octets are the same ‘1.1’, the third octet is where things start to change. This is our starting point. Now you have to be patient because this requires us to go back in time. List all eight bits in the third octet and convert the decimal number to binary, like so:

3rd Octet

<table>
<thead>
<tr>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Look at the patterns the zeros and ones make:

- Trace the 128 column down… is the binary the same? Yes
- Trace the 64 column down… is the binary the same? Yes
- Trace the 32 column down… is the binary the same? Yes
- Trace the 16 column down… is the binary the same? Yes
- Trace the 8 column down… is the binary the same? Yes
- Trace the 4 column down… is the binary the same? Yes
- Trace the 2 column down… is the binary the same? NO

At this point you can stop. How many bit columns did you answer “Yes” to? The answer is six. This is how many bits you will use from this octet to calculate the subnet mask.
We have already determined the first two octets are the same and the change occurs in the third octet, so you can put a zero in the third and fourth octets. Our ip address is 1.1.0.0.

You may be wondering where the zero came from in the fourth octet…well since the similarities ended six bits into the third octet, we can automatically put a zero in the third and fourth octet. It would be the same if the similarities ended in the first octet, we would simply put a zero in the second, third and fourth octets. The same methodology goes for the subnet mask. The first two octets are identical, in other words they have the same bits in common, so we start with 255.255.?.?. Where did the similarities end? The third octet. How many bits did the respective ip addresses have in common? Six. Which six bits did they have in common? 128, 64, 32, 16, 8, 4. If we add those numbers up, what decimal number do we get? 252. That is the number we put in our third octet: 255.255.252.0 (Remember, we don’t care about the 4th octet, so just put a zero).

Here is the full-blown command from above:

```
top(config-router)#area 64000 range 1.1.0.0 255.255.252.0
```

Now that we got that out of the way, we can summarize our routes on the bottom router. Enter the following commands:

```
bottom#conf t
bottom(config)#router ospf 1
bottom(config-router)#area 65000 range 192.168.1.0 255.255.255.224
```

Remember the loopback ip addresses? Starting from the left, which of the four octets begins to show differences?

- Loopback 0 192.168.1.5 /30
- Loopback 1 192.168.1.9 /30
- Loopback 2 192.168.1.13 /30
- Loopback 3 192.168.1.17 /30

This one will be a little trickier because all of our work will be done in the fourth octet.

Go ahead and break out the pencil and paper…here goes:

```
<table>
<thead>
<tr>
<th>4th Octet</th>
</tr>
</thead>
<tbody>
<tr>
<td>128  64  32 16  8  4  2  1</td>
</tr>
<tr>
<td>0   0   0   0   0   1   0   1 =5 (4th octet loop 0)</td>
</tr>
<tr>
<td>0   0   0   0   1   0   0   1 =9 (4th octet loop 1)</td>
</tr>
<tr>
<td>0   0   0   0   1   1   0   1 =13 (4th octet loop 2)</td>
</tr>
<tr>
<td>0   0   0   1   0   0   0   1 =17 (4th octet loop 3)</td>
</tr>
</tbody>
</table>
```

We have already determined the first three octets are the same and the change occurs in the fourth octet, so you can put a zero in that octet. So our ip address is 192.168.1.0. The first three octets are identical, in other words they have the same bits in common, so we start with 255.255.255.?. Where did the similarities end? The fourth octet. How many bits did the respective ip addresses have in common? Three. Which three bits did they
have in common? 128, 64, 32. If we add those numbers up, what decimal number do we get? 224. So our subnet mask is 255.255.255.224

Here is the full-blown command from above:

```
bottom(config-router)#area 65000 range 192.168.1.0 255.255.255.224
```

We must be careful with route summarizations because we do not want to summarize routes that do not belong to us!

Now let’s take a look at the middle router and compare before and after the summarizations in the top and bottom router:

```
middle#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
U - per-user static route, o - ODR, P - periodic downloaded
static route, T - traffic engineered route

Gateway of last resort is not set

1.0.0.0/32 is subnetted, 4 subnets
O IA    1.1.1.1 [110/65] via 10.0.0.5, Serial0/1
O IA    1.1.0.1 [110/65] via 10.0.0.5, Serial0/1
O IA    1.1.3.1 [110/65] via 10.0.0.5, Serial0/1
O IA    1.1.2.1 [110/65] via 10.0.0.5, Serial0/1

172.16.0.0/27 is subnetted, 4 subnets
C       172.16.1.128 is directly connected, Loopback3
C       172.16.1.32 is directly connected, Loopback0
C       172.16.1.96 is directly connected, Loopback2
C       172.16.1.64 is directly connected, Loopback1

10.0.0.0/30 is subnetted, 2 subnets
C       10.0.0.8 is directly connected, Serial0/0
C       10.0.0.4 is directly connected, Serial0/1

192.168.1.0/24 is subnetted, 4 subnets
O IA    192.168.1.9 [110/65] via 10.0.0.10, Serial0/0
O IA    192.168.1.13 [110/65] via 10.0.0.10, Serial0/0
O IA    192.168.1.15 [110/65] via 10.0.0.10, Serial0/0
O IA    192.168.1.17 [110/65] via 10.0.0.10, Serial0/0
```

```
middle#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
U - per-user static route, o - ODR, P - periodic downloaded
static route, T - traffic engineered route

BEFORE summarization

BEFORE summarization
```
Gateway of last resort is not set

1.0.0.0/22 is subnetted, 1 subnets
O IA  1.1.0.0 [110/65] via 10.0.0.5, Serial0/1
   172.16.0.0/27 is subnetted, 3 subnets
   C 172.16.1.32 is directly connected, Loopback0
   C 172.16.1.96 is directly connected, Loopback2
   C 172.16.1.64 is directly connected, Loopback1
   10.0.0.0/30 is subnetted, 2 subnets
   C 10.0.0.8 is directly connected, Serial0/0
   C 10.0.0.4 is directly connected, Serial0/1
192.168.1.0/27 is subnetted, 1 subnets
O IA  192.168.1.0 [110/65] via 10.0.0.10, Serial0/0

1. Take a look at the top and bottom router and see what has changed. What do you think the tables will look like?
2. Try figuring out the subnet masks again without looking at the explanation.
3. Erase the routers and come up with your own IP addressing scheme, then see if you can summarize multiple routes.
4. Try configuring a stub area, NSSA, and/or a totally stubby area.

Here are a few links I found on the Internet that might help to explain some of the OSPF terms

http://en.wikipedia.org/wiki/OSPF
http://asg.web.cmu.edu/rfc/rfc2178.html

Keep track of updates and changes at http://www.spcollege.edu/star/cisco Scroll to the bottom of the page and click on “Lab Manual Edits.”
Border Gateway Protocol (BGP)

**Objective:**
To learn the basics of setting up a one subnet BGP network and redistributing it with EIGRP.

**Tools and Materials:**
- (2) PC/workstations
- (3) Routers
- (2) Switches
- (4) Straight-through cables
- (2) DCE/DTE serial cable
- (2) rollover cables

**Lab Diagram:**

```
L0
ISP
s0

BGP 100
s1 Cult Deadcow

BGP 200
st con s0

st ro ro st

NIC
COM1

EIGRP 13

Workstation “A”

Workstation “B”

NIC
COM1
```

**Addressing:**

<table>
<thead>
<tr>
<th></th>
<th>ISP</th>
<th>Cult</th>
<th>Deadcow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routers Hostnames</td>
<td>E0  n/a 192.168.1.1/24</td>
<td>S0 210.1.1.1/24 (DCE) 192.168.2.1/24(DCE) n/a</td>
<td>S1 210.1.1.2/24 192.168.2.2/24</td>
</tr>
<tr>
<td>Loopback</td>
<td>193.168.1.1/24 (L0)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A  192.168.1.3</th>
<th>B  192.168.3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workstations</td>
<td>IP 255.255.255.0</td>
<td>SM 255.255.255.0</td>
</tr>
<tr>
<td></td>
<td>GW 192.168.1.1</td>
<td></td>
</tr>
</tbody>
</table>
**Background:**

BGP is primarily used between ISP’s for routing. In other words, it “is” the Internet. Right now there are about 100,000 BGP routes in the Internet. Unlike RIP, IGRP, or EIGRP you wouldn’t want to use BGP in a small network. Save this routing protocol for the huge corporations and Internet Service Providers. Some people think it is a very difficult protocol to configure and maintain while others think it is “a piece of cake...as long as you know what you are doing.” We are only going to touch on the real basics here. BGP is a very involved protocol and worthy of an entire course at the CCNP level at the least. Routers using BGP only exchange full routing tables when the connection is first established. After that there are no periodic updates, only when a change occurs. And then only the optimal route is broadcast not the entire table.

**Step-By-Step Instructions:**

1. Set up and cable the lab as shown. Put all the basics on the routers except for the routing protocols.
2. Between Cult and Deadcow enable EIGRP with an autonomous system number of 13.

```
cult(config)#router eigrp 13
cult(config-router)#network 192.168.1.0
cult(config-router)#network 192.168.2.0

deadcow(config)#router eigrp 13
deadcow(config-router)#network 192.168.2.0
deadcow(config-router)#network 192.168.3.0
```

Test those routes between cult and deadcow. Now let’s move on to BGP.

```
cult#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
       U - per-user static route, o - ODR
Gateway of last resort is not set

C  192.168.1.0/24 is directly connected, Ethernet0/0
C  210.1.1.0/24 is directly connected, Serial0/1
C  192.168.2.0/24 is directly connected, Serial0/0
```
3. Let’s add in the BGP. It too uses an autonomous system number. Let’s use 100 for the ISP and 200 for our serial 1 interface.

ISP(config)#router bgp 100
ISP(config-router)#no synchronization
ISP(config-router)#network 193.168.1.0
ISP(config-router)#network 210.1.1.0
ISP(config-router)#neighbor 210.1.1.2 remote-as 200

cult(config)#router bgp 200
cult(config-router)#no synchronization
cult(config-router)#network 210.1.1.0
cult(config-router)#redistribute eigrp 13
cult(config-router)#neighbor 210.1.1.1 remote-as 100

4. Next we need to redistribute our routing protocols:

cult(config)#router eigrp 13
cult(config-router)#redistribute bgp 200
cult(config-router)#passive-interface Serial0/1
cult(config-router)#default-metric 1000 100 250 100 1500

cult(config)#router bgp 200
cult(config-router)#redistribute eigrp 13

5. Now let’s see our ip routes:

ISP#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is not set

C   193.168.1.0/24 is directly connected, Loopback0
B   192.168.1.0/24 [20/0] via 210.1.1.2, 00:09:52
C   210.1.1.0/24 is directly connected, Serial0/0
B   192.168.2.0/24 [20/0] via 210.1.1.2, 00:09:52
B   192.168.3.0/24 [20/2195456] via 210.1.1.2, 00:07:14
6. We can use a command called show ip bgp to examine our bgp routes:
cult#sh ip bgp
BGP table version is 24, local router ID is 210.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>*&gt; 192.168.1.0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>32768</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 192.168.2.0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>32768</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 192.168.3.0</td>
<td>192.168.2.2</td>
<td>2195456</td>
<td>32768</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>*&gt; 193.168.1.0</td>
<td>210.1.1.1</td>
<td>0</td>
<td></td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>*  210.1.1.0</td>
<td>210.1.1.1</td>
<td>0</td>
<td></td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>*&gt;</td>
<td>0.0.0.0</td>
<td>0</td>
<td>32768</td>
<td></td>
<td>i</td>
</tr>
</tbody>
</table>

deadcow#sh ip bgp
% BGP not active

Supplemental Lab or Challenge Activities:
1. Go out to CISCO and find out what are the definitions and descriptions of the metrics.
2. Find out what is the difference between IBGP and EBGP.
3. How would you redistribute BGP with IGRP? RIP?
4. Try using your protocol inspector to capture BGP packets. Examine their structure carefully.
5. For what is the “no synchronization” command used? What about the “passive-interface” command?
6. When we use a clockrate command we have been using 56000. We know T-1 lines are much faster than that…what is the upper limit of our clockrate command? (hint: it’s in the millions)

So What Have I Learned Here?
In this lab you learned the very basics of the BGP routing protocol. Trust me…you just touched on the tip of the iceberg here.

Guest Router Name
Cult of the Dead Cow (CdC) is a hacking gang who have been publishing their hacking materials since the 1980’s. One of their more famous contributions is the software known as “Back Orifice tool for Windows.” This program, when installed on a computer, makes it very easy for a hacker to manipulate the workstation just like a puppeteer does with a puppet.
The Border Gateway Protocol (BGP) is an inter-autonomous system routing protocol. An autonomous system is a network or group of networks under a common administration and with common routing policies. All BGP AS numbers are assigned by ARIN (unlike IGP ASs which can use any number).

BGP is used to exchange routing information for the Internet and is the protocol used between Internet Service Providers. Customer networks, such as universities and corporations, usually employ an Interior Gateway Protocol (IGP) such as RIP or OSPF for the exchange of routing information within their networks. Customers connect to ISPs, and ISPs use BGP to exchange customer and ISP routes.

External and Interior BGP:

When BGP is used between autonomous systems, the protocol is referred to as External BGP (EBGP). If a service provider is using BGP to exchange routes within an AS, then the protocol is referred to as Interior BGP (IBGP).

BGP is a very robust and scalable routing protocol, as evidenced by the fact that BGP is the routing protocol employed on the Internet. The Internet BGP routing tables number more than 120,000 routes.

How BGP Works:

BGP does not use technical metrics, it uses policy-based routing. BGP uses IGPs (RIP, IGRP) to advertise external routes into ASs. BGP will get external routes to the front door of an AS, an IGP must take them in from there.

BGP uses the network command to advertise IGP routes into the BGP network. *BGP neighbors exchange full routing information when the TCP connection between neighbors is first established. When changes to the routing table are detected, the BGP routers send to their neighbors only those routes that have changed. BGP routers do not send periodic routing updates, and BGP routing updates advertise only the optimal path to a destination network.

BGP Updates:

BGP updates are carried in TCP segments. TCP connections must be established between devices before updates can be exchanged. To enable TCP connections, you must manually define peer connections.

BGP and IGP Synchronization:

*BGP cannot advertise routes that don’t already exist in the IP routing table. BGP running within an AS (IBGP) is dependent on an IGP to build IP routing tables. The no synchronization command allows BGP to advertise routes that are known to it, even if they are not in the IP routing table.
**BGP Basic Configuration**

**Directions:** In the following lab, you will configure BGP to enable dynamic routing for your network. In this basic BGP configuration, SanJose in AS-100 will be external neighbors to ISP1a in AS-200. ISP1b will be an internal neighbor to ISP1a in AS-200. Routers A+B are connected serially and ISP1b is connected to ISP1a via an Ethernet link. All routers have loopback addresses defined. The internetworking model is detailed below.

---

### Configuring BGP:

In this procedure, you will configure BGP on the routers, specifying the AS, the neighboring router, and the AS the neighbor is in.

- **SanJose**
  ```
  SanJose(config)# router bgp 100
  SanJose(config-router)# neighbor 172.16.20.2 remote-as 200
  ```

- **ISP1a**
  ```
  ISP1a(config)# router bgp 200
  ISP1a(config-router)# neighbor 172.16.20.1 remote-as 100
  ISP1a(config-router)# neighbor 172.16.30.2 remote-as 200
  ```

- **ISP1b**
  ```
  ISP1b(config)# router bgp 200
  ISP1b(config-router)# neighbor 172.16.30.1 remote-as 200
  ```
Monitoring and Testing the Configuration:
In this procedure, you will display the status of the BGP neighbor negotiation process, build a BGP routing table, and test BGP route functionality.

Type `sh ip bgp neighbors` to display your BGP neighbors.

```
SanJose#sh ip bgp neighbors
BGP neighbor is 172.16.20.2, remote AS 200, external link
   Index 1, Offset 0, Mask 0x2
   BGP version 4, remote Router ID 2.2.2.2
   BGP state = Established, table version = 1, up for 00:04:36
```

Notice the following:
- the remote AS of your BGP neighbor is listed.
- an external or internal BGP connection is specified.
- the remote router’s ID is its loopback address.
- the state of the BGP connection is established. If any other state is shown, there is a problem with the BGP connection.

Advertising a Network via BGP:
In order for a router to advertise a network to another BGP speaker, the network must already be present in the routing table of the advertising router. Since your routers have no remote routes in their routing tables you will advertise a network that is directly connected to the router.

```
SanJose(config)# router bgp 100
SanJose(config-router)# network 1.0.0.0
```

Displaying the BGP Table:
In this procedure, you will display the contents of the BGP table on your router.

On ISP1a, type `sh ip bgp` to display the BGP table.

```
ISP1a#sh ip bgp
Network    Next Hop       Metric LocPrf Weight Path
*> 1.0.0.0     172.16.20.1  0      0        100     I
```

Notice the following:
- this is the best route to use for the network, indicated by the `>` sign before the network number.
- the network was learned via the next hop address of 172.16.20.1, which is the serial interface of SanJose.
- the entry originated from an IGP or is a directly connected network, indicated by the `i`.  

438
On ISP1a, type `sh ip route` to display the BGP table.

```
ISP1a#sh ip route
B 1.0.0.0/8 [20/0] via 172.16.20.1, 09:10:46
  2.0.0.0/24 is subnetted, 1 subnets
  C 2.2.2.0 is directly connected, Loopback0
  172.16.0.0/24 is subnetted, 2 subnets
  C 172.16.30.0 is directly connected, Ethernet0
  C 172.16.20.0 is directly connected, Serial0
```

☞ Notice the following: - the router has the route to network 1.0.0.0 via 172.16.20.1, which is Router A’s serial interface.

On ISP1b, type `sh ip bgp` to display the BGP table.

```
ISP1b#sh ip bgp
Network      Next Hop            Metric LocPrf  Weight Path
* 1.0.0.0       172.16.20.1       0     0      100      I
```

☞ Notice the following: - the next hop displayed to reach network 1.0.0.0 is unreachable from ISP1b, and is thus useless to ISP1b.

On ISP1b, type `sh ip route` to display the BGP table.

```
ISP1b#sh ip route
3.0.0.0/24 is subnetted, 1 subnets
  C 3.3.3.0 is directly connected, Loopback0
  172.16.0.0/24 is subnetted, 1 subnets
  C 172.16.30.0 is directly connected, Ethernet0
```

☞ Notice the following: - the router has no route to network 1.0.0.0.

☞ The route to network 1.0.0.0 is not in the routing table of ISP1b because the next hop advertised is 172.16.20.1, which is unreachable to ISP1b. When routes are injected into the AS via EBGP, the next hop learned from EBGP is carried unaltered into IBGP.

**Fixing the Next-hop Address Problem:**

In this procedure, you will force ISP1a to advertise itself as the next hop for all BGP updates being sent to ISP1b.

```
ISP1a(config)# router bgp 200
ISP1a(config-router)# neighbor 172.16.30.2 next-hop-self
```
Resetting the BGP Session:
Anytime a configuration change is made to an established BGP neighbor, the BGP session with that neighbor must be reset.

ISP1b# clear ip bgp 172.16.30.1

This command will reset only the specific neighbor 172.16.30.1.

On ISP1b, type `sh ip bgp` to display the BGP table.

```
ISP1b# sh ip bgp
          Network      Next Hop          Metric LocPrf    Weight   Path
        * 1.0.0.0            172.16.30.1     0        0        100      I
```

Notice the following: - the next hop displayed to reach network is correct.

Disabling Synchronization:
Before BGP can announce any route, the route must already be present in its IP routing table. Since we are not running an IGP in the AS, there is no route to network 1.0.0.0 in the IP routing table, even though it exists in the BGP table. To enable BGP to advertise the routes known to it, but absent from the IP routing table, disable synchronization between BGP and IGP.

ISP1b(config)# router bgp 200
ISP1b(config-router)# no synchronization
ISP1b(config-router)# ^Z
ISP1b# clear ip bgp *

On ISP1b, type `sh ip route` to display the BGP table.

```
ISP1b# sh ip route
         B  1.0.0.0/8 [200/0] via 172.16.30.1, 00:00:49
      3.0.0.0/24 is subnetted, 1 subnets
C      3.3.3.0 is directly connected, Loopback0
      172.16.0.0/24 is subnetted, 1 subnets
C      172.16.30.0 is directly connected, Ethernet0
```

Notice the following: - the BGP generated route to network 1.0.0.0 appears in the routing table.
- BGP calculates no technical metrics for routes.
Overview:
BGP uses a set of parameters (attributes) that describe the characteristics of a route. The attributes are sent in the BGP update packets with each route. The router uses these attributes to select the best route to the destination. In this lab, we will explore manipulating these attributes to control BGP path selections.

Configuration:
All routers will be configured for BGP. OSPF will be used as the IGP within AS 200. RouterA is in AS100 and will be external BGP neighbors with RouterB and RouterC, which are in AS200. RouterB and RouterC will run IBGP to RouterD, which is also in AS200.

All routers are connected serially. RouterB will act as the DCE supplying clock to RouterA and RouterD. RouterC will also act as the DCE supplying clock for RouterD and RouterA. The IP addresses are assigned as shown below.

BGP Configurations
The BGP and OSPF configuration for the four routers are as follows.

RouterA(config)# router bgp 100
RouterA(config-router)# network 1.1.1.0 mask 255.255.255.0
RouterA(config-router)# network 2.2.2.0 mask 255.255.255.0
RouterA(config-router)# neighbor 192.1.1.2 remote-as 200
RouterA(config-router)# neighbor 193.1.1.2 remote-as 200
RouterB(config)# router ospf 1
RouterB(config-router)# network 194.1.1.0 0.0.0.255 area 0
RouterB(config-router)# router bgp 200
RouterB(config-router)# no synchronization
RouterB(config-router)# neighbor 192.1.1.1 remote-as 100
RouterB(config-router)# neighbor 194.1.1.1 remote-as 200
RouterB(config-router)# neighbor 195.1.1.2 remote-as 200
RouterB(config-router)# neighbor 194.1.1.1 next-hop-self
RouterB(config-router)# neighbor 195.1.1.2 next-hop-self

RouterC(config)# router ospf 1
RouterC(config-router)# network 195.1.1.0 0.0.0.255 area 0
RouterC(config-router)# router bgp 200
RouterC(config-router)# no synchronization
RouterC(config-router)# neighbor 193.1.1.1 remote-as 100
RouterC(config-router)# neighbor 194.1.1.2 remote-as 200
RouterC(config-router)# neighbor 195.1.1.1 remote-as 200
RouterC(config-router)# neighbor 195.1.1.1 next-hop-self
RouterC(config-router)# neighbor 194.1.1.2 next-hop-self

RouterD(config)# router ospf 1
RouterD(config-router)# network 194.1.1.0 0.0.0.255 area 0
RouterD(config-router)# network 195.1.1.0 0.0.0.255 area 0
RouterD(config-router)# router bgp 200
RouterD(config-router)# no synchronization
RouterD(config-router)# neighbor 194.1.1.2 remote-as 200
RouterD(config-router)# neighbor 195.1.1.2 remote-as 200

Monitoring and Testing the Configuration
Display the BGP table on all routers with the command `show ip bgp`.

RouterA#sh ip bgp
BGP table version is 3, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* 1.1.1.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>32768</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>* 2.2.2.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>32768</td>
<td>i</td>
<td></td>
</tr>
</tbody>
</table>

Take college courses on a new schedule—yours.

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education. Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

http://e.spcollege.edu

St. Petersburg College’s eCampus. Your time. Your place. Your future.
Notice the best path (which is indicated by the >) is through RouterB (192.1.1.0). Remember the ten decision steps that BGP goes through to select the best path -- since all other things were equal, the route from the router with the lowest RouterID is used.
This can be verified through the command `show ip bgp neighbors`. Notice that RouterB's router ID is 194.1.1.2 and RouterC's router ID is 195.1.1.2.

RouterD# `show ip bgp neighbors`

```
BGP neighbor is 194.1.1.2, remote AS 200, internal link
Index 0, Offset 0, Mask 0x0
BGP version 4, remote router ID 194.1.1.2
BGP state = Established, table version = 11, up for 00:11:56
Last read 00:00:56, hold time is 180, keepalive interval is 60 seconds
Minimum time between advertisement runs is 5 seconds
Received 91 messages, 0 notifications, 0 in queue
Sent 83 messages, 0 notifications, 0 in queue
Connections established 6; dropped 5
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 194.1.1.1, Local port: 179
Foreign host: 194.1.1.2, Foreign port: 11006

BGP neighbor is 195.1.1.2, remote AS 200, internal link
Index 0, Offset 0, Mask 0x0
BGP version 4, remote router ID 195.1.1.2
BGP state = Established, table version = 11, up for 00:11:40
Last read 00:00:40, hold time is 180, keepalive interval is 60 seconds
Minimum time between advertisement runs is 5 seconds
Received 103 messages, 0 notifications, 0 in queue
Sent 91 messages, 0 notifications, 0 in queue
Connections established 8; dropped 7
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 195.1.1.1, Local port: 179
Foreign host: 195.1.1.2, Foreign port: 11031
```

**Local Preference Attribute**

The local preference attribute is a degree of preference given to a BGP route to compare it with other routes to the same destination. This is the second highest attribute used in the BGP decision process (Cisco proprietary weight parameter is first). The local preference attribute only is local to the autonomous system and does not get passed to EIGRP neighbors. The higher the local preference, the more preferred the route is.

In this exercise we will configure RouterC to set the local preference for network 1.0.0.0 learned from RouterA to 200. Since the default local preference is 100, all routers in AS 200 will prefer the path through RouterC to reach network 1.0.0.0.

In order to manipulate the local preference, we need to define what routes will be manipulated through the use of an access list, define the policy that will be applied to those routes through a route map, and then assign the route map to a BGP neighbor.

1. Add access-list 1 to RouterC, permitting network 1.0.0.0:

   ```
   RouterC#configure terminal
   RouterC(config)#access-list 1 permit 1.0.0.0 0.255.255.255
   ```
2. Define a route map named **localpref** that sets the local preference of the route to 200 if it matches access-list 1 and 100 if it does not.

RouterC(config)#route-map localpref 10
RouterC(config-route-map)#match ip address 1
RouterC(config-route-map)#set local-preference 200
RouterC(config-route-map)#route-map localpref permit 20
RouterC(config-route-map)#set local-preference 100

3. Apply the route map to inbound traffic from BGP neighbor 193.1.1.1 (RouterA).

RouterC(config)#router bgp 200
RouterC(config-router)#neighbor 193.1.1.1 route-map localpref in

In order for the changes to take effect, the BGP neighbors must be reset. To do this, use the command **clear ip bgp ***. This causes the TCP session between neighbors to be reset, restarting the neighbor negotiations from scratch and invalidating the cache.

RouterC#clear ip bgp *

Display the BGP table on RouterD with the command **show ip bgp**. Notice the local preference of the route RouterC is now 200 and is the best route (indicated by the > sign).

RouterD# sh ip bgp
BGP table version is 4, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, >best,i -int
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>*&gt;i1.1.1.0/24</td>
<td>195.1.1.2</td>
<td>0</td>
<td>200</td>
<td>0</td>
<td>100 i</td>
</tr>
<tr>
<td>*&gt;i2.2.2.0/24</td>
<td>194.1.1.2</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>100 i</td>
</tr>
<tr>
<td>* i</td>
<td>195.1.1.2</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>100 i</td>
</tr>
</tbody>
</table>

Display the BGP table on RouterB with the command **show ip bgp**. The following is the output from the command. Notice that RouterB is also using the route advertised from RouterC to reach network 1.0.0.0.

Keep track of updates and changes at [http://www.spcollege.edu/star/cisco](http://www.spcollege.edu/star/cisco) Scroll to the bottom of the page and click on the “Lab Manual Edits.”
RouterB# show ip bgp

BGP table version is 4, local router ID is 194.1.1.2
Status codes: s suppressed, d damped, h history, * valid, >best, i -int
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1.0/24</td>
<td>192.1.1.1</td>
<td>0</td>
<td>0</td>
<td>100 i</td>
<td></td>
</tr>
<tr>
<td>2.2.2.0/24</td>
<td>195.1.1.2</td>
<td>200</td>
<td>0</td>
<td>100 i</td>
<td></td>
</tr>
<tr>
<td>* i</td>
<td>195.1.1.2</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>100 i</td>
</tr>
</tbody>
</table>

Route learned from RouterC
The Multi-Exit Discriminator (MED) attribute is the external metric of a route. Unlike the local preference attribute, the MED is exchanged between ASs; however, the MED that comes into an AS does not leave. As shown in the last section, local preference was used by the AS to influence its own outbound decision process. The MED can be used to influence the outbound decision of another AS. The lower the MED, the more preferred the route. In the figure below, RouterA sets the MED attribute for network 1.0.0.0 to 50 before advertising it to RouterC and 100 before advertising it to RouterB.

The routers in AS 200 will prefer the route through RouterC because it has the lowest MED. In order to manipulate the MED, we need to identify what networks will be manipulated through the use of an access list, define a policy that will be applied to those routes through a route map, and then assign the route map to a BGP neighbor.

Remove the map statement on RouterC:

```
RouterC(config)#router bgp 200
RouterC(config-router)#no neighbor 193.1.1.1 route-map localpref in
```
1. Add access-list 1 to RouterA, permitting network 1.0.0.0:

   \[
   \text{RouterA(config)} \# \text{access-list 1 permit 1.0.0.0 0.255.255.255}
   \]

2. Define two route maps, one named \text{set_med_50} and the other named \text{set_med_100}. The first route map sets the MED attribute for network 1.0.0.0 to 50, and the latter sets the MED attribute to 100.

   \[
   \text{RouterA(config)} \# \text{route-map set_med_50 10}
   \]
   \[
   \text{RouterA(config-route-map)} \# \text{match ip address 1}
   \]
   \[
   \text{RouterA(config-route-map)} \# \text{set metric 50}
   \]
   \[
   \text{RouterA(config-route-map)} \# \text{exit}
   \]
   \[
   \text{RouterA(config)} \# \text{route-map set_med_50 20}
   \]
   \[
   \text{RouterA(config-route-map)} \# \text{set metric}
   \]
   \[
   \text{RouterA(config-route-map)} \# \text{exit}
   \]

   \[
   \text{RouterA(config)} \# \text{route-map set med_100 10}
   \]
   \[
   \text{RouterA(config-route-map)} \# \text{match ip address 1}
   \]
   \[
   \text{RouterA(config-route-map)} \# \text{set metric 100}
   \]
   \[
   \text{RouterA(config-route-map)} \# \text{exit}
   \]
   \[
   \text{RouterA(config)} \# \text{route-map set_med_100 20}
   \]
   \[
   \text{RouterA(config-route-map)} \# \text{set metric}
   \]

Apply route map \text{set_med_50} on outbound routing updates to RouterC (193.1.1.2) and route map \text{set_med_100} on outbound routing updates to RouterB (192.1.1.2).

   \[
   \text{RouterA(config)} \# \text{router bgp 100}
   \]
   \[
   \text{RouterA(config-router)} \# \text{neighbor 193.1.1.2 route-map set_med_50 out}
   \]
   \[
   \text{RouterA(config-router)} \# \text{neighbor 192.1.1.2 route-map set_med_100 out}
   \]

In order for the changes to take effect, the BGP neighbors must be reset. To do this, use the command \text{clear ip bgp *}. This causes the TCP session between neighbors to be reset, restarting the neighbor negotiations from scratch and invalidating the cache.

   \[
   \text{RouterC#clear ip bgp *}
   \]

Display the BGP table on RouterB with the command \text{show ip bgp}. The following is the output from the command. Notice that the route to network 1.0.0.0 learned via 193.1.1.1 has a local preference of 50 and is the preferred route.
RouterB# `show ip bgp`

BGP table version is 8, local router ID is 194.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i -int
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>*&gt;1.1.1.0/24</td>
<td>195.1.1.2</td>
<td>50</td>
<td>100</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>*</td>
<td>192.1.1.1</td>
<td></td>
<td></td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>*&gt;2.2.2.0/24</td>
<td>192.1.1.1</td>
<td>0</td>
<td></td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>*</td>
<td>195.1.1.2</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

From RouterA, display the route maps that are being used with the command `show route-maps`. This command tells what access list is used by the match clause, and what set clause is applied and how many times it has been used. This command is very useful in troubleshooting possible route-map problems.

RouterA# `show route-map`

`route-map SET-MED-50, permit, sequence 10`
Match clauses:
  `ip address (access-lists): 1`
Set clauses:
  metric 50
Policy routing matches: 0 packets, 0 bytes

`route-map SET-MED-50, permit, sequence 20`
Match clauses:
Set clauses:
Policy routing matches: 0 packets, 0 bytes

`route-map SET-MED-100, permit, sequence 10`
Match clauses:
  `ip address (access-lists): 1`
Set clauses:
  metric 100
Policy routing matches: 0 packets, 0 bytes

`route-map SET-MED-100, permit, sequence 20`
Match clauses:
Set clauses:
Policy routing matches: 0 packets, 0 bytes

**AS Path Manipulation**

BGP always prefers the route with the shortest AS path. In this exercise we will configure RouterA to prepend two extra AS path numbers to network 1.0.0.0 (AS300 and AS400) before advertising this network to RouterC and RouterB.

In order to manipulate the AS path information, we need to identify which routes will be manipulated through the use of an access list, define a policy that will be applied to those routes through a route map, and then assign the route map to a BGP neighbor.
1. Add access-list 1 to RouterA, permitting network 1.0.0.0:

   RouterA(config)#access-list 1 permit 1.0.0.0 0.255.255.255

2. Define a route map named AS-Path that prepends two additional AS path numbers (AS300 and AS400) to the route if it matches access list 1.

   RouterA(config)#route-map AS-Path permit 10
   RouterA(config-route-map)#match ip address 1
   RouterA(config-route-map)#set as-path prepend 300 400
   RouterA(config-route-map)#exit
   RouterA(config)#route-map AS-Path 20
   RouterA(config-route-map)#set as-path prepend
   RouterA(config-route-map)#exit

Apply the route map to outbound routing updates to BGP neighbor 193.1.1.2 (RouterC) and neighbor 192.1.1.2 (RouterB).

   RouterA(config)# router bgp 100
   RouterA(config-router)# neighbor 193.1.1.2 route-map AS_Path out
   RouterA(config-router)# neighbor 192.1.1.2 route-map AS_Path out

In order for the changes to take effect, the BGP neighbors must be reset. To do this, use the command clear ip bgp * . This causes the TCP session between neighbors to be reset, restarting the neighbor negotiations from scratch and invalidating the cache.

   RouterA#clear ip bgp *

Display the BGP table on RouterB with the command show ip bgp. The following is the output from the command. Notice that the route to network 1.0.0.0 now has an AS path of [100 300 400].

   RouterB#sh ip bgp
   BGP table version is 9, local router ID is 194.1.1.2
   Status codes: s suppressed, d damped, h history, * valid, >best, i-int
   Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* i1.1.1.0/24</td>
<td>195.1.1.1</td>
<td>100</td>
<td>0</td>
<td>100</td>
<td>300 400 i</td>
</tr>
<tr>
<td>*&gt; 2.2.2.0/24</td>
<td>192.1.1.1</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>400 i</td>
</tr>
<tr>
<td>* i</td>
<td>195.1.1.2</td>
<td>100</td>
<td>0</td>
<td>100</td>
<td>i</td>
</tr>
</tbody>
</table>

**Route Filtering Based on Network Number**

The router can filter routing updates to and from a particular neighbor based on the network number. The filter is made up of an access list that is applied to all BGP updates that are sent to or received from a particular neighbor.

In this exercise we will configure a distribute list on RouterA to prevent prefix 1.0.0.0 /8 from being advertised into AS 200.
In order to filter routes based on network address, we need to identify network addresses through the use of an access list and apply that list to a BGP neighbor using a distribute list.

1. Define the access list on RouterA to deny network 1.0.0.0 /8.

   ```
   RouterA(config)#access-list 2 deny 1.0.0.0 0.255.255.255
   RouterA(config)#access-list 2 permit any
   ```

2. Apply the distribution list to both BGP neighbors.

   ```
   RouterA(config)#router bgp 100
   RouterA(config-router)#neighbor 193.1.1.2 distribute-list 2 out
   RouterA(config-router)#neighbor 192.1.1.2 distribute-list 2 out
   ```

In order for the changes to take effect, the BGP neighbors must be reset. To do this use the command `clear ip bgp *`. This causes the TCP session between neighbors to be reset, restarting the neighbor negotiations from scratch and invalidating the cache.

   ```
   RouterA#clear ip bgp *
   ```

Display the routes that are being advertised via BGP to neighbor 193.1.1.2 with the command `show ip bgp neighbors 193.1.1.2 advertised-routes`. The following is the output from the command. Notice that RouterA is now only advertising network 2.0.0.0.

   ```
   RouterA# sh ip bgp neighbors 193.1.1.2 advertised-routes
   ```

   ```
   BGP table version is 3, local router ID is 2.2.2.2
   Status codes: s suppressed, d damped, h history, * valid, > best, i-int
   Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop    Metric  LocPrf    Weight    Path
   *> 2.2.2.0/24  0.0.0.0       0              32768      i
   ```

Display the BGP table on RouterB with the command `show ip bgp`. The following is the output from the command. Notice that the route to network 1.0.0.0 is no longer in the BGP table.

   ```
   RouterB#sh ip bgp
   ```

   ```
   BGP table version is 11, local router ID is 194.1.1.2
   Status codes: s suppressed, d damped, h history, * valid, > best, i-int
   Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop      Metric  LocPrf    Weight    Path
   *> 2.2.2.0/24  192.1.1.1     0              0     100 i
   * i            195.1.1.2     0       100       0     100 i
   ```
BGP Soft Configuration
By Michael Gordon

BGP soft configuration enables policies to be configured and activated without resetting the BGP and TCP session. This allows the new policy to take effect without significantly affecting the network. Without BGP soft configuration, BGP is required to reset the neighbor TCP connection in order for the new changes to take effect. This is accomplished using the `clear ip bgp` command.

There are two types of BGP soft reconfiguration: outbound reconfiguration which will make the new local outbound policy take effect without resetting the BGP session, and inbound soft reconfiguration, which enables the new inbound policy to take effect.

The problem with inbound reconfiguration is that in order to generate new inbound updates without resetting the BGP session, all inbound updates (whether they are accepted or rejected) need to be stored by the router. This is memory intensive, and wherever possible it should be avoided.

To avoid the memory overhead needed for inbound soft reconfiguration, the same outcome could be achieved by doing an outbound soft reconfiguration at the other end of the connection. Outbound soft reconfiguration can be triggered with the following command: `clear ip bgp [ /address] soft out` For inbound soft reconfiguration, an additional router command needs to be added before a soft reconfiguration can be issued. This command tells the router to start storing the received updates: `neighbor [address] soft-reconfiguration inbound` Inbound soft reconfiguration can than be triggered with the following command: `clear ip bgp [ /address] soft in`

**Regular Expressions**

In the previous section, we looked at identifying routes based on IP address. In this section we will use regular expressions to identify routes based on AS path information. A regular expression is a pattern to match against an input string. When a regular expression is created, it specifies the pattern that a string must match.

The following is a list of keyboard characters that have special meaning when used in regular expressions:

<table>
<thead>
<tr>
<th>Character</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>.</td>
<td>Match any character including white space.</td>
</tr>
<tr>
<td>Asterisk</td>
<td>*</td>
<td>Match zero or more sequences of the pattern.</td>
</tr>
<tr>
<td>Addition Sign</td>
<td>+</td>
<td>Match one or more sequences of the pattern.</td>
</tr>
<tr>
<td>Question Mark</td>
<td>?</td>
<td>Matches zero or one occurrences of the pattern.</td>
</tr>
<tr>
<td>Caret</td>
<td>^</td>
<td>Begins with.</td>
</tr>
<tr>
<td>Dollar Sign</td>
<td>$</td>
<td>Ends with.</td>
</tr>
<tr>
<td>Underscore</td>
<td>_</td>
<td>Match the following.</td>
</tr>
<tr>
<td>Brackets</td>
<td>[ ]</td>
<td>Match a single value in range.</td>
</tr>
<tr>
<td>Hyphen</td>
<td>-</td>
<td>Separates the endpoints of a range.</td>
</tr>
</tbody>
</table>
Disabling Route Filtering based on Network Number

In this procedure, we will disable route filtering from the previous exercise.

RouterA(config)#router bgp 100
RouterA(config-router)#no neighbor 192.1.1.2 distribute-list 2 out
RouterA(config-router)#no neighbor 193.1.1.2 distribute-list 2 out

Filtering Based on AS Path

For this exercise, let's configure a regular expression in conjunction with a filter list on RouterC that will prevent any network that passes through AS 300 from being sent via BGP to RouterD.

Filtering routes based on AS path information can very useful when all routes from a particular AS need to be filtered. If filtering based on AS path was not used, the administrator would have to list each route one by one or potentially filter on a prefix. AS path filtering provides an efficient alternative to this.

In order to filter routes based on AS path information, we need to identify the AS path based on the defined regular expression and apply this to a BGP neighbor through a filter list:

1. Define the regular expression to deny any route that passed through AS300.

   RouterC(config)#ip as-path access-list 1 deny _300_
   (deny any route that passes through AS300)

   RouterC(config)#ip as-path access-list 1 permit . *

   Use the show ip bgp regexp command to see what routes the regular expression matches. The following is the output from the command. Note that network 2.0.0.0 is the only route that matches the regular expression (_300_). This command is very useful in verifying that the regular expression covers the routes to which you intend.

   RouterC# show ip bgp regexp _300_

   RouterC#sh ip bgp regexp _300_
   BGP table version is 13, local router ID is 195.1.1.2
   Status codes: s suppressed, d damped, h history,* valid,>best,i- int
   Origin codes: i - IGP, e - EGP, ? - incomplete

   Network       Next Hop    Metric  LocPrf Weight  Path
   * i1.1.1.0/24 194.1.1.2     0     100    0   100 300 400 i
   *>            193.1.1.1     0              0   100 300 400 i

2. Apply the filter list to BGP neighbor 195.1.1.1.

   RouterC(config)#router bgp 200
   RouterC(config-router)#neighbor 195.1.1.1 filter-list 1 out
In order for the changes to take effect, the BGP neighbor must be reset. To do this, use the command `clear ip bgp *`. This causes the TCP session between neighbors to be reset, restarting the neighbor negotiations from scratch and invalidating the cache.

```
RouterC# clear ip bgp *
```

Display the AS path access list on RouterC with the command `show ip as-path-access-list`. The following is the output from the command. This command is very useful in quickly determining what strings will be permitted or denied.

```
RouterC# sh ip as-path-access-list
 AS path access-list 1
 deny _300_
 permit .-*
```

Display the BGP filter list configured on RouterC with the command `show ip bgp filter-list 1`. The following is the output from the command. This command shows which of the routes conform to a specified “filter list” and therefore will be passed.

```
RouterC# show ip bgp filter-list 1

BGP table version is 5, local router ID is 195.1.1.2
Origin codes: i - IGP, e - EGP, ? - incomplete

Network       Next Hop  Metric LocPrf Weight Path
* i2.2.2.0/24 194.1.1.2     0    100       0   100 i
*>            193.1.1.1     0              0   100 i
```

Display the BGP table on RouterD with the command `show ip bgp`. The following is the output from the command. Notice that the route to network 1.0.0.0 via RouterC is no longer present in the routing table.

```
RouterD# sh ip bgp
BGP table version is 14, local router ID is 4.4.4.4
Origin codes: i - IGP, e - EGP, ? - incomplete

Network       Next Hop  Metric  LocPrf Weight Path
*>i1.1.1.0/24 194.1.1.2   0     100     0      100 300 400 i
*>i2.2.2.0/24 194.1.1.2   0     100     0      100 i
* i           195.1.1.2   0     100     0      100 i
```
The following is a list of the regular expressions and their significance:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>300</em></td>
<td>Match any routes that pass via AS 300.</td>
</tr>
<tr>
<td>_300$</td>
<td>Match any routes that originated in AS 300.</td>
</tr>
<tr>
<td>^300_</td>
<td>Only match routes received from AS 300.</td>
</tr>
<tr>
<td>^300$</td>
<td>Only match routes that originated from AS 300 and did not pass through any other AS.</td>
</tr>
<tr>
<td>*</td>
<td>All routes.</td>
</tr>
</tbody>
</table>

Looking for college courses?
We deliver.

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College's eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education.

Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

St. Petersburg College’s eCampus
http://e.spcollege.edu

Your time. Your place. Your future.
Introduction
ISIS is an interior gateway protocol (IGP) developed for routing the OSI stack. It's a link state protocol (uses SPF algorithm) that has become increasingly popular among top tier ISPs. ISIS was developed by the ISO. Incidentally, at about the same time the IAB (Internet Architecture Board) developed OSPF (hence the similarities). At the time there was a movement to adopt ISIS because it was assumed that IP would be replaced with the OSI suite. To aid in the transition, Integrated IS-IS was developed, which could route IP and CLNS.

*Radia Perlman was one of the chief designers of IS-IS.

IS-IS Addressing:
In the OSI world, routers are ISs (intermediate systems) and hosts are ESs (end systems). The NSAP Address (Network Service Access Point) is the address used to identify an IS. Unlike IP addressing, there will be a single NSAP address for the entire router.

ex. 07.0000.3090.c7df.00

The NSAP address is divided into 3 parts:

<table>
<thead>
<tr>
<th>Area Number</th>
<th>System ID</th>
<th>NSAP Selector (NSEL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifies area (used by L2)</td>
<td>node (used by L1)</td>
<td>service (like IP port)</td>
</tr>
<tr>
<td>variable length (min 2 bytes)</td>
<td>6 bytes</td>
<td>1 byte</td>
</tr>
</tbody>
</table>

How IS-IS Works:
Routers running IS-IS send IIH (IS-IS Hello packets) out all IS-IS enabled interfaces to discover neighbors and establish adjacencies. Routers report their adjacencies to a designated intermediate system (DIS). The DIS is responsible for conducting flooding over the LAN and maintaining synchronization. IS-IS does not have a backbone area like OSPF area 0. The IS-IS backbone is a contiguous collection of L2 capable routers, each of which can be in a different area. Routers can be either L1, L2 (like OSPF backbone routers) or L1/L2 (like OSPF ABRs). *Unlike OSPF, L1/L2 routers do not advertise L2 info to L1 routers. Therefore L1 routers have no knowledge of destinations outside of its own area.

L1 Routers knows the topology of only their own area. On routers running Integrated IS-IS, a default IP route will automatically be installed on L1 routers pointing to the nearest L2 or L1/L2 router. If there is more than one L2 router in the area, the L2 router will be selected based on lowest cost.
*Therefore, all IS-IS areas are stub areas.
L2 Routers knows about different areas, but do not have L1 information about their own area. *While in OSPF, area borders are marked with routers (ABRs), in IS-IS area borders are marked by the links connecting areas.

Configuring ISIS

You need at least 2620XM routers with IOS 12.2+

Objective

In this lab, you configure ISIS on three Cisco routers. First you configure loopback interfaces then you configure Ethernet interfaces. A default static route must be redistributed and you must control election of DIS for L1 and L2 routing. Connected networks in Denver must be considered as L2 and as L1 in Sydney.

Scenario

The backbone core of International Supplies located in Montreal consists of three routers. One is connected to the Internet (Montreal) and the two other are connected to US operational centers (Denver) and Asia/Pacific (Sydney). AP management wants to further implement ISIS in their local routers but US management wants to keep their independence and prefer to be considered as "external" in the routing process. You must demonstrate your ability to configure basic ISIS operation to match these requirements. Each loopback interface corresponds to one operational center.
**Step 1** Build and configure the network according to the diagram, but do not configure ISIS yet. A switch or a hub is required to connect the three routers via Ethernet. Use ping to verify your work and test connectivity between the Ethernet interfaces.

**Step 2** Configure the default route for Montreal:

```
Montreal(config)#ip route 0.0.0.0 0.0.0.0 Serial0/0
```

**Step 3** Activate CLNS routing in each router.

```
...(config)#clns routing
```

**Step 4** Define on which interface ISIS must be activated:

```
Montreal(config)#interface Ethernet0/0
Montreal(config-if)#ip router isis

Denver(config)#interface Ethernet0/0
Denver(config-if)#ip router isis

Sydney(config)#interface Loopback0
Sydney(config-if)#ip router isis
Sydney(config)#interface Loopback1
Sydney(config-if)#ip router isis
Sydney(config)#interface Loopback2
Sydney(config-if)#ip router isis
Sydney(config)#interface Loopback3
Sydney(config-if)#ip router isis
Sydney(config)#interface Ethernet0/0
Sydney(config-if)#ip router isis
```

**Step 5** Activate ISIS routing on each router, don't forget to distribute default-route and non-ISIS connected networks.

```
Montreal(config)#router isis
Montreal(config-router)#net 49.0001.0000.0000.0001.00
Montreal(config-router)#default-information originate
Montreal(config-router)#passive-interface Loopback0

Denver(config)#router isis
Denver(config-router)#net 49.0001.0000.0000.0002.00
Denver(config-router)#redistribute connected

Sydney(config)#router isis
Sydney(config-router)#net 49.0001.0000.0000.0003.00
Sydney(config-router)#passive-interface Loopback0
Sydney(config-router)#passive-interface Loopback1
Sydney(config-router)#passive-interface Loopback2
Sydney(config-router)#passive-interface Loopback3
```
**Step 6** After you enable ISIS routing protocol, verify its operation using show commands. Several important show commands can be used to gather ISIS routing process and CLNS addressing.

First, issue the `show ip protocols` command on any of the three routers.

What is the difference between Denver and Sydney regarding the loopback interfaces?

If you issue the `show ip route` command on Montreal, you'll see the difference:

```
Montreal#show ip route
[.../*...]
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
C  192.168.129.0/30 is directly connected, Serial0/0
   10.0.0.0/24 is subnetted, 9 subnets
   i L1 10.1.11.0 [115/10] via 10.1.1.3, Ethernet0/0
   i L1 10.1.10.0 [115/10] via 10.1.1.3, Ethernet0/0
   i L1 10.1.9.0 [115/10] via 10.1.1.3, Ethernet0/0
   i L1 10.1.8.0 [115/10] via 10.1.1.3, Ethernet0/0
   C  10.1.1.0 is directly connected, Ethernet0/0
   i L2 10.1.7.0 [115/10] via 10.1.1.2, Ethernet0/0
   i L2 10.1.6.0 [115/10] via 10.1.1.2, Ethernet0/0
   i L2 10.1.5.0 [115/10] via 10.1.1.2, Ethernet0/0
   i L2 10.1.4.0 [115/10] via 10.1.1.2, Ethernet0/0
S*  0.0.0.0/0 is directly connected, Serial0/0
```

What is the difference between networks from Denver and from Sydney?

What is the default metric for the ISIS Route? _________________________

What is the administrative distance for ISIS? __________________________

Keep track of updates and changes at [http://www.spcollege.edu/star/cisco](http://www.spcollege.edu/star/cisco) Scroll to the bottom of the page and click on the “Lab Manual Edits.”
**Step 7** ISIS use the NSAP address to determine area number. The `show clns` and `show clns neighbors [details]` will allow you to discover the area, hold down timer and which address are used to identify the different routers.

Montreal#`show clns`
Global CLNS Information:
  1 Interfaces Enabled for CLNS
  NET: 49.00001.0000.0000.0001.00
  **Configuration Timer: 60, Default Holding Timer:300**, Packet Lifetime 64
  ERDPDU's requested on locally generated packets
  Intermediate system operation enabled (CLNS forwarding allowed)
  IS-IS level-1-2 Router:
  **Routing for Area: 49.0001**

Montreal#`show clns neighbors details`
<table>
<thead>
<tr>
<th>System Id</th>
<th>Interface</th>
<th>SNPA</th>
<th>State</th>
<th>Holdtime</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sydney</td>
<td>Fa0/0</td>
<td>0004.c052.1d40</td>
<td>Up</td>
<td>28</td>
<td>L1L2</td>
</tr>
<tr>
<td>IS-IS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Area Address(es):  49.0001</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>IP Address(es):  10.1.1.3*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Denver</td>
<td>Fa0/0</td>
<td>000a.b758.f2c0</td>
<td>Up</td>
<td>9</td>
<td>L1L2</td>
</tr>
<tr>
<td>IS-IS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Area Address(es):  49.0001</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>IP Address(es):  10.1.1.2*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uptime: 01:16:12</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Area:** ___________________________________________________________

**Timers:** _________________________________________________________

What kind of address is used to identify neighbor routers?

**Step 8** As ISIS is a link state protocol, we can discover the different databases generated by ISIS and CLNS routing operations:

Montreal#`show isis database`
IS-IS Level-1 Link State Database:
<table>
<thead>
<tr>
<th>LSPID</th>
<th>LSP Seq Num</th>
<th>LSP Checksum</th>
<th>LSP Holdtime</th>
<th>ATT/P/OL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Montreal.00-00</td>
<td>* 0x00000009</td>
<td>0x9F84</td>
<td>871</td>
<td>0/0/0</td>
</tr>
<tr>
<td>Denver.00-00</td>
<td>0x00000006</td>
<td>0x699C</td>
<td>531</td>
<td>0/0/0</td>
</tr>
<tr>
<td>Denver.01-00</td>
<td>0x00000009</td>
<td>0x6F4A</td>
<td>983</td>
<td>0/0/0</td>
</tr>
<tr>
<td>Sydney.00-00</td>
<td>0x0000000E</td>
<td>0x6F4A</td>
<td>983</td>
<td>0/0/0</td>
</tr>
</tbody>
</table>

IS-IS Level-2 Link State Database:
<table>
<thead>
<tr>
<th>LSPID</th>
<th>LSP Seq Num</th>
<th>LSP Checksum</th>
<th>LSP Holdtime</th>
<th>ATT/P/OL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Montreal.00-00</td>
<td>* 0x000000F</td>
<td>0x38A7</td>
<td>426</td>
<td>0/0/0</td>
</tr>
<tr>
<td>Denver.00-00</td>
<td>0x000000F</td>
<td>0x9FB1</td>
<td>522</td>
<td>0/0/0</td>
</tr>
<tr>
<td>Denver.01-00</td>
<td>0x0000008</td>
<td>0x2320</td>
<td>636</td>
<td>0/0/0</td>
</tr>
<tr>
<td>Sydney.00-00</td>
<td>0x000000E</td>
<td>0x4214</td>
<td>523</td>
<td>0/0/0</td>
</tr>
</tbody>
</table>

460
Montreal\#\texttt{show isis topology}

IS-IS paths to level-1 routers

\begin{tabular}{|c|c|c|c|c|}
\hline
System Id & Metric & Next-Hop & Interface & SNPA \\
\hline
Montreal & -- & & & \\
Denver   & 10  & Denver & e0/0 & 000a.b758.f2c0 \\
Sydney   & 10  & Sydney & e0/0 & 0004.c052.1d40 \\
\hline
\end{tabular}

IS-IS paths to level-2 routers

\begin{tabular}{|c|c|c|c|c|}
\hline
System Id & Metric & Next-Hop & Interface & SNPA \\
\hline
Montreal & -- & & & \\
Denver   & 10  & Denver & e0/0 & 000a.b758.f2c0 \\
Sydney   & 10  & Sydney & e0/0 & 0004.c052.1d40 \\
\hline
\end{tabular}

\textbf{Step 9} All link state protocols must always establish adjacencies and elect a broadcast network a router that is responsible for flooding LSP updates on the network.

To see the DR election process, just execute the following commands:

\begin{verbatim}
Montreal# debug isis adj-packets
IS-IS Adjacency related packets debugging is on
Montreal# clear clns neighbors
\end{verbatim}

You will notice that there is no "BDR" election but two DRs are elected: one for L1 operations another for L2 operations. In the following debug, Denver will fulfil these two roles.

\begin{verbatim}
01:38:32: ISIS-Adj:...
Rec L1 IIH from 000a.b758.f2c0 (Ethernet0/0), cir type L1L2, cir id 0000.0000.0002.01, length 1497
\textbf{New adjacency, level 1} for 000a.b758.f2c0
Sending L1 LAN IIH on Ethernet0/0, length 1497
Rec L1 IIH from 0004.c052.1d40 (Ethernet0/0), cir type L1L2, cir id 0000.0000.0002.01, length 1497
\textbf{New adjacency, level 1} for 0004.c052.1d40
Rec L2 IIH from 000a.b758.f2c0 (Ethernet0/0), cir type L1L2, cir id 0000.0000.0002.01, length 1497
\textbf{New adjacency, level 1 & 2} for 000a.b758.f2c0
Sending L2 LAN IIH on Ethernet0/0, length 1497
Sending L1 LAN IIH on Ethernet0/0, length 1497
Rec L1 IIH from 0004.c052.1d40 (Ethernet0/0), cir type L1L2, cir id 0000.0000.0002.01, length 1497
\textbf{L1 adj count 1} \hspace{1cm} (Confirmed with Sydney)
Adjacency state goes to Up
Run level-1 DR election for Ethernet0/0
\textbf{New level-1 DR 0000.0000.0001} on Ethernet0/0
Sending L1 LAN IIH on Ethernet0/0, length 1497
Rec L1 IIH from 000a.b758.f2c0 (Ethernet0/0), cir type L1L2, cir id 0000.0000.0002.01, length 1497
L1 adj count 2 \hspace{1cm} (Confirmed w/ Denver)
L2 adj count 1
Adjacency state goes to Up
Run level-1 DR election for Ethernet0/0
\end{verbatim}
New level-1 DR 0000.0000.0002 on Ethernet0/0  (Denver becomes DR-L1)

Rec L2 IIH from 000a.b758.f2c0 (Ethernet0/0), cir type L1L2, cir id 0000.0000.0002.01, length 1497
Run level-2 DR election for Ethernet0/0
No change
Sending L1 LAN IIH on Ethernet0/0, length 1497
Rec L1 IIH from 000a.b758.f2c0 (Ethernet0/0), cir type L1L2, cir id 0000.0000.0002.01, length 1497
Rec L2 IIH from 000a.b758.f2c0 (Ethernet0/0), cir type L1L2, cir id 0000.0000.0002.01, length 1497
Rec L2 IIH from 0004.c052.1d40 (Ethernet0/0), cir type L1L2, cir id 0000.0000.0002.01, length 1497
L1 adj count 1
New adjacency, level 1 & 2 for 0004.c052.1d40
Sending L2 LAN IIH on Ethernet0/0, length 1497
Rec L1 IIH from 000a.b758.f2c0 (Ethernet0/0), cir type L1L2, cir id 0000.0000.0002.01, length 1497
Rec L2 IIH from 000a.b758.f2c0 (Ethernet0/0), cir type L1L2, cir id 0000.0000.0002.01, length 1497
Rec L1 IIH from 0004.c052.1d40 (Ethernet0/0), cir type L1L2, cir id 0000.0000.0002.01, length 1497
L1 adj count 2
L2 adj count 2
Adjacency state goes to Up
Run level-1 DR election for Ethernet0/0
No change
Run level-2 DR election for Ethernet0/0
No change
...from now, only LS exchange with DR.

Step 10  Finally, we want to control which router becomes the DR for L1 and for L2.

Montreal#show clns interfaces
Ethernet0/0 is up, line protocol is up
   Checksums enabled, MTU 1497, Encapsulation SAP
   ERPDUs enabled, min. interval 10 msec.
   RDPDUs enabled, min. interval 100 msec., Addr Mask enabled
   Congestion Experienced bit set at 4 packets
   CLNS fast switching enabled
   CLNS SSE switching disabled
   DEC compatibility mode OFF for this interface
   Next ESH/ISH in 43 seconds
Routing Protocol: IS-IS
   Circuit Type: level-1-2
   Interface number 0x0, local circuit ID 0x1
   Level-1 Metric: 10, Priority: 64, Circuit ID: Denver.01 (L1 DR)
   Number of active level-1 adjacencies: 2
   Level-2 Metric: 10, Priority: 64, Circuit ID: Denver.01 (L2 DR)
   Number of active level-2 adjacencies: 2
   Next IS-IS LAN Level-1 Hello in 5 seconds
   Next IS-IS LAN Level-2 Hello in 1 seconds

In the above output, Denver is DR for L1 and L2 operations. We want to elect Sydney as DR for L1 operations and keep Denver as DR for L2 operations.
To obtain this result, type the following commands:

```text
Sydney(config)#interface Ethernet0/0
Sydney(config-if)#isis priority 120 level-1
Sydney(config-if)#isis priority 1 level-2

Denver(config)#interface Ethernet0/0
Denver(config-if)#isis priority 120 level-2
Denver(config-if)#isis priority 1 level-1
```

Check the result on all three routers:

```text
Montreal#show clns interfaces
Ethernet0/0 is up, line protocol is up
[.../...]
Routing Protocol: IS-IS
  Circuit Type: level-1-2
  Interface number 0x0, local circuit ID 0x1
  Level-1 Metric: 10, Priority: 64, Circuit ID: Sydney.01
  Number of active level-1 adjacencies: 2
  Level-2 Metric: 10, Priority: 64, Circuit ID: Denver.01
  Number of active level-2 adjacencies: 2
  Next IS-IS LAN Level-1 Hello in 2 seconds
  Next IS-IS LAN Level-2 Hello in 2 seconds
```

**Hungry for more???

You are almost there for the CompTIA Network+ and Security+ Certifications too!
For more information:

http://www.comptia.org/certification/default.aspx

Building successful careers for students everywhere…
Phase 1: Topology
1. Physically cable the routers as depicted in the corresponding diagram.
2. The frame relay cloud between R1, R2, and R3 should be partially meshed with no PVC between R1 and R2.
3. The link from R4 to R5 should be 2 Mbps with R5 providing clocking.
4. Each router should have only the necessary physical interfaces enabled up along with one loopback each.

Phase 2: IP Addressing
5. Address all interfaces as detailed above.
6. R3’s S0 must use subinterfaces, as detailed above.

Phase 3: CLNS Addressing
In this procedure, you will enable IS-IS on all routers and advertise NSAPs.
7. Use each router's Loopback address as the System ID.
8. Uses the following area IDs for each router:

<table>
<thead>
<tr>
<th>Router</th>
<th>Area ID 1</th>
<th>Area ID 2</th>
<th>Area ID 3</th>
<th>Area ID 4</th>
<th>Area ID 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>47.0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>47.0002</td>
<td></td>
<td>47.0001</td>
<td>47.0002</td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>47.0001</td>
<td></td>
<td>47.0002</td>
<td>47.0004</td>
<td>47.0004</td>
</tr>
<tr>
<td>R4</td>
<td>47.0004</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>47.0004</td>
</tr>
</tbody>
</table>
Phase 4: Basic ISIS Configuration

9. Using the `is-type level-1` command, configure all routers as Level 1 only.
10. Using the `isis priority` command, configure R3’s E0 as the DR for the segment between itself and R4.
11. Using the `isis hello interval` command, increase the hello timer interval on the segment connecting R3 and R4 to every 30 seconds.
12. Using the `isis metric` command, assign the metric 20 to R5’s Serial 0.

Phase 5: ISIS Security

13. Using the `isis password` command, an area password of CZarea should be configured on R3.
14. Using the `isis password` on adjacent interfaces, all adjacencies established should be authenticated with the password Czadj.

Phase 6: Completion Criteria

15. All routers must be able to successfully reach all other routers.

Phase 7: Monitoring IS-IS

Displaying the IP Routing Tables

All routing tables should appear as follows:

```
r1#sh ip route
Codes:C - connected,S - static,I - IGRP,R - RIP,M - mobile,B - BGP
       D - EIGRP,EX - EIGRP external,O - OSPF,IA -OSPF inter area
       N1-OSPF NSSA external type 1,N2- OSPF NSSA external type 2
       E1 - OSPF external type 1,E2-OSPF external type 2, E - EGP
       i-IS-IS,L1-IS-IS level-1,L2-IS-IS level-2,ia-IS-IS int ar.
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

   150.100.0.0/16 is variably subnetted, 9 subnets, 3 masks
 i L1    150.100.0.250/32 [115/30] via 150.100.0.1, Serial0.103
 i L1    150.100.0.251/32 [115/20] via 150.100.0.1, Serial0.103
 C       150.100.0.249/32 is directly connected, Loopback0
 i L1    150.100.0.252/32 [115/30] via 150.100.0.1, Serial0.103
 i L1    150.100.0.253/32 [115/40] via 150.100.0.1, Serial0.103
 i L1    150.100.0.208/28 [115/30] via 150.100.0.1, Serial0.103
 i L1    150.100.0.192/28 [115/20] via 150.100.0.1, Serial0.103
 C       150.100.0.0/29 is directly connected, Serial0.103
 i L1    150.100.0.8/29 [115/20] via 150.100.0.1, Serial0.103
```
r2# sh ip route

Gateway of last resort is not set

150.100.0.0/16 is variably subnetted, 9 subnets, 3 masks
C 150.100.0.250/32 is directly connected, Loopback0
i L1 150.100.0.251/32 [115/20] via 150.100.0.9, Serial0.203
i L1 150.100.0.249/32 [115/30] via 150.100.0.9, Serial0.203
i L1 150.100.0.252/32 [115/30] via 150.100.0.9, Serial0.203
i L1 150.100.0.253/32 [115/40] via 150.100.0.9, Serial0.203
i L1 150.100.0.208/28 [115/30] via 150.100.0.9, Serial0.203
i L1 150.100.0.192/28 [115/20] via 150.100.0.9, Serial0.203
C 150.100.0.8/29 is directly connected, Serial0.203

r3# sh ip route

Gateway of last resort is not set

150.100.0.0/16 is variably subnetted, 9 subnets, 3 masks
i L1 150.100.0.250/32 [115/20] via 150.100.0.10, Serial0.302
C 150.100.0.251/32 is directly connected, Loopback0
i L1 150.100.0.249/32 [115/20] via 150.100.0.2, Serial0.301
i L1 150.100.0.252/32 [115/20] via 150.100.0.193, Ethernet0
i L1 150.100.0.253/32 [115/30] via 150.100.0.193, Ethernet0
i L1 150.100.0.208/28 [115/20] via 150.100.0.193, Ethernet0
C 150.100.0.192/28 is directly connected, Ethernet0
C 150.100.0.0/29 is directly connected, Serial0.301
C 150.100.0.8/29 is directly connected, Serial0.302

r4# sh ip route

Gateway of last resort is not set

150.100.0.0/16 is variably subnetted, 9 subnets, 3 masks
i L1 150.100.0.250/32 [115/30] via 150.100.0.194, Ethernet0
i L1 150.100.0.251/32 [115/20] via 150.100.0.194, Ethernet0
i L1 150.100.0.249/32 [115/20] via 150.100.0.194, Ethernet0
i L1 150.100.0.252/32 [115/30] via 150.100.0.194, Ethernet0
C 150.100.0.253/32 is directly connected, Loopback0
i L1 150.100.0.208/28 [115/20] via 150.100.0.209, Serial1
C 150.100.0.208/28 is directly connected, Serial1
C 150.100.0.192/28 is directly connected, Ethernet0
i L1 150.100.0.0/29 [115/20] via 150.100.0.194, Ethernet0
i L1 150.100.0.8/29 [115/20] via 150.100.0.194, Ethernet0
r5# sh ip route

Gateway of last resort is not set

    150.100.0.0/16 is variably subnetted, 9 subnets, 3 masks
i L1  150.100.0.250/32 [115/50] via 150.100.0.210, Serial0
i L1  150.100.0.251/32 [115/40] via 150.100.0.210, Serial0
i L1  150.100.0.249/32 [115/50] via 150.100.0.210, Serial0
i L1  150.100.0.252/32 [115/30] via 150.100.0.210, Serial0
C     150.100.0.253/32 is directly connected, Loopback0
C     150.100.0.208/28 is directly connected, Serial0
i L1  150.100.0.192/28 [115/30] via 150.100.0.210, Serial0
i L1  150.100.0.0/29 [115/40] via 150.100.0.210, Serial0
i L1  150.100.0.8/29 [115/40] via 150.100.0.210, Serial0

Displaying IS-IS Adjacencies:
All IS-IS adjacency tables should appear as follows:

r1# sh clns is-nei
System Id    Interface State Type Priority Circuit Id Format
1501.0000.0251 Se0.103     Up   L1   0         00       Phase V

r2# sh clns is-nei
System Id    Interface State Type Priority Circuit Id Format
r3             Se0.203     Up   L1   0         01       Phase V

r3# sh clns is-nei
System Id    Interface State Type Priority Circuit Id Format
r2             Se0.302     Up   L1   0         00       Phase V
1501.0000.0249 Se0.301     Up   L1   64        00       Phase V
r4             Et0         Up   L1   64        r3.03    Phase V

r4# sh clns is-nei
System Id    Interface State Type Priority Circuit Id Format
r3             Et0         Up   L1   127       r3.03    Phase V
r5             Se1         Up   L1   0         00       Phase V

r5# sh clns is-nei
System Id    Interface State Type Priority Circuit Id Format
r4             Se0         Up   L1   0         00       Phase V

Take college courses on a new schedule--yours.
Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education. Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

http://e.spcollege.edu

St. Petersburg College’s eCampus. Your time. Your place. Your future.
Paper Lab: Routing Protocols

**Objective:**
To be able to compare and contrast between the routing protocols used so far in our studies: RIP, RIP version 2, IGRP, EIGRP, BGP and OSPF.
On your test you may see this as a drag and drop or even matching. In this lab I have created paper “exercises” to help “simulate” this as best as I can.

**Link State—Distance Vector—Hybrid**

Put each of the protocols into their “type” of routing protocol. Identify which algorithm is used for each.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Link State</th>
<th>Distance Vector</th>
<th>Hybrid</th>
<th>Alg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIP</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RIP version 2</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGRP</td>
<td></td>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIGRP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BGP</td>
<td>no</td>
<td></td>
<td>SPF</td>
<td></td>
</tr>
<tr>
<td>OSPF</td>
<td></td>
<td>no</td>
<td>SPF</td>
<td></td>
</tr>
</tbody>
</table>

Which protocol(s) would be best used or more likely used in each situation and why?

1. Your company is connecting to the Internet via an ISP.
2. You wish to have your subnet mask information sent along with routing information.
3. Your company is running nothing but CISCO equipment for networking.
4. You are working in a small company using older equipment from CISCO.
5. Your company is using CISCO equipment along with IBM, Nortel, and Bay networking equipment.
6. You are working in a company that seems to merge many times with other companies. They also like to “absorb” smaller companies by purchasing them.

Which protocols use autonomous system numbers in order to be configured? (circle all that apply)

RIP   IGRP   RIPv2   OSPF   BGP   EIGRP

Which protocols do not pass subnet mask information? (circle all that apply)

RIP   IGRP   RIPv2   OSPF   BGP   EIGRP

Which protocols pass the entire routing table? (circle all that apply)

RIP   IGRP   RIPv2   OSPF   BGP   EIGRP

What time interval for each protocol are updates/tables sent? (RIP 60, IGRP 90, etc)

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Updates</th>
<th>Invalid</th>
<th>Hold-down</th>
<th>Flush</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIP</td>
<td>30 seconds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RIP version 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGRP</td>
<td>270</td>
<td></td>
<td>670</td>
<td></td>
</tr>
</tbody>
</table>

469
<table>
<thead>
<tr>
<th>Protocol</th>
<th>Bandwidth</th>
<th>Reliability</th>
<th>Load</th>
<th>MTU</th>
<th>Delay</th>
<th>K-metrics</th>
<th>Hop count</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIP</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>RIPv2</td>
<td></td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGRP</td>
<td>No</td>
<td></td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIGRP</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>BGP</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>OSPF</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

**Enable routing**
If you type in:

```
router(config)#router rip
router(config-router)#network 172.16.1.1
```

then what will appear with a show run?

a. `router rip
tenwork 172.16.1.0`

b. `router rip
tenwork 172.16.1.1`

c. `router rip
tenwork 172.16.1.1 255.255.255.0`

d. `router rip
tenwork 172.16.1.0 255.255.255.0`

e. `router rip
tenwork 172.16.0.0`
If you type in:

```
router(config)#router igrp 38
router(config-router)#network 172.16.1.1
```

then what will appear with a show run?

a. `router igrp 38
   network 172.16.1.0`

b. `router igrp 38
   network 172.16.1.1`

c. `router igrp 38
   network 172.16.1.1 255.255.255.0`

d. `router igrp 38
   network 172.16.1.0 255.255.255.0`

e. `router igrp 38
   network 172.16.0.0`

If you type in:

```
router(config)#router eigrp 38
router(config-router)#network 172.16.1.1 255.255.255.0
```

then what will appear with a show run?

a. `router eigrp 38
   network 172.16.1.0`

b. `router eigrp 38
   network 172.16.1.1`

c. `router eigrp 38
   network 172.16.1.1 255.255.255.0`

d. `router eigrp 38
   network 172.16.1.0 255.255.255.0`

e. `router eigrp 38
   network 172.16.0.0`
Section Objective/Overview:
To learn how to create simple/basic wildcard masks for use with Access Control Lists.

A while back you learned about subnet masks. We use wildcard masks to instruct our devices to “only pay attention” to certain information. The easiest way I know to explain how to set up a wildcard mask for beginners is: a wildcard mask is usually the exact opposite of a subnet mask (in terms of binary one’s and zero’s). After you learn this then you will learn how to manipulate a wildcard mask a bit more (a bit...that’s a joke, get it?).

If we had a network 172.16.1.0/24 and wanted to use a routing protocol:

- With RIP, IGRP, EIGRP, BGP (with subnet mask):
  - network 172.16.1.0  255.255.255.0
  - let’s see that subnet mask in binary:
    - 11111111.11111111. 11111111.00000000

- With OSPF (which uses wildcard mask):
  - network 172.16.1.0  0.0.0.255
  - let’s see that wildcard mask in binary:
    - 00000000.00000000. 00000000.11111111

People confuse wildcard masks with subnet masks all the time. They are similar after all because they both are “masks” but they serve different purposes. A wildcard mask helps an access control list (acl) determine which IP addresses to implement the access control list commands upon (either to permit or deny something or a group of things). Think about the acl like a bouncer at a nightclub. The bouncer can determine from a list who is permitted entry, who is denied entry, or who to ignore. So who gets in? Pam, Matt, and Mike all can get in because they are on the bouncers “A” list to be permitted. Rich does not get in because he is on the bouncers list to be denied access...maybe he caused a bunch of problems or had a really bad hair cut, who knows? Now what about Dave? Whether he is permitted really depends upon the judgment of the bouncer. In our case the “bouncer” is our acl statements, so how we set them up will determine what happens to information not specifically mentioned in our statements. Before we get too in-depth with this let’s start off by taking some addresses and subnet masks and writing some really basic wildcard masks.
“Simple Wildcard Masks”

Turn the subnet mask into a very simple wildcard mask:

201.45.25.38/24

First, we need to break this condensed address into its address and subnet mask:

201.45.25.38 /24 which equals 255.255.255.0

Then we can convert the subnet mask into binary:

201.45.25.38 11111111.11111111.11111111.00000000

Notice how we have twenty-four contiguous (all together) one’s and compare that to our condensed address (/24). That is how we can condense our subnet masks. Since basic subnet masks are the opposite of wildcard masks we need to invert or change all “one’s” into “zero’s” and all “zero’s” into “ones.”

Subnet mask
From: 201.45.25.38 11111111.11111111.11111111.00000000
       ↓                                                                
To:    201.45.25.38 00000000.00000000.00000000.11111111
        ↓                                                                
Basic Wildcard Mask

Later when you learn more about acl’s you will learn that this mask will allow us to permit or deny the specific IP address 201.45.25.38 the ability to do something. But we would have to revert the basic wildcard mask back into decimal for our statement.

201.45.25.38 0.0.0.255
Unlike subnet masks you will not see this as a condensed address of any sort because wildcard mask bits do not necessarily have to be contiguous which makes it a whole new ballgame. Before we start doing that though, let me have you try writing some basic wildcard masks just like I did above.

Try writing some basic wildcard masks for these IP addresses and subnet masks:

(1) 198.23.145.17/24
(2) 204.17.18.19/27
(3) 123.67.2.231/16

Now let’s go over each of them:

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Subnet Mask</th>
<th>Un-Condensed Subnet Mask</th>
<th>Converted to Binary</th>
<th>Inverted Ones and Zeros</th>
<th>Converted Back to Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>198.23.145.17/24</td>
<td>/24</td>
<td>255.255.255.0</td>
<td>11111111.11111111.11111111.00000000</td>
<td>00000000.00000000.00000000.11111111</td>
<td>0.0.0.255</td>
</tr>
<tr>
<td>204.17.18.19/27</td>
<td>/27</td>
<td>255.255.255.224</td>
<td>11111111.11111111.11111111.11000000</td>
<td>00000000.00000000.00000000.00011111</td>
<td>255.255.255.224</td>
</tr>
<tr>
<td>123.67.2.231/16</td>
<td>/16</td>
<td>255.255.0.0</td>
<td>11111111.11111111.00000000.00000000</td>
<td>00000000.00000000.11111111.11111111</td>
<td>0.0.255.255</td>
</tr>
</tbody>
</table>

Let’s end this section by having you try some more on your own (use the worksheet on the next page and check your answers in the back when you are finished):

(1) 220.1.45.33/29
(2) 192.168.1.1/24
(3) 112.34.63.89/17
(4) 43.123.65.67/10
(5) 129.53.234.253/22
(6) 223.17.28.96/27
(7) 143.17.0.0/16

Section Summary:
In this section you learned how to write simple or basic wildcard masks. From here you will learn more on manipulating those wildcard masks to “fit” certain circumstances. As you know subnet mask bits must be all contiguous one’s but wildcard masks do not have to use contiguous ones. You’ll learn about that next and then how to apply all of this information when writing access control lists.
**Simple Wildcard Mask Worksheet**

<table>
<thead>
<tr>
<th>Pick out subnet mask</th>
<th>Un-condense subnet mask</th>
<th>Convert subnet mask to binary</th>
<th>Invert one’s and zero’s</th>
<th>Convert back into decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pick out subnet mask</td>
<td>Un-condense subnet mask</td>
<td>Convert subnet mask to binary</td>
<td>Invert one’s and zero’s</td>
<td>Convert back into decimal</td>
</tr>
<tr>
<td>Pick out subnet mask</td>
<td>Un-condense subnet mask</td>
<td>Convert subnet mask to binary</td>
<td>Invert one’s and zero’s</td>
<td>Convert back into decimal</td>
</tr>
<tr>
<td>Pick out subnet mask</td>
<td>Un-condense subnet mask</td>
<td>Convert subnet mask to binary</td>
<td>Invert one’s and zero’s</td>
<td>Convert back into decimal</td>
</tr>
<tr>
<td>Pick out subnet mask</td>
<td>Un-condense subnet mask</td>
<td>Convert subnet mask to binary</td>
<td>Invert one’s and zero’s</td>
<td>Convert back into decimal</td>
</tr>
<tr>
<td>Pick out subnet mask</td>
<td>Un-condense subnet mask</td>
<td>Convert subnet mask to binary</td>
<td>Invert one’s and zero’s</td>
<td>Convert back into decimal</td>
</tr>
</tbody>
</table>
“Kicking it up a notch”

Ok…so wildcard masks are not as simple as just “flipping” the subnet masks bits. In fact, as I said earlier, wild card masks really do not have anything in common with subnet masks other than the word “mask.” When we are writing the wildcard mask we do not use the subnet mask that much. Our wildcard mask actually is highly dependent upon the IP address and not so much the subnet mask. It makes sense if you think about it…we want to permit or deny an IP address not specifically a subnet mask. Let’s extend our last discussion to show how we actually develop wildcard masks based upon the combination of our IP address and subnet mask.

A nice, neat, simple rule: zero bits denote “exact” match bits…think of that little razor knife: an “exact-o” knife. Let’s dig in to an example:

CISCO might ask you to “write a wildcard mask to permit or deny only the host IP address of 172.16.2.34/24”

Can you see what they are asking? They want an exact match for a host IP address here. Let’s convert the IP address to binary:

\[
\begin{align*}
172.16.2.34/24 \\
172.16.2.34 & \quad 255.255.255.0 \\
10101100.00010000.00000010.00100010 & \quad 11111111.11111111.11111111.00000000
\end{align*}
\]

Since they want an exact match for all bits then the wildcard mask is filled in with zero’s (ok…so the bits conversion wasn’t needed but give me a break…you will see why we added this step in next…). In this specific case (when dealing with exact IP addresses) we did not need to use the subnet mask for writing our wildcard mask.

\[
\begin{align*}
10101100.00010000.00000010.00100010 \\
00000000.00000000.00000000.00000000
\end{align*}
\]

Therefore, when we convert this wildcard mask back to decimal we get a wildcard mask of 0.0.0.0 for our exact host match. The access control list statement would include the phrase “172.16.2.34 0.0.0.0”

IP address  Wildcard Mask

Now let’s have you try one…write a wildcard mask to permit or deny the host 45.23.67.123/8
It’s actually kind of a no-brainer now because you know any host would have the IP address given plus 0.0.0.0 for the wildcard mask. Therefore our answer is “45.23.67.123 0.0.0.0”

Next we will try writing a wildcard mask to permit or deny an entire subnet. You may want to use this for permitting only one subnet to have access to something (like letting only accounting people have access to the accounting server at corporate headquarters) Write a wildcard mask to permit or deny a entire subnet containing the IP address of 172.16.2.34/27 Can you see what they are asking? They want an exact match for the subnet containing the host IP address of “34” here. Let’s convert the IP address to binary:

10101100.00010000.00000010.00100010

Then, since we are working with subnets, let’s figure out the network, subnet, and host portions:

network.network.network.network.subnet host

Since they want an exact match for all network plus subnet bits then the wildcard mask is filled in with zero’s in the network and subnet portions and one’s in the host portion (we don’t care about the hosts since every possible host would be permitted or denied with this mask):

10101100.00010000.00000010.00100010

00000000.00000000.00000000.00011111

Therefore, when we convert this wildcard mask back to decimal we get a wildcard mask of 0.0.0.31 for our subnet wildcard mask (Use the bit bashing worksheet in Appendix A if needed). The portion of our access control lists would be written as:

The access control list statement would include the phrase “172.16.2.34 0.0.0.31” or even “172.16.2.32 0.0.0.31” to be more precise (the subnet number)

Let’s try a couple here:

1. “write a wildcard mask to permit or deny only the subnet address of 192.168.1.129/25”
2. “write a wildcard mask to permit or deny only the subnet address of 143.23.46.252/27”
3. “write a wildcard mask to permit or deny only the subnet address of 14.23.46.252/11”
4. “write a wildcard mask to permit or deny only the subnet address of 150.50.50.50/20”
5. “write a wildcard mask to permit or deny only the subnet address of 214.0.0.5/24”
Finally as we have mentioned and hinted a couple of times, unlike subnet masks, wildcard masks do not have to be contiguous (all zeros in a row)...we can mask out certain IP’s. Write a wildcard mask to permit or deny only the *odd numbered* IP’s in the entire subnet containing the IP address of 172.16.2.34/27 Let’s convert the IP address to binary: 10101100.00010000.00000010.00100010 Then let’s figure out the network, subnet, and host portions:

10101100.00010000.00000010.00100010
nnnnnnnn.nnnnnnnnn.nnnnnnnnn.sshhhhh

*network.network.network.* 

The access control list statement would include the phrase “172.16.2.34 0.0.0.30” This one can be confusing...later on when you learn about writing access control lists doing something like this will depend upon whether you are permitting or denying something. For now just realize the bits do not have to be contiguous.

*Section Summary:*

Well by now you have mastered the intricacies of writing the address and wildcard mask portion of an access control list. Now it’s time to start learning about acl’s themselves.

**Pro-Cote Inc.**

Specially Coatings Applicators  
877-588-4300

Certified Conductive and Static Dissipative Floor applications for every need:

- Facilities with highly sensitive electronic equipment
- Computer facilities
- Electronic manufacturing & assembly
- Facilities containing AGV lines
- Clean rooms
- Manufacturing/processing flammable or combustible materials
- Munitions plants

Pro-Cote Inc. is a proud installer of Valspar Flooring Systems  
www.pro-cote-usa.com

Pro-Cote Inc.  
P. O. Box 1766  
Largo, FL  33779  
Toll Free (877) 588-4300

478
Name: _____________________

Quiz 1: Access Control Lists and Wildcard Masks

True  False  1. All wildcard masks must use contiguous one’s.
True  False  2. Wildcard masks are opposites of subnet masks.
True  False  3. All subnet masks must use contiguous one’s.
True  False  4. Access control lists are always used in networks.
True  False  5. Subnet masks are always used when writing wildcard masks.

6. Write a wildcard mask that will mask the 192.168.1.0/24 network. We are looking for an exact match of the network and subnet portions only.
7. Write a wildcard mask that will mask the host portions of the 192.168.1.0/27 network. We are looking for an exact match of the host portions only.
8. Write a wildcard mask that will mask the odd IP addresses of the 172.16.23.0/16 network. We are looking for an exact match of the network, subnet and odd-numbered IP’s in the host portion.
9. Write a wildcard mask that will mask the last subnet of the 10.128.0.0/10 network.
10. Write a wildcard mask that will mask the second subnet of the 200.210.128.0/27 network.

Score: __________
Access Control List Theory

Equipment:
For this lab I used model 2611 routers with an IOS version of 12.0(13).

Background:
An access control lists (ACL) is a sequential collection of statements that control access to or from a network or subnet. The ACL statements are processed in the order in which they appear. There really is nothing magical about them...we just need to use them carefully and understand the logic of ACL’s. Too many times teachers or textbooks just glance quickly over them and really do not explain them well. In this lab you should gain a good foundation of ACL theory and then apply it later. ACL’s consume large amounts of resources since every single packet coming and going is compared against every single ACL statement. In this respect we want to use them sparingly. Large amounts of ACL statements are best left to firewall and security devices...if you use lots of ACL statements you are actually turning your router into a firewall device. Let’s look at the basics of ACL’s first. Creating and implementing ACL’s is a two-step process:

1. create the ACL
2. apply the ACL to an interface

You can write ACL’s for a variety of conditions and scenario’s, some of which we learned with our wildcard masks. In CCNA training you typically will learn about 3 of the basic ACL’s: Standard, Extended, and Named. Two of the other ACL’s you will learn about typically in CCNP training are Dynamic (a.k.a “Lock and Key”) and Reflexive.

(1) Creating an ACL:
We write our ACL’s at the configuration prompt because we are going to then apply it to an interface. All ACL statements begin with the phrase “access-list” so let’s go through an example using a 2611 router with an IOS of 12.0(13). Let’s do some basic command formatting research:

Router>en
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#access-list ?

Yeah, that question mark can answer a bunch of questions so be sure to use it whenever you can. What you will see is the information covered in table 1 (page 16). For now let’s just use the most basic ACL called “standard” and pick one number from its basic range of 1 to 99.

Router(config)#access-list 1 ?
deny Specify packets to reject
permit Specify packets to forward

Just as we have said we use these lists like a bouncer does to permit or deny something so we see it here. Let’s pick “permit” (I flipped a coin...heads for permit).
Router(config)#access-list 1 permit ?
  Hostname or A.B.C.D  Address to match
  any                  Any source host
  host                 A single host address

We see our command starting to grow…now we can select a hostname by IP address, use the word “any” or use the word “host” to signify a specific IP address will follow. Let’s pick “host” and write one for permitting host 192.168.1.1 into something.

Router(config)#access-list 1 permit host ?
  Hostname or A.B.C.D  Host address
Router(config)#access-list 1 permit host 192.168.1.1
Router(config)#

We can see our router took this command. When I did just host (with no question mark) it would not take it. Therefore we are at our most basic phase of our command. Of course we have more options but we will cover them later. Just to get you into the swing of using wildcard masks if we were to not use the word “host” but used just the IP address of 192.168.1.1 we would have needed to add the wildcard mask “0.0.0.0” to say we need an exact match of our host address.

Router(config)#access-list 1 permit 192.168.1.1 0.0.0.0

Next, we have to apply it to an interface. This tells the router how to act upon the information and from which interface. We can use it to let someone from Sales have access to the web but not the accounting department but, at the same time has the accounting department have access to both the web and sales department. Confused? Yeah…let’s just plow ahead and hope it “clicks” soon, if it doesn’t already.

Router(config)#interface serial0/0
Router(config-if)#ip access-group ?
  <1-199>      IP access list (standard or extended)
  <1300-2699>  IP expanded access list (standard or extended)
  WORD         Access-list name

Here we can see that we need our ACL number again…this ties us back into the acl statement we just wrote. You will learn more about this later.

Router(config-if)#ip access-group 1 ?
  in   inbound packets
  out  outbound packets

Once we have tied it back we need to tell the ACL grouping to work on either the incoming or outgoing packets. This is one of the most confusing topics for people to understand because the best answer here is “it depends.” It depends on which ACL, which router, which interface, how it is used and a whole bunch of other factors. There really is no hard and fast rule to “always use this.” Experience will teach you best. In the short term we’ll give you some hard and fast things for beginners. And that is it for
applying a very basic standard ACL to a router. Before we get more into it let’s do a few more examples and then let you look at behaviors of ACL’s.

Write a standard acl to permit host 172.16.1.2 (don’t worry about applying it to an interface for now):

```
Router>  
Router#  
Router(config)#  
```

You should have come up with one of the following:

```
Router(config)#access-list 1 permit 172.16.1.2 0.0.0.0  
or  
Router(config)#access-list 1 permit host 172.16.1.2  
```

1. Write an access control list to permit a host from the 192.168.1.14/24 network (don’t worry about applying it to an interface for now).
2. Write an access control list to deny a host from the 192.168.2.23/24 network (don’t worry about applying it to an interface for now).
3. Write an access control list to permit all odd numbered IP addresses in the 192.168.1.47/24 network (don’t worry about applying it to an interface for now).
4. Write an access control list to deny all even numbered IP addresses in the 192.168.1.47/24 network (don’t worry about applying it to an interface for now).
5. Write an access control list to permit the second subnet in the 192.168.1.1/27 subnet (don’t worry about applying it to an interface for now).

These exercises should help start to “bring it all together” with the wildcard masks. Let’s look more at the behavior of ACL’s.

(2) Behaviors of ACL’s:

In general all ACL’s follow some very basic behaviors on your router that you will need to know in order to understand them better:

1. ACL’s are sequentially processed
2. ACL’s are compared until a match is made…if no match is made then the packets are dropped and not processed.
3. There is an implicit “deny” statement at the end of every permit statement, BUT no implicit “permit” statement for every deny…watch out!
4. Place standard ACL’s as close to the destination as possible. For now use “out” with standard ACL’s on the interface (more on this later).
5. Place extended ACL’s as close to the source as possible. (The S’s do not go together) For now use “in” with extended ACL’s on the interface.
Let’s extend our earlier example of writing some ACL’s to permit or deny specific hosts to illustrate our points here. We’ll start with this example:

access-list 1 permit host 192.168.1.1 ← exact match? No go to next 1
access-list 1 permit host 172.16.1.1 ← exact match? No go to next 1
access-list 1 permit host 10.0.0.1 ← exact match? No go to next 1

```
check with ACL
```

incoming request 220.21.21.1

```
No other ACL’s request is denied; packet is dropped
```

Can you see a way to write our acl in a different manner that still achieves the same goal? I hope so. It would look like this:

```
access-list 1 permit 192.168.1.1 0.0.0.0
access-list 1 permit 172.16.1.1 0.0.0.0
access-list 1 permit 10.0.0.1 0.0.0.0
```

There is also a caveat to our example based upon our rules. We’ll go over that in a bit. And now another one which does match our access control list:

access-list 1 permit host 192.168.1.1 ← exact match? No go to next 1
access-list 1 permit host 172.16.1.1 ← exact match? Yes!

```
Entry allowed 😊
```

```
access-list 1 permit host 10.0.0.1
```

```
check with ACL
```

```
Entry allowed
```

incoming request 172.16.1.1

So you should have a good idea now that ACL’s are sequentially processed. Let’s see how interchanging “permit” and “deny” can change our results from our earlier example.
access-list 1 permit 192.168.1.1 0.0.0.0
access-list 1 permit 172.16.1.1 0.0.0.0
access-list 1 permit 10.0.0.1 0.0.0.0
access-list 1 deny any \(\text{Implied deny statement (not shown on router)}\)

Basically this access list is only permitting three IP addresses and denying all others. It is denying all others because at the end of every “permit” there is an implicit “deny” as shown in italics. But what happens if we flip the “permit” and “deny?” Let’s try it out:

access-list 1 deny 192.168.1.1 0.0.0.0
access-list 1 deny 172.16.1.1 0.0.0.0
access-list 1 deny 10.0.0.1 0.0.0.0
access-list 1 permit any \(\text{Nothing implied at all...does not exist}\)

Is this supposed to mean to deny these three IP addresses and allow everything else? No, in this access list you have denied access to everyone. You will have to specifically add the fourth line to deny those three addresses and allow everyone else:

access-list 1 deny 192.168.1.1 0.0.0.0
access-list 1 deny 172.16.1.1 0.0.0.0
access-list 1 deny 10.0.0.1 0.0.0.0
access-list 1 permit any

Are you starting to see how complicated these things can be yet? Heck, my head is spinning just from writing this thing. The last portion of this topic is placing access control lists. Remember our most basic rules in creating an ACL: first you create it then you apply it. Well now we will learn our trick on applying an acl to an interface. For the most part you should not have your “S’s” together. In other words your standard access control lists should be placed as close to your destination as possible and your extended access control lists should be places as close to your source as possible. Let’s go to the example again! When creating a standard acl on Router A to not allow access for the repair/Mfg group to have access to the Accounting/Admin network which interface in the diagram below to which interface should the acl be applied? All other subnets should have access to accounting.
Creating the ACL should be second hand to you by now:

```
RouterA> en
RouterA# config t
RouterA(config)# access-list 1 deny 192.168.1.128 0.0.0.63
RouterA(config)# access-list 1 permit any
```

In the first two lines we are navigating to our configuration mode. In the third line we are denying access to the repair network (subnet 128; IP addresses 129-190; Broadcast address 191):

```
<table>
<thead>
<tr>
<th>Deny 192.168.1.128 0.0.0.63</th>
</tr>
</thead>
<tbody>
<tr>
<td>11000000.10101000.00000000.00000000.10 000000 IP address</td>
</tr>
<tr>
<td>00000000.00000000.00000000.00000000.00 111111 Wildcard Mask</td>
</tr>
</tbody>
</table>
```

“Must be an exact Match” Don’t care

And now we must apply the acl:

```
RouterA> en
RouterA# config t
RouterA(config)# interface s0/1
RouterA(config-if)# access-group 1 out
```

We put the Standard ACL on router A (as close to the destination from the source—repair/mfg) using “out” with standard ACL’s. Don’t worry—in or out—still confuses a bunch of people…just keep to the rules and you will be fine. Don’t forget to always check with your instructor—they have to earn their pay somehow.

(3) Numbering ACL’s
Like so many things with computers, Access Control Lists are also numbered. We have different numbers for our different purposes, protocols, and types of ACL’s. Let’s look at those numbers now:
Section Summary

By learning and remembering these three things when using ACL’s (rules of creating, behavior, and numbering) you will be more successful in your endeavors with ACL’s on your routers. Too many times have I seen ACL’s choke a router down to a screeching halt simply because one of these rules was not followed. Let’s look at the theory of operation of each of these five types of ACL’s next.

Standard ACL’s

A standard ACL controls access using an *IP address or range of addresses*. The best way to figure these out is to dig right in and learn by doing! Let’s write a standard ACL to for hosts on the sales network to be denied access to the HR network, but allow them access to the engineering network and the WWW.

(1) Now lets create our ACL:

```bash
Router(config)#access-list 1 deny 192.168.40.0 0.0.0.255
Router(config)#access-list 1 permit any
```
Here we created our access-list and gave it the number 1 (tells us it is a standard ACL…see table 1). Then we put in our source IP addresses (in this case a network) and the wildcard mask. In this mask we wanted to exactly match the network and subnet portion and did not really care about the host portions. Therefore our mask became 0.0.0.255 (nnnnnnnn.nnnnnnnn.sssssss.sssssss). (2) Now we just need to do the second step: apply it to an interface. Since this is a standard ACL we want to apply it as close to the destination as possible using “out.” If we look at our diagram we can see that the Ethernet interface 0/0 is the closest to the destination network.

```
Router(config)#interface e0/0
Router(config-if)#ip access-group 1 out
```

**Extended ACL’s**

An extended ACL controls access to specific ports for IP addresses. Here we are doing basically the same thing but restricting access for something specific like ftp access, telnet access, or even icmp access. In a coming lab you will learn more about the commands. For now just kind of look this example over. Using our lab diagram again let’s write an ACL for the EGR network to have no (deny) telnet access to the HR network:

(1) Create the ACL:
```
Router(config)#access-list 100 deny tcp 192.168.30.0 0.0.0.255 any eq 23
Router(config)#access-list 100 permit ip any any
```

(2) Apply the ACL to an interface:
```
Router(config)#int e0/1
Router(config-if)#ip access-group 100 in
```

Using our lab diagram again let’s write an ACL for the EGR network to have no (deny) ability to ping (icmp) to the HR network:

(1) Create the ACL:
```
Router(config)#access-list 100 deny icmp 192.168.30.0 0.0.0.255
Router(config)#access-list 100 permit icmp any any
```

(2) Apply the ACL to an interface:
```
Router(config)#int e0/2
Router(config-if)#ip access-group 100 in
```
Named ACL’s
A named ACL uses a name instead of a number to do the same thing as standard or extended ACL’s. In a coming lab you will learn more about the commands. For now just kind of look this example over. Let’s write a named ACL to for hosts on the sales network to be denied access to the HR server, but allow them access to the marketing network and the WWW. Notice the changes in the prompt.

(1) Create the ACL:
```
Router(config)#ip access-list standard no_salesHR
Router(config-std-nacl)#deny 192.168.40.0 0.0.0.255
Router(config-std-nacl)#permit ip any
```

(2) Apply the ACL to an interface:
```
Router(config)#interface e0/0
Router(config-if)#ip access-group no_salesHR out
```

Dynamic ACL’s
The dynamic ACL is a combination of the above three except that it adds authentication. It is usually used when someone needs to have access to a router from home, like a network technician.

(1) Create the ACL:
```
RouterA(config)#access-list 101 permit tcp any host 192.168.1.67 eq telnet
RouterA(config)#access-list 101 dynamic authtest timeout 120
```

(2) Apply the ACL to an interface:
```
RouterA(config)#line vty 0
RouterA(config-line)#login local
RouterA(config-line)#autocommand access-enable timeout 5
```

Reflexive ACL’s
Heck we can use our bouncer example again for this one. Let’s say for a moment you are in that club dancing the night away and realize you ran out of money and need to run to your car for the credit card. Most clubs will stamp your hand to allow you to re-enter as long as you come back in a specific period of time, usually by the close of business. A reflexive ACL works in the same manner by stamping an outgoing packet to allow it to return within a specified period of time. Here’s an example that you will cover in a later lab on reflexive ACL’s. You should be able to tell it has elements of standard, extended, and named ACL’s within the configuration:

(1) Create the ACL:
```
BrFW(config)#ip access-list extended filterincoming
BrFW(config)#permit icmp 10.0.0.0 0.255.255.255 any reflect internaltraffic
BrFW(config)#deny icmp any any
BrFW(config)#evaluate internaltraffic
BrFW(config)#ip access-list extended filteroutgoing
BrFW(config)#permit icmp 10.0.0.0 0.255.255.255 any reflect internaltraffic
BrFW(config)#evaluate internaltraffic
```
Then we need to apply them to the interface:

```plaintext
BrFW(config)#int e0/0
BrFW(config-if)#ip access-group filterincoming in
BrFW(config-if)#ip access-group filteroutgoing out
```

*So What Have I Learned Here?*
In this lab you learned the intricacies of writing standard, extended and named access control lists. There is not a lot of material written about ACL’s so you just have to come up with your own ideas, test them, and learn from them…again…learning by doing. Now that we may have our “theories” down the next few labs will allow us to put ACL’s to work in our networks and Learn By DOING!

---

Looking for college courses?
We deliver.

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience *and* the quality instruction that has made SPC an international leader in online education.

Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

St. Petersburg College’s eCampus

[http://e.spcollege.edu](http://e.spcollege.edu)

*Your time. Your place. Your future.*
Quiz:
For the following ACL’s tell if the ACL is correct or incorrectly written for the instructions given. Use the lab diagram above as a guide. If the ACL is incorrect, then re-write it correctly to achieve its goals. (2 point each)

1. Write an ACL for the EGR network to be denied access to the Sales network using a standard ACL. Those crafty engineers like to mess with the Sales database files (like changing due dates of projects and stuff).

```
Router(config)#access-list 101 deny 192.168.10.0 0.0.0.255
Router(config)#int e0/2
Router(config-if)#ip access-group 101 out
```

Correct? __________________
Incorrect? __________________
What’s wrong? _________________________________________________
______________________________________________________________
______________________________________________________________
______________________________________________________________

2. Write an ACL for no one to have telnet use using an extended ACL.

```
Router(config)#access-list 10 deny tcp 192.168.0.0 0.0.255.255
Router(config)#access-list 10 permit tcp any any
Router(config)#int e0/0
Router(config-if)#ip access-group 101 in
Router(config)#int e0/1
Router(config-if)#ip access-group 101 in
Router(config)#int e0/2
Router(config-if)#ip access-group 101 in
```

Name: __________________________
Now let’s try to “free-hand” some ACL’s

3. Write a standard ACL to permit access from the EGR network (ip numbers 192.168.30.24, 192.168.30.37, 192.168.30.45 and 192.168.30.221) to the Sales network. Assume these are the IP addresses for supervisors. All other IP’s from the EGR should be denied access to the Sales network.

4. Write a named ACL to do the same thing.

5. Write an extended ACL to deny FTP access to everyone in the network.

6. Write a named ACL to allow only the EGR network to have www access.

7. Just for giggles lets allow the sales and HR network to have www access but not have dns access. In this manner they can get to web pages only if they know the specific dot-decimal address of the web page. Tee-hee, isn’t this a snort?

8. Write an extended ACL to allow only the HR people with odd numbered ip addresses to have the ability to use FTP.

The “Script Kiddie Cookbook” Available Mid-September 2004 at
http://www.lulu.com/learningbydoing

This book title is based upon the Anarchist Cookbook that came out in the 1980’s. In that tremendously popular book it showed you how to build bombs at home. Similarly the Script Kiddie Cookbook will show you step by step hacking and hacking methodology. You will go to jail if you use this information improperly!!!

Learn the basics about computer security with this book. This book assumes you know nothing about security and starts there. All tools and software used are freeware. Other books will build upon the materials in this book. Learn about the nuts and bolts of SPAM, port scanning, Knoppix tools, cookies, and bring it all together with exercises in hacking at legal gaming sites.

GET THE BOOK THE GOVERNMENT DOES NOT WANT YOU TO OWN!!!
**Standard Access Control Lists**

*Objective:*  
To implement a standard access control list on a simple network.

*Tools and Materials:*  
(4) Win 98 workstations  
(6) straight-through cables  
(2) routers (I used 2611 with 12.0(13))  
(1) DCE/DTE cable  
(2) switches (or one switch with 2 VLAN’s)

*Lab Diagram:*

```
ISP
s0/0
s0/1
company

Net. Admin  Sales1  EGR1  EGR2

IP 192.168.1.2/24  192.168.1.3/24  192.168.3.2/24 192.168.3.3/24
GW 192.168.1.1  192.168.1.1  192.168.3.1 192.168.3.1
```

**Addressing**

- **Router**  
  - S0/0 (DCE)  
    - n/a 192.168.2.1/24  
  - S0/1 (DTE)  
    - 192.168.2.2/24 n/a  
  - E0/0  
    - 192.168.1.1/24 n/a  
  - E0/1  
    - 192.168.3.1/24 n/a  
  - L0  
    - n/a 172.16.1.1/16

**Step-By-Step Instructions:**

1. Set up and cable the lab as shown. Use RIPv2 for routing. Enable file sharing on each computer.
2. Test ping from each workstation to each other and to the loopback interface. Troubleshoot as needed.
3. Make a folder on the desktop of each computer.
4. Make four text files and put one in each workstation. One message should be “This is my note for the one dot two workstation” that should be saved as 1dot2.txt and saved in that folder on the 192.168.1.2 workstation. It could look like this:

![Image of Notepad with text]

Repeat for each workstation. Put a shortcut for each desktop folder on each workstation. It should look like this:

![Image of desktop folder shortcuts]

5. Try to access the folders and text files on each workstation from each other workstation. It should work just fine and jim dandy.
6. Write a standard ACL to deny access for the host 192.168.1.2 to the 192.168.3.0 network. Step 1: create the ACL:
7. Step 2: apply the ACL to an interface. Since this is a standard ACL it should be placed nearest the destination as possible using “out.”

```
gates(config)#access-list 10 deny 192.168.1.2 0.0.0.0 OR 
gates(config)#access-list 10 deny host 192.168.1.2
```

8. From 192.168.1.2 try to ping 192.168.3.3. It should not work and be unreachable:

```
Pinging 192.168.3.3 with 32 bytes of data:
Reply from 192.168.1.1: Destination net unreachable.
Reply from 192.168.1.1: Destination net unreachable.
Reply from 192.168.1.1: Destination net unreachable.
Reply from 192.168.1.1: Destination net unreachable.
```

```
Ping statistics for 192.168.3.3:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Average round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

9. Use Windows explorer to find the computer. It won’t find it:

10. Try the shortcut to the 192.168.3.3 folder from 192.168.1.2. It won’t work. In fact the computer will appear to freeze and give you an icky message like this:
11. Try steps 8-10 again but from the 192.168.1.3 workstation. It should work fine because we only denied the host 192.168.1.2. Oh fudge! We forgot our pecking order with ACL’s…they are sequential and we need permits for denies. Let’s go add that in:

```
gates(config)#access-list 10 permit any
```

12. Now it should work fine…If you have any problems reboot the computers. Microsoft is quirky in small networks. I had to do it several times too. What the heck it may take some time but when you charge $100 an hour…who cares?

13. Ok…let’s play…let’s verify that we really got our “out” statement correct by changing it to “in” and see what happens.

```
gates(config)#int e0/1
 gates(config-if)#no ip access-group 10 out
 gates(config-if)#ip access-group 10 in
```

Everything will still work…drat! That is not what we wanted! We need to make those acl’s earn their keep. Let’s try again.

14. Let’s finish off this puppy with some show and debug commands.

```
gates#sh access-lists
 Standard IP access list 10
 deny 192.168.1.2
 permit any
```

This will show us, in brief, our standard access list statements. And, to the big kahuna:

```
gates#debug ip packet detail
 IP packet debugging is on (detailed)
gates#
18:32:05:   ICMP type=8, code=0
18:32:05:   IP: s=192.168.1.1 (local), d=192.168.1.2
 (Ethernet0/0), len 56, sending
18:32:05:   ICMP type=3, code=13
18:32:06:   IP: s=192.168.1.2 (Ethernet0/0), d=192.168.3.2
 (Ethernet0/1), len 60,
 access denied
18:32:06:   ICMP type=8, code=0
```
This will show us a bunch of things...here is what some of those codes mean:

Type 8 with code 0 is for an “ICMP echo.”
Type 3 with code 13 is for “administratively prohibited.”

So What Have I Learned Here?
In this lab you learned how to implement a standard ACL in a simple network. You also learned about the basic show and debug commands for use with ACL’s. Finally you got a refresher in basic networking with Microsoft. In the next lab you will work with extended ACL’s.

Hungry for more???
You are almost there for the CompTIA Network+ and Security+ Certifications too!
For more information:

http://www.comptia.org/certification/default.aspx

Building successful careers for students everywhere...
Practice Quiz/Lab:
Set up the following lab. Allow access from one department into all others but deny some access to others. Vary your approach. Use EIGRP.

Variations:
1. Try making this lab using only class “B” private IP addresses
2. Try making this lab using only class “A” private IP addresses
3. Try making this lab using all three classes.
4. Try making this lab using VLSM on each network, including the serial line. At first use only one private IP address class.
5. Then use multiple IP address classes.
6. Vary your routing protocol with multiple IP address classes and VLSM. You will have to use RIPv2 or EIGRP.
7. Use multiple routing protocols and redistribution.
8. Add in an Internet connection somewhere.
9. Repeat 1-7 with the Internet connection (put a loopback on the ISP router).
10. Make your Internet connection a passive interface so you do not broadcast your routes to the Internet.
11. Make the Internet connection a BGP line.
**Quiz:**
Design a network for a school that uses even-numbered IP addresses for teachers and odd-numbered IP addresses for students. This network is using 3 or more routers using a different routing protocol other than RIPv2. Use at least one of each of the private IP address classes throughout. The teachers should be able to access the students and the web but the students should only be able to access the web (and are denied access to teachers). Don’t forget the web connection. The border router should not advertise its routes to the web.

Network Design:

---

**Scoring:**

<table>
<thead>
<tr>
<th>Component</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>2</td>
</tr>
<tr>
<td>IP addressing</td>
<td>2</td>
</tr>
<tr>
<td>Cabling</td>
<td>2</td>
</tr>
<tr>
<td>ACL’s</td>
<td>2</td>
</tr>
<tr>
<td>Overall function</td>
<td>2</td>
</tr>
</tbody>
</table>

Total: ____________
**Extended Access Control Lists**

**Objective:**
To implement an extended access control list on a simple network.

**Tools and Materials:**
(4) Windows 98 workstations
(6) straight-through cables
(2) routers (I used 2611 with 12.0(13))
(1) DCE/DTE cable
(2) switches (or one switch with 2 VLAN’s)

**Lab Diagram:**

Addressing

<table>
<thead>
<tr>
<th>Router</th>
<th>company</th>
<th>ISP</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0/0 (DCE)</td>
<td>n/a</td>
<td>192.168.2.1</td>
</tr>
<tr>
<td>S0/1 (DTE)</td>
<td>192.168.2.2</td>
<td>n/a</td>
</tr>
<tr>
<td>E0/0</td>
<td>192.168.1.1</td>
<td>n/a</td>
</tr>
<tr>
<td>E0/1</td>
<td>192.168.3.1</td>
<td>n/a</td>
</tr>
<tr>
<td>L0</td>
<td>n/a</td>
<td>172.16.1.1/16</td>
</tr>
</tbody>
</table>

Keep track of updates and changes at [http://www.spcollege.edu/star/cisco](http://www.spcollege.edu/star/cisco) Scroll to the bottom of the page and click on the “Lab Manual Edits.”
**Step-By-Step Instructions:**

1. Clear the ACL’s on the router. Verify with “show run” after you clear them.

   ```
   gates(config)#no access-list 10
   gates(config)#int e0/1
   gates(config-if)#no ip access-group 10 out
   ```

2. Test ping from each workstation to each other and to the loopback interface. They should all work…sometimes you need to erase everything and start over. Do this if needed.

3. Write an extended ACL to deny icmp from 192.168.1.2 to everywhere. Step 1: create the ACL:

   ```
   gates(config)#access-list 138 deny icmp host 192.168.1.2 any
   gates(config)#access-list 138 permit ip any any
   ```

   Isn’t that weird how with extended ACL’s you have to use “ip any any” and with standard ACL’s you only needed “any?”

4. Step 2: apply the ACL to an interface. Since this is an extended ACL it should be placed nearest the source as possible using “in.”

   ```
   gates(config)#int e0/0
   gates(config-if)#ip access-group 138 in
   ```

5. From 192.168.1.2 try to ping 192.168.3.3. It should not work and be unreachable:

   ```
   C:\WINDOWS\Desktop>ping 192.168.3.3
   Pinging 192.168.3.3 with 32 bytes of data:
   Reply from 192.168.1.1: Destination net unreachable.
   Reply from 192.168.1.1: Destination net unreachable.
   Reply from 192.168.1.1: Destination net unreachable.
   Reply from 192.168.1.1: Destination net unreachable.
   Ping statistics for 192.168.3.3:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
   Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum = 0ms, Average = 0ms
   C:\WINDOWS\Desktop>
   ```

6. Try to ping from 192.168.1.2 to 192.168.3.2 and 172.16.1.1…both will not work.

7. Let’s assume this person will need to be able to ping to 172.16.1.1 but not to 192.168.3.0. So let’s modify our ACL a bit:

   ```
   gates(config)#no access-list 138
   **(you can see where a text editor would be helpful right?)
   gates(config)#access-list 138 deny icmp host 192.168.1.2 192.168.3.0 0.0.0.255
   gates(config)#access-list 138 permit icmp any any
   ```
Let’s look at our statement. We set up ACL 138 to deny ICMP from (source) host 192.168.1.2 to (dest) 192.168.3.0 (network) with a wildcard mask to match the network 0.0.0.255.

8. Now let’s try the show access lists again:

```
gates#sh access-list
Extended IP access list 138
   deny icmp host 192.168.1.2 192.168.3.0 0.0.0.255 (14 matches)
   permit icmp any any (4 matches)
gates#```

Aha! With extended ACL’s we can see the number of matches (or attempts) to get through our little router “mini-firewall.” We can even see from who it comes and how many times an attempt was made. Hmmm…almost like a protocol inspector. The debug ip packet details will show similar results.

9. Let’s add another ACL to stop 192.168.3.2 from telnetting to 172.16.1.1. But first let’s try to telnet to be certain it works. If it works you should see:

```
10. Now let’s create the extended ACL:

```
gates(config)#access-list 150 deny tcp host 192.168.3.2 any eq 23
gates(config)#access-list 150 permit tcp any any
```

11. And apply it to the interface:

```
gates(config)#int e0/1
gates(config-if)#ip access-group 150 in
```

12. Now telnet should work on 192.168.3.3 but not on 192.168.3.2. You will see this type of message if telnet is not working:
So What Have I Learned Here?
In this lab you learned how to implement an extended ACL in a simple network. You also learned about the basic show and debug commands for use with ACL’s. Finally you got a refresher in basic networking with Microsoft. In the next lab you will work with named ACL’s.

Take college courses on a new schedule--yours.
Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education. Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

http://e.spcollege.edu

St. Petersburg College’s eCampus. Your time. Your place. Your future.
Practice Quiz/Lab:
Set up the following lab. Allow access from one department into all others but deny some access to others using EXTENDED ACL’s. Vary your approach. Use EIGRP.

Variations using EXTENDED ACL’s:
1. Try making this lab using only class “B” private IP addresses
2. Try making this lab using only class “A” private IP addresses
3. Try making this lab using all three classes.
4. Try making this lab using VLSM on each network, including the serial line. At first use only one private IP address class.
5. Then use multiple IP address classes.
6. Vary your routing protocol with multiple IP address classes and VLSM. You will have to use RIPv2 or EIGRP.
7. Use multiple routing protocols and redistribution.
8. Add in an Internet connection somewhere.
9. Repeat 1-7 with the Internet connection (put a loopback on the ISP router).
10. Make your Internet connection a passive interface so you do not broadcast your routes to the Internet.
11. Make the Internet connection a BGP line.
12. Use a variety of permit and deny statements for the various options: ftp, telnet, http, dns, icmp etc. Be creative and have fun with it.
13. Try denying access to certain websites for certain users.
14. How would you stop users from accessing instant messaging?
Quiz:
In the last quiz you designed a network for a school that uses even-numbered IP addresses for teachers and odd-numbered IP addresses for students. This network is using 3 or more routers using a different routing protocol other than RIPv2. Use at least one of each of the private IP address classes throughout. The teachers should be able to access the students and the web but the students should only be able to access the web (and are denied access to teachers). Don’t forget the web connection. The border router should not advertise its routes to the web. Now I want you to deny ftp and telnet access to students. Plus I want you to “filter” out access to the following websites: www.2600.com, www.porn.com, www.adultfun.com, and www.xxx.com for all users.

Network Design:

Scoring: Design 2 points ___________
IP addressing 2 points ___________
Cabling 2 points ___________
ACL’s 2 points ___________
Overall function 2 points ___________
Total: ___________
Named Access Control Lists

Objective:
To implement a named access control list on a simple network.

Tools and Materials:
(4) Windows 98 workstations
(6) straight-through cables
(2) routers (I used 2611’s with 12.0(13))
(1) DCE/DTE cable
(2) switches (or one switch with 2 VLAN’s)

Lab Diagram:

<table>
<thead>
<tr>
<th>Addressing</th>
<th></th>
<th></th>
<th>ISP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router</td>
<td>company</td>
<td></td>
<td>192.168.2.1</td>
</tr>
<tr>
<td>S0/0 (DCE)</td>
<td>n/a</td>
<td></td>
<td>192.168.2.2</td>
</tr>
<tr>
<td>S0/1 (DTE)</td>
<td>192.168.2.2</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>E0/0</td>
<td>192.168.1.1</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>E0/1</td>
<td>192.168.3.1</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>L0</td>
<td>n/a</td>
<td></td>
<td>172.16.1.1/16</td>
</tr>
</tbody>
</table>

Step-By-Step Instructions:
1. Clear the ACL’s on the router. Verify with “show run” after you clear them.
2. Test ping from each workstation to each other and to the loopback interface.
3. Write a named ACL to deny icmp from 192.168.1.2 to everywhere. Include a named ACL to deny telnet from 192.168.3.2 to everywhere. Step 1: create the ACL:
4. Step 2: apply the ACL to an interface. Since this is an extended ACL it should be placed nearest the source as possible using “in.”

   gates(config)#int e0/0
   gates(config-if)#ip access-group no_ping in
   gates(config)#int e0/1
   gates(config-if)#ip access-group no_telnet in

5. From 192.168.1.2 try to ping 192.168.3.3. It should not work and be unreachable:

   C:\WINDOWS\Desktop>ping 192.168.3.3
   Pinging 192.168.3.3 with 32 bytes of data:
   Reply from 192.168.1.1: Destination net unreachable.
   Reply from 192.168.1.1: Destination net unreachable.
   Reply from 192.168.1.1: Destination net unreachable.
   Reply from 192.168.1.1: Destination net unreachable.
   Ping statistics for 192.168.3.3:
     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
     Approximate round trip times in milli-seconds:
       Minimum = 0ms, Maximum = 0ms, Average = 0ms

   C:\WINDOWS\Desktop>

6. Try to ping from 192.168.1.2 to 192.168.3.2 and 172.16.1.1…both will not work. Telnet to 172.16.1.1 should work on 192.168.3.3 but not on 192.168.3.2. You will see this type of message if telnet is not working:

   ![Connect failed message]

   So What Have I Learned Here?
   In this lab you learned how to implement a named ACL in a simple network. You learned we can replace standard and extended ACL’s with named ACL’s to help us out and to be able to use more than 100 ACL’s on a router (even though we don’t want to do that). In the next lab we will turn it up a bit by creating a protocol inspector on our router by using ACL statements.

506
Are you enjoying the materials? Well be on the lookout for some other manuals and textbooks on

http://www.lulu.com/learningbydoing and

http://www.spcollege.edu/star/cisco

The “Script Kiddie Cookbook” Available Mid-August 2004 at

http://www.lulu.com/learningbydoing

This book title is based upon the Anarchist Cookbook that came out in the 1980’s. In that tremendously popular book it showed you how to build bombs at home. Similarly the Script Kiddie Cookbook will show you step by step hacking and hacking methodology. You will go to jail if you use this information improperly!!!

Learn the basics about computer security with this book. This book assumes you know nothing about security and starts there. All tools and software used are freeware. Other books will build upon the materials in this book. Learn about the nuts and bolts of SPAM, port scanning, Knoppix tools, cookies, and bring it all together with exercises in hacking at legal gaming sites.

GET THE BOOK THE GOVERNMENT DOES NOT WANT YOU TO OWN!!!

In preparation for release in 12-18 months:
Basham, M.J. Learning by Doing: Acceptable Use Policy's and their implementation in networking.
Basham, M.J. Learning by Doing: CCNP Switching Essentials.
**Practice Quiz/Lab:**
Set up the following lab. Allow access from one department into all others but deny some access to others using NAMED ACL’s. Vary your approach. Use EIGRP.

![Network Diagram]

**Variations using NAMED ACL’s:**
1. Try making this lab using only class “B” private IP addresses
2. Try making this lab using only class “A” private IP addresses
3. Try making this lab using all three classes.
4. Try making this lab using VLSM on each network, including the serial line. At first use only one private IP address class.
5. Then use multiple IP address classes.
6. Vary your routing protocol with multiple IP address classes and VLSM. You will have to use RIPv2 or EIGRP.
7. Use multiple routing protocols and redistribution.
8. Add in an Internet connection somewhere.
9. Repeat 1-7 with the Internet connection (put a loopback on the ISP router).
10. Make your Internet connection a passive interface so you do not broadcast your routes to the Internet.
11. Make the Internet connection a BGP line.
12. Use a variety of permit and deny statements for the various options: ftp, telnet, http, dns, icmp etc. Be creative and have fun with it.
13. Try denying access to certain websites for certain users.
14. How would you stop users from accessing instant messaging?
Quiz:
In this quiz you will repeating what you did before but this time use NAMED ACL’s as efficiently as possible. In the last quiz you designed a network for a school that uses even-numbered IP addresses for teachers and odd-numbered IP addresses for students. This network is using 3 or more routers using a different routing protocol other than RIPv2. Use at least one of each of the private IP address classes throughout. The teachers should be able to access the students and the web but the students should only be able to access the web (and are denied access to teachers). Don’t forget the web connection. The border router should not advertise its routes to the web. Now I want you to deny ftp and telnet access to students. Plus I want you to “filter” out access to the following websites: www.2600.com, www.porn.com, www.adultfun.com, and www.xxx.com for all users.

Network Design:

Scoring:  
Design  2 points  __________
IP addressing  2 points  __________
Cabling  2 points  __________
ACL’s  2 points  __________
Overall function  2 points  __________

Total: __________
Dynamic Access Control Lists

**Background:**
Dynamic ACL’s are used to temporarily open a hole from the Internet into a private network. You may use this as a network administrator to gain access to a remote network at some point, however, there are security risks with doing this that are not covered here. In our example below if you are workstation C and wish to gain access to workstation A or B (which may also be a server or other networking device) then you could make a dynamic ACL to serve your needs.

Think more about how you could use this besides yourself…maybe you have a salesman on the road who may need access to a sales database; maybe you have some students who need to access a test server from home; or even having your club members be able to get access to a server with role-playing games on it. Dynamic ACL’s are nothing more than a modified extended ACL that uses authentication in the form of user names and passwords. That is why dynamic ACL’s are sometimes referred to as “Lock and Key.” The usernames and passwords act like locks and keys. Normally you would not use this unless you had a fancy authentication server called a RADIUS or TACACS+ to verify authentication but dynamic acl’s can be done as stand-alones on CISCO routers, but it is not recommended. Since most people do not have a RADIUS or TACACS+ server we’ll keep it simple here.

**Commands**
The basic command for setting up a dynamic ACL is:

```
Router(config)#access-list 101 dynamic mylist deny ip any any
```
Let’s go step through the help features and see what options are available with dynamic
ACL’s:

Router(config)#access-list 101 ?
deny Specify packets to reject
dynamic Specify a DYNAMIC list of PERMITs or DENYs
permit Specify packets to forward

Router(config)#access-list 101 dynamic ?
WORD Name of a dynamic list

Router(config)#access-list 101 dynamic mylist ?
deny Specify packets to reject
permit Specify packets to forward
timeout Maximum time for dynamic ACL to live

Router(config)#access-list 101 dynamic mylist timeout ?
<1-9999> Maximum time to live

Note: Timeout is in seconds and is not necessarily needed.

Router(config)#access-list 101 dynamic mylist timeout 120 ?
deny Specify packets to reject
permit Specify packets to forward

At this point you are just writing your ACL.

Lab: Let’s write a dynamic ACL to allow a workstation C (172.18.21.2) the ability to
telenet from home with a timeout of 120 seconds.

Router(config)#access-list 101 permit tcp any host 172.18.21.2 eq
telnet
Router(config)#access-list 101 dynamic mystuff timeout 120 permit
ip any any

Router(config)#line vty 0
Router(config-line)#login local
Router(config-line)#autocommand access-enable timeout 5

Router(config)#interface ethernet0/0
Router(config-if)#ip access-group 101 in
**Practice Quiz/Lab:**
Set up the following lab. Allow access from one department into all others but deny some access to others using DYNAMIC ACL’s. Vary your approach. Use EIGRP.

Variations using DYNAMIC ACL’s:
1. Try making this lab using only class “B” private IP addresses
2. Try making this lab using only class “A” private IP addresses
3. Try making this lab using all three classes.
4. Try making this lab using VLSM on each network, including the serial line. At first use only one private IP address class.
5. Then use multiple IP address classes.
6. Vary your routing protocol with multiple IP address classes and VLSM. You will have to use RIPv2 or EIGRP.
7. Use multiple routing protocols and redistribution.
8. Add in an Internet connection somewhere.
9. Repeat 1-7 with the Internet connection (put a loopback on the ISP router).
10. Make your Internet connection a passive interface so you do not broadcast your routes to the Internet.
11. Make the Internet connection a BGP line.
12. Use a variety of permit and deny statements for the various options: ftp, telnet, http, dns, icmp, etc. Be creative and have fun with it.
13. Try denying access to certain websites for certain users.
14. How would you stop users from accessing instant messaging?
**Quiz:**
In this quiz you will repeating what you did before but this time use DYNAMIC ACL’s as efficiently as possible. Set up a lab that will allow outside sales people to be able to telnet into your Sales server. Your design and addresses are up to you.

Network Design:

**Scoring:**
- Design: 2 points
- IP addressing: 2 points
- Cabling: 2 points
- ACL’s: 2 points
- Overall function: 2 points

Total: __________
**Objectives:**
To learn how a router can be set up as a mini-firewall using reflexive access control lists. Hopefully this will be a good transition from routers to firewalls.

**Tools and Materials:**
(2) routers
(3) switches (or one with 3 VLAN’s)
(6) straight-through cables
(1) DTE/DCE cable
(3) workstations

**Lab Diagram:**
Addressing:

<table>
<thead>
<tr>
<th>Router</th>
<th>ISP</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0/0</td>
<td>214.72.83.12/24 (DTE) 214.72.83.11/24 (DCE)</td>
</tr>
<tr>
<td>E0/0</td>
<td>10.0.1.1/24 50.0.1.1/24</td>
</tr>
<tr>
<td>E0/0</td>
<td>206.16.1.1/24 n/a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workstations</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>10.0.1.2/24</td>
<td>206.16.1.2/24</td>
<td>50.0.1.2/24</td>
</tr>
<tr>
<td>GW</td>
<td>10.0.1.1</td>
<td>206.16.1.1</td>
<td>50.0.1.1</td>
</tr>
</tbody>
</table>

**Background:**
In the last lab you learned that dynamic ACL’s are used to temporarily open a hole from the Internet into a private network. Reflexive ACL’s are the opposite: reflexive access control lists are used to temporarily open a hole from within the network into the Internet. We just learned about the standard, extended, and named access control lists (ACL’s) and how they work. We were told that too many ACL’s effectively turn the router into a firewall and severely degrades its overall performance. In fact routers and firewalls are very close in construction…they just have slightly different operating systems. Plus they cost about the same. Here are the front and rear views of a CISCO PIX Firewall.

Not too different huh? In this lab you will learn about a fifth type of access control list called a “reflexive” access control list. The reflexive access control list allows certain information out of a router port with a time to live counter. If the requested information returns before the timer expires then it is let back into that interface. Only information that originates from that interface is therefore allowed out and back in. Kind of like having a back stage pass huh? Take me to the green room! Typically firewalls allow private addresses (and address translation) on an “inside” portion of a network—totally shielded from the outside. Plus they have a “DMZ” zone which is not shielded from the outside...we tend to put our pesky sales people who are only contract employees out there. If you leap into the CISCO security certificate training then this lab provides a nice transition into the PIX firewall course.

**Step-By-Step Instructions:**
1. Cable the lab as shown.
2. Set up the basics and interfaces for each router. Use EIGRP or RIP version 2 for your routing protocol.
3. Put the IP addresses, masks, and gateways on the workstations.
4. Test ping from each workstation to the others. It should work just fine.
5. Let’s make an ACL to simulate a firewall:

```
BrFW(config)#access-list 1 permit 10.0.0.0 0.255.255.255
BrFW(config)#access-list 1 deny any
BrFW(config)#int e0/0
BrFW(config-if)#ip access-group 1 out
```

6. Test ping again. Workstation B and C should be able to ping each other but not to A. Workstation A should not be able to get past any interface on its router (request times out).

7. Even though that ACL works let’s remove that ACL and make a better one using reflexive ACL’s. This one will not only keep people out of the inside network but will not “imprison” the inside network. We will set it up to be able to use icmp to and from the inside network but anything outside of the network will not be able to ping into it (destination net unreachable).

```
BrFW(config)#ip access-list extended filterincoming
BrFW(config-ext-nacl)#permit icmp 10.0.0.0 0.255.255.255 any reflect internaltraffic
BrFW(config-ext-nacl)#deny icmp any any
BrFW(config-ext-nacl)#evaluate internaltraffic

BrFW(config)#ip access-list extended filteroutgoing
BrFW(config-ext-nacl)#permit icmp 10.0.0.0 0.255.255.255 any reflect internaltraffic
BrFW(config-ext-nacl)#evaluate internaltraffic
```

Then we need to apply them to the interface:

```
BrFW(config)#int e0/0
BrFW(config-if)#ip access-group filterincoming in
BrFW(config-if)#ip access-group filteroutgoing out
```

What we are doing here is creating two named ACL’s (filterincoming and filteroutgoing). Then we select which icmp addresses will be allowed (with wildcard mask) and then, in the same command, turn it into a reflexive ACL with the reflect command. Last in that command we create a temporary placeholder called “internaltraffic” which will hold our source information for the duration of the timer. When the packets come back we ask it to be evaluated with the information in our temporary placeholder “internaltraffic.” Finally, the reflexive ACL is applied to an interface. Notice how we used both in and out for our extended part…I told you earlier there are many uses of ACL’s and you would start learning more later.

8. Test ping again. Workstation B and C should be able to ping each other but not to A. Workstation A should now be able to ping everything.
So What Have I Learned Here?
In this lab you have learned the basics of firewall technology. As you progress in your studies you will learn more about techniques related to firewalls and security including content based access control, dynamic access control lists (lock and key), and AAA.

Practice Quiz/Lab:
Set up the following lab. Allow access from within the network to the Internet using Reflexive ACL’s. Use EIGRP.

Variations using REFLEXIVE ACL’s:
1. Try making this lab using only class “B” private IP addresses.
2. Try making this lab using only class “A” private IP addresses.
3. Try making this lab using all three classes.
4. Try making this lab using VLSM on each network, including the serial line. At first use only one private IP address class.
5. Then use multiple IP address classes.
6. Vary your routing protocol with multiple IP address classes and VLSM. You will have to use RIPv2 or EIGRP.
7. Use multiple routing protocols and redistribution.
8. Add in an Internet connection somewhere.
9. Repeat 1-7 with the Internet connection (put a loopback on the ISP router).
10. Make your Internet connection a passive interface so you do not broadcast your routes to the Internet.
11. Make the Internet connection a BGP line.
12. Use a variety of permit and deny statements for the various options: ftp, telnet, http, dns, icmp, etc. Be creative and have fun with it.
13. Try denying access to certain websites for certain users.
14. How would you stop users from accessing instant messaging?
Quiz:
In the last quiz you designed a network for a school that uses even-numbered IP addresses for teachers and odd-numbered IP addresses for students. This network is using 3 or more routers using a different routing protocol other than RIPv2. Use at least one of each of the private IP address classes throughout. The teachers should be able to access the students and the web but the students should only be able to access the web (and are denied access to teachers). Don’t forget the web connection. The border router should not advertise its routes to the web. Make a reflexive ACL to allow inside users to access the outside but not allow the outside to have access to the inside.

Network Design:

Scoring: 
- Design: 2 points
- IP addressing: 2 points
- Cabling: 2 points
- ACL’s: 2 points
- Overall function: 2 points

Total: ____________
Supplemental Lab or Challenge Activity:
1. Go out to CISCO and do some research on the features of the PIX firewall.
2. One problem with PIX firewall is they only work with IP. No IPX, Apple, XNS, etc. How could you get around that sort of problem?

Looking for college courses? 
We deliver.

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education.

Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

St. Petersburg College’s eCampus
http://e.spcollege.edu

Your time. Your place. Your future.
Fun with Access Control Lists

In this chapter we “let our hair down a bit” and have some fun with our ACL’s. It is my intention this chapter will help you learn a bit about how to apply your knowledge in very real world situations which, in turn, will enhance your learning. Not to mention this stuff is really kind of fun too.

Part I: Making a Protocol Inspector with ACL’s

Tools and Materials:
(2) workstations
(4) straight-through cables
(1) DCE/DTE serial cable
(2) routers
(2) switches (or 1 with 2 VLAN’s)

Lab Diagram:

Addressing:

<table>
<thead>
<tr>
<th>Router</th>
<th>goodguys</th>
<th>ISP</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0/0 (DCE)</td>
<td>n/a</td>
<td>220.100.50.1/24</td>
</tr>
<tr>
<td>S0/1 (DTE)</td>
<td>220.100.50.2/24</td>
<td>n/a</td>
</tr>
<tr>
<td>E0/0</td>
<td>192.168.1.1/24</td>
<td>172.16.1.1/16</td>
</tr>
</tbody>
</table>

Background:
A denial of service attack (DoS) occurs when disruption of services to legitimate users occurs. Denial of service attacks are gaining in number as evidenced in the media. Lately we have seen denial of service attacks that have crashed the networks of Yahoo, Ebay, Buy.com, CNN.com, E*Trade, ZDNet, Microsoft, and others. Initiating DoS
attacks are very simple…the tools are readily available over the Internet. To launch a
doS attack the attacker needs only a Linux/UNIX box with one of the following types of
programs: Trinoo, TFN, TFN2K, and Stacheldraht.

There are essentially three main categories of denial of service attacks: smurf,
fraggle, and sync attacks. A smurf attack (not the little blue guy) is caused by a flood of
icmp messages. A fraggle attack is caused by a flood of UDP packets. A sync attack is
cased by a flood of TCP packets. As we can see all three are closely related. We can
actually build a mini-protocol inspector to help us detect these three types of DoS attacks
when other equipment is not available.

Allow me to “set the stage…”

You are the network administrator in a small company…you do not have the big
bucks to buy those expensive protocol analyzers and network inspectors.
However, you have noticed your internet speeds, while guaranteed at T-1 for your
38 users, has actually been extremely slow. In fact, everyday it seems to get
slower. Also the computers have been randomly crashing and being disconnected
from the network with no clear indications why they have been doing that. You
are really starting to rack your brain over this one…

What is happening is your network is the victim of one of these denial of service attacks.
You can put a small acl which acts like a protocol inspector. Let’s see how.

Step-By-Step Instructions:
1. Set up and cable the lab as shown.
2. Add in our “mini-protocol inspector”

   goodguys(config)#access-list 100 permit icmp any any echo
   goodguys(config)#access-list 100 permit icmp any any echo-reply
   goodguys(config)#access-list 100 permit udp any any eq echo
   goodguys(config)#access-list 100 permit udp any any eq echo any
   goodguys(config)#access-list 100 permit tcp any any established
   goodguys(config)#access-list 100 permit tcp any any
   goodguys(config)#access-list 100 permit ip any any
   goodguys(config)#int s0/1
   goodguys(config-if)#ip access-group 100 in

The first two lines helps us monitor and record Smurf attacks, the next two helps
us monitor and record fraggle attacks, and the next two help us monitor for sync
attacks. Once we know where the attacks are coming from we can write other
acl’s to stop them (and to tell the authorities).

3. Let’s use the “evil” workstation to launch a vicious icmp flood to our goody two
shoes network using DOS

   ***Remember this is highly illegal…do not do this outside of lab conditions***

   Ping 192.168.1.2 -t -l 50000 (or try 500 then 5000)
4. Then let’s up it a bit by opening more DOS windows and slamming goody some more…three or four windows should suffice.
5. When we have had our fun we can use control+C to stop the ping storm.
6. Next we can use the show access-list command to look for matches (and potential attacks).

```
goodguys#show access-list
Extended ip access list 100
permit icmp any any echo (610 matches)
permit icmp any any echo-reply
permit udp any any eq echo
permit udp any any eq echo any
permit tcp any any established
permit tcp any any
permit ip any any (88 matches)
```

We have a good clue that an icmp flood (DoS) is occurring because of the large number of matches. Next we need to log our inputs and view the source ip addresses.
7. To start logging we just tack it on the end of the line with our matches. We don’t do it right away because it chews up valuable router resources. We save it for when we need it. First we copy and paste our acl to a notepad. Then we erase access-list 100 from our router:

```
goodguys(config)#no access-list 100
```

Then we make the changes to our acl in the notepad and then copy and paste it back into our router. Since we are interested only in the icmp section that will be all that is put back. In this manner we are conserving our resources. Since the icmp is throwing up a “red flag” with us we opt to log it and enable logging to run as the events happen:

```
goodguys(config)#access-list 100 permit icmp any any echo log-input
goodguys(config)#access-list 100 permit icmp any any echo-reply
goodguys(config)#logging buffered
```

The last line will let us see any notices as they occur…we will also see them in the log.
8. Next start ethereal on 192.168.1.2 and then start the pings again from 172.16.1.2.
9. Now we can repeat our ping storm, stop it, stop our ethereal and view our log:

```
goodguys(config)#sh log
```

You should see something like this:

```
goodguys#sh log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Console logging: level debugging, 49 messages logged
    Monitor logging: level debugging, 0 messages logged
```
Buffer logging: level debugging, 19 messages logged
Trap logging: level informational, 53 message lines logged
Log Buffer (4096 bytes):

00:27:14: %SYS-5-CONFIG_I: Configured from console by console
00:28:45: %SEC-6-IPACCESSLOGDP: list 100 permitted icmp 172.16.1.2 (Serial0/1 *HDLC*) -> 192.168.1.1 (8/0), 1 packet
00:29:11: %SEC-6-IPACCESSLOGDP: list 100 permitted icmp 172.16.1.2 (Serial0/1 *HDLC*) -> 220.100.50.2 (8/0), 1 packet
00:30:17: %SEC-6-IPACCESSLOGDP: list 100 permitted icmp 172.16.1.2 (Serial0/1 *HDLC*) -> 192.168.1.2 (8/0), 208 packets
00:32:29: %SEC-6-IPACCESSLOGDP: list 100 permitted icmp 220.100.50.1 (Serial0/1*HDLC*) -> 220.100.50.2 (8/0), 1 packet
00:34:09: %SYS-5-CONFIG_I: Configured from console by console
00:34:17: %SEC-6-IPACCESSLOGDP: list 100 permitted icmp 172.16.1.2 (Serial0/1 *HDLC*) -> 192.168.1.1 (8/0), 3 packets
00:36:04: %SYS-5-CONFIG_I: Configured from console by console
00:37:49: %SEC-6-IPACCESSLOGDP: list 100 permitted icmp 172.16.1.2 (Serial0/1 *HDLC*) -> 220.100.50.2 (8/0), 4 packets
00:37:50: %SYS-5-CONFIG_I: Configured from console by console
00:39:28: %SYS-5-CONFIG_I: Configured from console by console
00:41:18: %SEC-6-IPACCESSLOGDP: list 100 permitted icmp 172.16.1.2 (Serial0/1 *HDLC*) -> 192.168.1.2 (8/0), 35 packets
00:45:45: %SYS-5-CONFIG_I: Configured from console by console
00:46:18: %SEC-6-IPACCESSLOGDP: list 100 permitted icmp 172.16.1.2 (Serial0/1 *HDLC*) -> 192.168.1.2 (8/0), 247 packets

Can you see all the icmp packets below? Notice how most are fragmented.

10. So now we can stop our evil workstation (if only temporarily) using our log information:

goodguys(config)#access-list 1 deny host 172.16.1.2
goodguys(config)#access-list 1 permit any
goodguys(config)#int e0/0
goodguys(config-if)#ip access-group 1 out
11. Then when the evil workstation pings again the “destination is unreachable.” The evil workstation will change ip addresses or targets…hopefully the later.

**Supplemental Lab or Challenge Activity:**

1. Information from this lab was obtained from the CISCO website…I just made up new IP addresses, ACL’s numbers and added workstations. Go out to the website and find these papers:
2. Find out what “AAA” is from the CISCO website (NOT car insurance company you big goofs).
3. Investigate CISCO security certificate information from the website.
4. Can you use a debug to see those icmp packets? Try it.
5. Go out and research what trouble fragmented packets can cause on networking equipment.

**So What Have I Learned Here?**

Whew! This one can be rough. Don’t get too frustrated…ACL’s can cause problems and solve them too. I actually had to re-install my routing protocol after loading the ACL’s…stupid routers. Here you have learned about how you can apply access control lists in a little bit different manner. You have learned about denial of service attacks and icmp attacks in particular. Later, as you become more skilled, you can simulate tcp and udp attacks on your own private networks too.

---

**Part II: Hardening Security on CISCO routers with ACL’s**

A while back I saw this neat document on the National Security Agency website ([http://www.nsa.gov/snac/index.html](http://www.nsa.gov/snac/index.html)) and thought it would be fun to include this since it shows you some security basics on CISCO routers using ACL’s. I have actually re-written the text a bit so it makes a little more sense and have added in some insight into why you would use these.

(1) First it recommends logging access list port messages and doing so properly:

```text
router(config)#access-list 101 deny udp any range 1 65535 any range 1 65535 log  
router(config)#access-list 101 deny tcp any range 1 65535 any range 1 65535 log  
router(config)#access-list 101 deny ip any any log  
router(config)#access-list 101 permit ip any any
```

In this set you will basically being denying all udp and tcp ports access to the network first, logging any input (good for later record keeping) and then finally allowing them to access the network. They will never know they were being “checked out.” But you will have some really good traffic records, just in case.

(2) Next, the NSA guide recommends setting up your router to only allow your people to have access to your stuff. You never know who may have used a sniffer or protocol inspector and found out your addressing scheme and later tried to gain access with a faked address. I know it sounds like really common sense stuff, but every little bit helps
in security. Sometimes when I am at lectures or seminars and the speaker just happens to leave some tidbit of information about their IP address, mask, and gateway I just cannot help myself to write it down. I never use the information but you never know who just might use that information. Let’s look at the network design and then the recommended ACL:

```
router(config)#access-list 102 permit ip {your network} {wildcard} any
router(config)#access-list 102 deny ip any any log
router(config)#access-list 103 permit ip any {your network} {wildcard}
router(config)#access-list 103 deny ip any any log
router(config)#interface e0/0 {inside Ethernet}
router(config-if)#ip access-group 102 in
router(config-if)#interface e0/1
router(config-if)#ip access-group 102 out
router(config-if)#ip access-group 103 in
```

Similar to this you can block out any obviously “fake” or “spoofed” addresses that just cannot happen for good measure. Assume your network is 200.12.13.0/24. You would want to eliminate those obvious choices for having access from the Internet into your network. If you were a hacker then you could try these addresses to “spoof” or fake your way into a network…kind of like buying a fake id to get in.

```
router(config)#access-list 104 deny ip 200.12.23.0 0.0.0.255 any log
router(config)#access-list 104 deny ip 0.0.0.0 0.255.255.255 any log
router(config)#access-list 104 deny ip 10.0.0.0 0.255.255.255 any log
router(config)#access-list 104 deny ip 127.0.0.0 0.255.255.255 any log
router(config)#access-list 104 deny ip 169.254.0.0 0.0.255.255 any log
router(config)#access-list 104 deny ip 172.16.0.0 0.15.255.255 any log
router(config)#access-list 104 deny ip 192.168.0.0 0.0.255.255 any log
router(config)#access-list 104 permit ip any any
```

Why are these extended? They sound like standard acl’s to me…why we are not using extended may be because you never know when we want to add blocking some protocol, like tcp, to our list. At least we already have the option and do not have to add more acl’s and access-groups to our interfaces. Remember we want to use acl’s as sparingly as possible. Our first line denies anyone the ability to use our own addresses from the outside to gain access. The next bunch are just reserved addresses that someone may try to use to attempt access. It’s really not much but every hole we close is one less exploit for our networks.

(3) If your network does not need any multicasting then you would be wise to deny all multicasting. Watch all those users of instant messaging come screaming too!

```
router(config)#access-list 105 deny ip 224.0.0.0 15.255.255.255 any log
```

Ok. In this section we learned about some fun stuff we can do with access control lists. I hope this has given you a deeper understanding of how we can use these things to enhance our network security.
Are you enjoying the materials? Well be on the lookout for some other manuals and textbooks on http://www.lulu.com/learningbydoing and http://www.spcollege.edu/star/cisco

The “Script Kiddie Cookbook” Available Mid-September 2004 at http://www.lulu.com/learningbydoing

This book title is based upon the Anarchist Cookbook that came out in the 1980’s. In that tremendously popular book it showed you how to build bombs at home. Similarly the Script Kiddie Cookbook will show you step by step hacking and hacking methodology. You will go to jail if you use this information improperly!!!

Learn the basics about computer security with this book. This book assumes you know nothing about security and starts there. All tools and software used are freeware. Other books will build upon the materials in this book. Learn about the nuts and bolts of SPAM, port scanning, Knoppix tools, cookies, and bring it all together with exercises in hacking at legal gaming sites.

GET THE BOOK THE GOVERNMENT DOES NOT WANT YOU TO OWN!!!

In preparation for release in 12-18 months:
Basham, M.J. Learning by Doing: Acceptable Use Policy's and their implementation in networking.
Basham, M.J. Learning by Doing: CCNP Switching Essentials.
Access Control List Test

True or False (2 points each) Circle your choice.
1. TRUE FALSE When writing ACL’s it doesn’t matter which order the ACL statements are entered.
2. TRUE FALSE ACL’s are used in every network.
3. TRUE FALSE There is an implied permit for every deny statement in an ACL.
4. TRUE FALSE It is possible to mix standard, extended, and named ACL’s when grouping to an interface.
5. TRUE FALSE There is an implied deny at the end of a group of permit statements in an ACL.
6. TRUE FALSE There is an implied permit statement at the end of a group of permit statements in an ACL.
7. TRUE FALSE Using as few as ACL’s as possible is best.
8. TRUE FALSE Dynamic ACL’s typically allow access from the inside out.
9. TRUE FALSE You should always use “In” for Standard ACL grouping on interfaces.
10. TRUE FALSE Access Control Lists can be used to make a protocol inspector on a router.

Multiple Choice (3 points each) Circle your choice for the MOST correct answer (more than one may be correct).

11. Which of the following access control lists would you use to deny ftp access to a network?
   a. Standard Access Control List
   b. Extended Access Control List
   c. Named Access Control List
   d. A and/or B
   e. B and/or C

12. Which of the following access control lists would you use to deny access to a specific user to a network?
   a. Repulsive Access Control List
   b. Standard Access Control List
   c. Extended Access Control List
   d. A and/or B
   e. B and/or C

13. Which of the following access control lists would you use to permit web access to a specific group of users on a network?
   a. Extended Access Control List
   b. Web Access Control List
   c. Standard Access Control List

527
d. None of the above

14. Which of the following access control lists would you use to deny telnet access to a network?
   a. Telnet Access Control List
   b. Standard Access Control List
   c. Extended Access Control List
   d. Nominal Access Control List
   e. None of the above

15. Which of the following ACL ranges is used for Appletalk addresses? Circle all that apply.
   a. 1-99
   b. 100-199
   c. 600-699
   d. 1300-1999
   e. 2000-2699

16. Which of the following ACL ranges is used for IP extended access lists? Circle all that apply.
   a. 1-99
   b. 100-199
   c. 600-699
   d. 1300-1999
   e. 2000-2699

17. Which of the following ACL ranges is used for IPX standard access lists? Circle all that apply.
   a. 1-99
   b. 100-199
   c. 600-699
   d. 1300-1999
   e. 2000-2699

18. Which of the following access group statements generally are used together for beginners writing access control lists? Circle all the two that apply.
   a. Standard-source-in
   b. Standard-source-out
   c. Standard-destination-in
   d. Standard-destination-out
   e. Extended-source-in
   f. Extended-source-out
   g. Extended-destination-in
   h. Extended-destination-out
19. What is the rule for “implicit” statements when writing access control lists?
   a. There is no implicit statements
   b. There is an implicit deny for every permit statement
   c. There is an implicit permit for every deny statement
   d. There is always an implicit statement
   e. None of the above

20. Which of the following types of access control lists are used to generate small “holes” for the user to access from the inside of a network to the outside of a network?
   a. Standard Access Control List
   b. Extended Access Control List
   c. Named Access Control List
   d. Dynamic Access Control List
   e. Reflexive Access Control List

Fix the acl statements (5 points each)
21. Router(config)#access-list 12 deny tcp host 192.168.1.1 any eq ftp
    Fixed:

22. Router(config)#access-list 101 deny ip 192.168.1.1 any
    Fixed:

23. Router(config-std-nacl)#deny 192.168.30.0 0.0.0.255 any eq http
    Fixed:

24. Router(config)#access-list 12 deny 192.168.12.0 255.255.255.0
    Fixed:

25. Router(config)#ip access group no_Sales out
    Fixed:

26. Router(config)#access list 101 permit 192.168.0.0 0.0.255.255 any eq ftp
    Fixed:
Write an acl statement (10 points each)

27. Write an access control list statement to protect the general security of the private IP classes. Be sure to include your network 220.23.24.25/27.

28. Write an access control list to deny access into your network by telnet, ftp, or tftp. Come up with your own network numbers and draw a diagram.
Access Control List’s

On these pages you will find EXACT access control lists that have been tested and verified in a laboratory setting (unlike your textbook).

Deny a specific source host the ability to ping a destination

Router-B(config)#access-list 10 deny host 192.5.5.2
Router-B(config-if)#ip access-group 10 out

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router A</td>
<td>Router B</td>
</tr>
<tr>
<td>E0 192.5.5.1/24</td>
<td>E0 219.17.100.1/24</td>
</tr>
<tr>
<td>S0 201.100.11.1/24</td>
<td>S1 201.100.11.2/24</td>
</tr>
<tr>
<td>W0 192.5.5.2/24</td>
<td>W2 219.17.100.2/24</td>
</tr>
</tbody>
</table>

Q. Why do we use out?
A. Think of it this way...we are not pinging into our destination, but leaving (going out) during the ping operation from the source.

Q. Why do we put the acl on router B?
A. This operation deals with a standard acl. In this evolution we put our acl close to the destination. Remember: the s’s do not go together (source and standard)

Q. What happened to the 0.0.0.0 for the wildcard mask?
A. We use the word “host” instead.
To permit only workstation 192.5.5.2, change the “deny” to “permit.” Test it by using permit host 192.5.5.3.
Deny an entire subnet source the ability to ping a destination host

Router-B(config)#access-list 10 deny 192.5.5.2 0.0.0.255
Router-B(config-if)#ip access-group 10 out

Q. Why did we need a wildcard mask?
A. We needed it in place of the word “host.”

Q. How do we format our wildcard mask?
A. We use EXACTO! The network and subnet portion need to be an exact match, so we use a 0 in the mask...since we do not care about any host portion we put 1's or "255."

Q. Why did we put the ACL on router B?
A. We keep our Standard lists close to the Destination...we keep the S’s apart.
Deny telnet access from a workstation through the ethernet interface on a router

Router-A(config)#access-list 100 deny tcp 192.5.5.2 .0.0.0.0 any eq 23
Router-A(config-int)#ip access-group 100 in

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>E0</td>
<td>E0</td>
</tr>
<tr>
<td>S0</td>
<td>S1</td>
</tr>
<tr>
<td>W0</td>
<td>W2</td>
</tr>
</tbody>
</table>

In our address the first two octets must match exactly, the last two do not matter.
Lab set up notes:
You must also set up the source and destination workstations, complete with tcp/ip address, subnet mask, and gateway address, in order for this to work properly.

Hint: set up the network and test ping (or telnet, ie.) from source to destination. THEN implement the ACL. If your ACL is working correctly, then you should receive a “Destination host/network unreachable error” message.
Router Security and Extended ACL’s

This lab continuously gets raves from my students…they all love it!!! So here is the scenario: You will be setting up your border router. The instructor will be on a computer and will be “hacking” into your router using various methods (such as FTP, Telnet, etc.). Don’t worry…the instructor won’t be killing your config, just adding some description commands or changing your passwords. Your job will be to get your configurations set up and working properly, pinging and pulling HTTP through the BR (border router) and (if applicable) the IR (inside router). Once you can do that THEN start putting the ACL’s in your router. I have two different setups: one if you have a dual Ethernet router and one if you only have single Ethernet routers; either way, the end result is the same.

For this lab I used a Cisco 2611 with IOS 12.x.x(T) and, on the other setup, I used two Cisco 2610’s with IOS 12.x.x.(T).

Set up #1:

```
192.168.1.10/24
```

```
br config
E0/0 static ip 192.168.1.1/24
E0/1 192.168.1.1/24 n/a
S0/0 n/a 192.168.10.1/30 dce
S0/1 n/a n/a 192.168.10.2/30 dte
```

Step 1 in this lab is for you to find out the ip address that is being assigned statically or dynamically to your classroom pc. That is the “static” ip address that we will be using to connect to the outside interface of the border router (br). Use the GUI TCP/IP properties or the dos prompt with the **ipconfig** command. Don’t forget to record the gateway too.

```
Next, I want you to set up the border router using the stock config on the next page. Don’t just type it in, I want to get you used to saving your config’s and dumping from a text file. Type these commands into a text file, then copy and paste them in from notepad or word into the hyperterminal session.
```
Set up #1 dump configs:

br:
en
config t
hostname br
enable secret cisco
enable password class
line con 0
exec-t 0 0
logging syn
exit
line vty 0 4
login
password matt
exit
int e0/1
ip address 192.168.1.1 255.255.255.0
no shut
exit
router eigrp 38
network 192.168.1.1
redistribute static
exit

Set up #2 dump configs:

br:
en
config t
hostname br
enable secret cisco
enable password class
line con 0
exec-t 0 0
logging syn
exit
line vty 0 4
login
password matt
exit
int s0/1
ip address 192.168.10.2 255.255.255.0
description DTE to ir s0/0
no shut
exit
router eigrp 38
network 192.168.10.2
redistribute static
exit

ir:
en
config t
hostname br
enable secret cisco
enable password class
line con 0
exec-t 0 0
logging syn
exit
line vty 0 4
login
password matt
exit
int s0/0
ip address 192.168.10.1 255.255.255.0
no shut
description DCE to br s0/1
clock rate 56000
exit
int e0/0
ip address 192.168.1.1 255.255.255.0
no shut
router eigrp 38
network 192.168.10.1
network 192.168.1.1
redistribute static
exit

After dumping in your config the next thing you need to do is add your static ip address on the outside interface of your border router (don’t forget the no shut command too). Since we do not want to make it a practice to pass our routing information over the net we need to make a static route from the border router to the web. You can add the passive interface command later but leave it off for now. Add one thing, test, and then move on to the next.

You should now be able to ping the original workstation gateway and beyond. If you cannot, then you need to enable web services on your router. In later IOS’s this is enabled by default with the ip http-server global command:

```
router(config)#ip http-server
```
Can you browse the web now? Good! Now we are ready to move along. Tell your instructor you have completed the set up and give them the static ip address. Don’t even try giving them an incorrect one because the instructor will be actively port scanning the subnet for active ip addresses and netbios information. They may even begin launching some DDos against you with the WHAX or Knoppix OS’s if you give them the wrong information! Be nice, play nice, everything is nice. The instructor will then, whether you know it or not, hack into your router using a variety of methods; FTP, Telnet, etc. You next goal is to write an extended ACL that does not allow them to come back into your router. By the end of class you should have locked down a variety of entry ports into your router. Fun, isn’t it?

**Double-Dog Dare Challenge Lab:**
Ok, these are not for the faint at heart:

**Variant #1:**
Add port address translation with address overload to the border router.

**Variant #2:**
Add a modem to the auxiliary port and tighten down that security.

**Variant #3: The “Triple Dog-Dare”**
Tighten down against using private ip addresses from the outside; reserved addresses from the outside; permit only traffic from valid inside ip addresses to the outside; disable: cdp, finger, bootp, ip source routing, multicasting, proxy arp, ip unreachable, ntp, snmp, and dns; prevent DOS TCP SYN, SMURF, DDos/Trinoo, Stacheldraht, Trinityv3, and Subseven attacks; and filter icmp messages inbound and outbound. Deny TCP and UDP for ports (except where noted): 1 (Tcpmux), 7 (echo), 9 (discard), 11 (Systat, TCP only), 13 (daytime), 15 (netstat, tcp only), 19 (chargen), 37 (time), 43 (whois, tcp only), 67 (bootp, udp only), 69 (tftp, udp only), 93 (subdup, tcp only), 111 (sunrpc), 135 (loc-srv), 137 (netbios-sn), 138 (netbios-dgm), 139 (netbios-ssn), 177 (xdmcp, udp only), 445 (netbios-ds, tcp only), 512 (rexec, tcp only), 515 (lpr, tcp only), 517 (talk, udp only), 518 (ntalk, udp only), 540 (uucp, tcp only), 1900 (Microsoft Upnp ssdp), 5000 (Microsoft Upnp ssdp), 2049 (nfs, udp only), 6000-6003 (X window system, tcp only), 6667 (irc, tcp only), 12345 (netbus, tcp only), 12346 (netbus, tcp only), and 31337 (back orifice). I triple dog-dare you. That should keep you busy for quite some time.
Building the Router Toolbox

access-list 101 deny ip 0.0.0.0 0.0.0.0 any log
access-list 101 deny ip 10.0.0.0 0.255.255.255 any log
access-list 101 deny ip 127.0.0.0 0.255.255.255 any log
access-list 101 deny ip 169.254.0.0 0.255.255.255 any log
access-list 101 deny ip 172.16.0.0 0.15.255.255 any log
access-list 101 deny ip 192.168.0.0 0.0.255.255 any log
access-list 101 deny ip 224.0.0.0 15.255.255.255 any log
access-list 101 deny tcp any any eq 27655 log
access-list 101 deny udp any any eq 31335 log
access-list 101 deny udp any any eq 27444 log
access-list 101 deny tcp any any eq 16600 log
access-list 101 deny tcp any any eq 65000 log
access-list 101 deny tcp any any eq 33270 log
access-list 101 deny tcp any any eq 39168 log
access-list 101 deny tcp any any eq 6711 6712 log
access-list 101 deny tcp any any eq 6776 log
access-list 101 deny tcp any any eq 6669 log
access-list 101 deny tcp any any eq 2222 log
access-list 101 deny tcp any any eq 7000 log
access-list 101 deny icmp any any echo log
access-list 101 deny icmp any any redirect log
access-list 101 deny icmp any any mask-request log
access-list 101 deny udp any any range 33400 34400 log
access-list 101 permit ip <internal addresses—wildcard mask> any echo
access-list 101 permit ip <internal addresses—wildcard mask> any parameter-problem
access-list 101 permit ip <internal addresses—wildcard mask> any packet-too-big
access-list 101 permit ip <internal addresses—wildcard mask> any source quench
access-list 101 permit udp <internal address—wildcard mask> range 33400 34400
access-list 101 permit ip <internal addresses—wildcard mask>

interface <to ISP>
ip access-group 101 in
Introduction:

Cisco Secure is a network security server, which runs on UNIX and Windows. TACACS+ (Terminal Access Controller Access System) is a protocol that provides security verification between the NAS (Network Access Server) and the network security server.

*RADIUS* (Remote Access Dial-In User Service) is another protocol that does the same thing.

Installing CS Server:

1. Install the software. TACACS+ will appear as a new service under Windows services.
2. The installation will generate basic AAA configuration that can be copied and pasted into the router.

Cisco Secure uses a Web-based interface for management. Browse to the loopback address (127.0.0.1) or the address on the server to access the Web GUI.

3. After the installation is completed, create a remote administrator account.
4. Add user accounts and assign the user to a group. A user can have individual properties, or inherit those of the group.
5. If desired, restrict or allow specific commands for all users assigned to a group.

Keep track of updates and changes at [http://www.spcollege.edu/star/cisco](http://www.spcollege.edu/star/cisco) Scroll to the bottom of the page and click on the “Lab Manual Edits.”

Take college courses on a new schedule--yours.

Choose from over 400 online courses in dozens of majors, including our new bachelor's degree program in Technology Management. St. Petersburg College's eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education. Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

[http://e.spcollege.edu](http://e.spcollege.edu)

St. Petersburg College's eCampus. Your time. Your place. Your future.
Implementing Cisco Secure Server
By Micheal Gordon

Directions: In the following lab, you will configure your Network Access Server to communicate with a Cisco Secure server, to authenticate users and run authorization for all IOS commands entered.

Creating a Back Door:
Before enabling AAA on the router, it is good practice to configure a means of entry into the router that will bypass the TACACS+ authentication process. This will act as a "safety net" in case of a configuration or authentication problem.

To enable authentication, authorization, and accounting, enter the following command before any other "aaa" commands:

```
router(config)#aaa new-model
```

Add the "no_tacacs" command line to bypass any authentication required on the console port of the NAS. This way, the NAS configuration can be reviewed from the console port and reconfigured if necessary.

```
router(config)#aaa authentication login no_tacacs enable
router(config)#line con 0
router(config-line)#login authentication no_tacacs
```

To gain access through the console port of the NAS, use the usual enable secret password.

Configuring the NAS for AAA:
In this procedure, you will configure the NAS Authentication, Authorization, and Accounting parameters. Enter the following command to require authentication for terminal sessions (telnet).

```
router(config)#aaa authentication login default tacacs+
```

Enter the following command for authorization. These commands, in conjunction with the CiscoSecure ACS Group Setup, determine if a terminal session is allowed.
Enter the following command to capture accounting information such as the starting and ending time of a remote client.

```
router(config)#aaa accounting exec acct-list start-stop tacacs+
```

**Establishing Communication between the NAS and the CS server:**
In this procedure, you will perform the necessary configuration to ensure the CS server and the NAS have communication. Identify the IP address of the Windows 2000 server running the ACS program.

```
router(config)#tacacs-server host 172.16.10.200 single-connection
```

Using "single" in the command line allows for a single TCP connection between the NAS and the CS server to improve performance.

Define the TACACS+ key. An identical TACACS+ key has been entered in the CS server.

```
router(config)# tacacs-server key cisco
```

The TACACS+ key is case sensitive.

**Completing Command Authorization:**
In the procedure, you will complete command authorization configuration. As soon as this command is entered, you will have to exit and re-enter the router to perform any privileged level commands.

Enter the following command to run authorization for all commands at the specified privilege level. This will require every line entered by the user to be authorized by TACACS+.

```
router(config)#aaa authorization commands 15 default tacacs+
```

Your NAS is now configured for TACACS+ operation.
Testing Cisco Secure Server for Functionality:
Configure the CS server with a user account named "Henry", pwd "12345", and assign it to the Default Group. The group has been configured to have certain privilege-level 15 commands restricted. In this procedure, you will test the user authentication and command authorization processes. After which, RouterA must disconnect, allowing RouterB to connect and test, followed by RouterC.

1. End your current Telnet session.
2. Telnet back into your router.
3. When prompted for a username, enter Henry.
4. When prompted for a password, enter 54321.

   User authentication should fail.

5. When prompted for a username again, enter Henry.
6. When prompted for a password, enter 12345.

   User authentication should be successful.

7. Enter config mode, and attempt to change the secret password.

   You should receive a message of Command authorization failed.

8. Attempt the change the hostname of the router.

   You should be successful.

9. Attempt to save your changes.

   You should receive a message of Command authorization failed.

If your NAS responds according to previous steps, TACACS+ and the CS server are functioning properly.
Reverse Engineering with Subnets and Subnet Design

Outline:
- Show IP route command
- VLSM
- RIP and RIPV2 with Redistribution
- Redistribution of RIP and IGRP
- Static and Discontiguous Routes
- Gateway of Last Resort
- Broken Labs
- Troubleshooting help

Objective:
Oh my gosh…there are just so many topics that come up in this lab. This essentially reviews subnet design from the CCNA classes and adds discussions on routing protocol choice, route summarization (truncation at the classful boundary), VLSM, static and dynamic routing, and the show ip route command. This is thick.

Tools and Materials:
I used three to five 2610 routers (single Ethernet with a dual serial WIC) for this lab (with IOS version 12.0(13)). If you use other models, then your outputs may vary slightly. Use the troubleshooting chart at the end of this lab if you encounter problems.
(3-5) DCE/DTE cables
(1-3) PC workstations
(1-3) console cables
(5-6) straight-through cables

Background:
Many times while troubleshooting networks you may find yourself in an “unusual” position of not having a topology diagram. I know if it was your network you would have one because you all have had documenting, documenting, and more documenting drummed into your head. So how can you make one quickly? That’s right…using the “show ip route” command. There is just so much helpful information packed in there. In this lab you should start seeing everything “coming together.” Just as a refresher let’s go over some of the finer points of the command. Here we see a simple three router network diagram. We can see all kinds of goodies in our show ip route command:
routerB#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets
C 10.1.1.0 is directly connected, Serial0/1
11.0.0.0/24 is subnetted, 1 subnets
C 11.1.1.0 is directly connected, Serial0/0

First of all I want you to notice the router prompt. I used router B in the previous example. Throughout this lab be certain to notice the router prompt to give you an idea of which router I am referring to. Let’s start at the top...we can see all kinds of codes for the types of routing protocols. Some of the ones you will want to pay particularly close attention to include the connected (C), static (S), IGRP (I), RIP (RIP), BGP (B), EIGRP (D—note...D not E), and OSPF (O). Later on you can worry about the others. Then we see a statement “Gateway of last resort is not set.” Usually this is set on the router that leads to the “outside world” with a “ip route” statement. Basically it is used for information that is unaccounted for in a network design...this statement means “if you cannot find the destination in your routing table then use the gateway of last resort ip address for your destination....then the gateway will decide what to do with it. Again, more stuff for you to worry about later. Then we get to the “meat and potatoes” of the command: the actual routes being广播ed and learned from other router broadcasts. Since there are no Ethernet, Loopback, or other serial routes being broadcast we only see the directly connected routes on our middle router. If we were to add a fourth router we would see the routes being learned via a routing protocol:
This is what you would expect to see on a four-router example with the routing protocol working properly:

```
routerB#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
      U - per-user static route, o - ODR
Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets
 C   10.1.1.0 is directly connected, Serial0/1
11.0.0.0/24 is subnetted, 1 subnets
 C   11.1.1.0 is directly connected, Serial0/0
 I   12.0.0.0/8 [100/10476] via 10.1.1.2, 00:01:21, Serial0/0
```

Now, going back to our three-router example, what do you suppose we would see on routerA and routerC (IGRP is being used for routing)? Take a second to think about it and then turn to the next page.
routerA#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

    10.0.0.0/24 is subnetted, 1 subnets
C   10.1.1.0 is directly connected, Serial0/0
I   11.0.0.0/8 [100/10476] via 10.1.1.2, 00:01:21, Serial0/0

routerC#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

I  10.0.0.0/8 [100/10476] via 11.1.1.2, 00:00:23, Serial0/1
    11.0.0.0/24 is subnetted, 1 subnets
C   11.1.1.0 is directly connected, Serial0/1

Here we can see the routes being learned via the IGRP routing protocol. If the routing
protocol was not working properly, then we would only see just the directly connected
routes. Why didn’t I tell you the IP addresses that are being used? Remember we are
trying to make a drawing because we don’t already have one. From the information here
can you draw the logical topology? Don’t forget to include the masks. Turn the page for
the answers when you are done

![Logical topology diagram]

IP addresses:
    routerA    routerB    routerC
S0/0
S0/1

550
As Emeril likes to say…”Let’s kick it up a notch!” Let’s add in some loopback interfaces on each router and broadcast the routes. On routerA, if we add loopback interfaces 172.16.1.1/24 (loop 1), 172.17.1.1/24 (loop 2), and 172.18.1.1/24 (loop 3), then what would you expect to find in the output from the “show ip route” command on routerA, routerB, and routerC? Use the worksheet on the next page to fill in your answers.
routerA#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

routerB#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

routerC#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set
Let’s check your answers:

routerA#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
     D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
     i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
     U - per-user static route, o - ODR

Gateway of last resort is not set

    172.17.0.0/24 is subnetted, 1 subnets
     C      172.17.1.0 is directly connected, Loopback2
     172.16.0.0/24 is subnetted, 1 subnets
     C      172.16.1.0 is directly connected, Loopback1
     172.18.0.0/24 is subnetted, 1 subnets
     C      172.18.1.0 is directly connected, Loopback3
     10.0.0.0/24 is subnetted, 1 subnets
     C      10.1.1.0 is directly connected, Serial0/0
     I      11.0.0.0/8 [100/10476] via 10.1.1.2, 00:00:12, Serial0/0

routerB#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
     D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
     i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
     U - per-user static route, o - ODR

Gateway of last resort is not set

    I      172.17.0.0/16 [100/8976] via 10.1.1.1, 00:00:01, Serial0/1
    I      172.16.0.0/16 [100/8976] via 10.1.1.1, 00:00:01, Serial0/1
    I      172.18.0.0/16 [100/8976] via 10.1.1.1, 00:00:01, Serial0/1
     10.0.0.0/24 is subnetted, 1 subnets
     C      10.1.1.0 is directly connected, Serial0/1
     11.0.0.0/24 is subnetted, 1 subnets
     C      11.1.1.0 is directly connected, Serial0/0

routerC#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
     D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
     i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
     U - per-user static route, o - ODR

Gateway of last resort is not set

    I      172.17.0.0/16 [100/10976] via 11.1.1.2, 00:00:11, Serial0/1
    I      172.16.0.0/16 [100/10976] via 11.1.1.2, 00:00:12, Serial0/1
    I      172.18.0.0/16 [100/10976] via 11.1.1.2, 00:00:12, Serial0/1
     10.0.0.0/8 [100/10476] via 11.1.1.2, 00:00:12, Serial0/1
     11.0.0.0/24 is subnetted, 1 subnets
     C      11.1.1.0 is directly connected, Serial0/0

routerC#
Ok…now let’s go ahead and go crazy with our loopbacks. On routerB add loopbacks 172.19.1.1/24 (loop4), 172.20.1.1/24(loop5), and 172.21.1.1/24(loop6). And, you guessed it, on routerC add loopbacks 172.22.1.1/24(loop7), 172.23.1.1/24(loop8), and 172.24.1.1/24(loop9). What would you see?

routerA#show ip route

Codes:  
C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
* - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

routerB#show ip route

Codes:  
C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
* - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

routerC#show ip route

Codes:  
C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
* - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set
Let's check your answers:

**routerA#sh ip route**

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is not set

172.17.0.0/24 is subnetted, 1 subnets
C   172.17.0.0/24 is directly connected, Loopback2
C   172.16.0.0/24 is subnetted, 1 subnets
C   172.16.1.0 is directly connected, Loopback1
I   172.19.0.0/16 [100/8976] via 10.1.1.2, 00:00:27, Serial0/0
C   172.18.0.0/24 is subnetted, 1 subnets
C   172.18.1.0 is directly connected, Loopback3
I   172.21.0.0/16 [100/8976] via 10.1.1.2, 00:00:27, Serial0/0
I   172.20.0.0/16 [100/8976] via 10.1.1.2, 00:00:27, Serial0/0
I   172.23.0.0/16 [100/10976] via 10.1.1.2, 00:00:27, Serial0/0
I   172.22.0.0/16 [100/10976] via 10.1.1.2, 00:00:27, Serial0/0
I   172.24.0.0/16 [100/10976] via 10.1.1.2, 00:00:27, Serial0/0
C   172.19.0.0/16 is subnetted, 1 subnets
C   172.18.0.0/24 is subnetted, 1 subnets

**routerB#sh ip route**

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is not set

I   172.17.0.0/16 [100/8976] via 10.1.1.1, 00:00:01, Serial0/1
I   172.16.0.0/16 [100/8976] via 10.1.1.1, 00:00:02, Serial0/1
C   172.19.0.0/24 is subnetted, 1 subnets
C   172.19.1.0 is directly connected, Loopback4
I   172.18.0.0/16 [100/8976] via 10.1.1.1, 00:00:02, Serial0/1
C   172.21.0.0/24 is subnetted, 1 subnets
C   172.18.1.0 is directly connected, Loopback6
C   172.20.0.0/24 is subnetted, 1 subnets
C   172.20.1.0 is directly connected, Loopback5
I   172.23.0.0/16 [100/8976] via 11.1.1.1, 00:00:35, Serial0/0
I   172.22.0.0/16 [100/8976] via 11.1.1.1, 00:00:35, Serial0/0
I   172.24.0.0/16 [100/10976] via 11.1.1.1, 00:00:35, Serial0/0
C   172.19.0.0/16 is subnetted, 1 subnets
C   10.1.1.0 is directly connected, Serial0/1
I   11.0.0.0/8 [100/10476] via 10.1.1.2, 00:00:28, Serial0/0
C   11.1.1.0 is directly connected, Serial0/0
got the finer points? see which address is actually used for those routes that are learned? the close one or the far one? the interface it came in on or the one that it came from? good things to recognize. let’s revisit the “reverse engineering” part by drawing this little network from our routerC “show ip route” information. first, let’s start with our directly connected networks…those are easiest.

We can also add our serial connection S0/1 since our information shows it to be directly connected too. Without looking at the running configuration or interface configuration can we deduce the actual ip address of S0/1 from the show ip route? Yup…look in those routes learned via a routing protocol. We know (1) that the ip address is part of the 11.1.1.0/24 network and we know (2) that routing is learned via 11.1.1.2, Serial0/1 so we
can deduce the IP address of Serial0/1 is 11.1.1.2. We can reasonably guess the IP address of Serial0/0 over on RouterB is 11.1.1.1 since they both need to be in the same subnet. To verify we can look at the show ip route information on RouterB. So now all we are left with is the routes learned via a routing protocol.

I 172.17.0.0/16 [100/10976] via 11.1.1.2, 00:00:39, Serial0/1
I 172.16.0.0/16 [100/10976] via 11.1.1.2, 00:00:39, Serial0/1
I 172.19.0.0/16 [100/8976] via 11.1.1.2, 00:00:40, Serial0/1
I 172.18.0.0/16 [100/10976] via 11.1.1.2, 00:00:40, Serial0/1
I 172.21.0.0/16 [100/8976] via 11.1.1.2, 00:00:40, Serial0/1
I 172.20.0.0/16 [100/8976] via 11.1.1.2, 00:00:40, Serial0/1

…
I 10.0.0.0/8 [100/10476] via 11.1.1.2, 00:00:40, Serial0/1

We can verify which end is the DCE and which is DTE using the “sh controller” command:

```
routerB#show controller s0/0
Interface Serial0/0
Hardware is PowerQUICC MPC860
DCE V.35, clockrate 56000

…
```

Let’s look at it on the dte end:
```
routerC#show controller s0/1
Interface Serial0/1
DTE V.35 TX and RX clock detected

…
```

Ok…let’s remove the cable and look at it again:
```
routerC#show controller s0/1
Interface Serial0/1
No serial cable attached

```

Good way to see if there is a cable plugged in…you can’t always fly to another location just to check to see if the cable is plugged in…this is a quick way to find out. Here is another output (as long as we are on the subject):

No clockrate set on DCE:
```
routerB#show controller s0/0
Interface Serial0/0
Hardware is PowerQUICC MPC860
DCE V.35, no clock

```

```
routerC#show controller s0/1
Interface Serial0/1
DTE V.35 clocks stopped

```

557
We know that these are the networks that exist outside of RouterC but we really do not have a good idea of how they are put together with the exception of all of this information comes into RouterC through Serial0/1. To get a better idea we need to go to RouterB and get information from it’s show ip route. From there we can deduce that the other serial line is the 10.0.0.0/8 network and the rest are loopback networks on RouterA and RouterB. Now we can finish off our logical diagram (don’t forget to add DCE/DTE) You can double-check this by setting up the lab (Learn by doing).

![Diagram of three routers connected through serial interfaces]

**IP addresses:**
- routerA
- routerB
- routerC

- S0/0
- S0/1
- Loop1
- Loop2
- Loop3
- Loop4
- Loop5
- Loop6
- Loop7
- Loop8
- Loop9

Let’s take a second and look at the way our routes that are learned via a routing statement are displayed…

```
I    172.17.0.0/16 [100/10976] via 11.1.1.2, 00:00:39, Serial0/1
```

First we see this route is learned via the IGRP (I) routing protocol. That’s a no-brainer because we see all of the codes in the table above. Next we see the network that was broadcast [some metrics…I’ll come back to this in a minute] and received through which interface [Serial0/1], and how long it has been since the last update [00:00:39]. Your knowledge of routing protocols will be important for looking at that timer…for example if that update timer said 00:1:49 with IGRP would we need to be concerned? Heck yes…remember IGRP updates every 90 seconds (01:30)…Obviously there would be a bit of a problem to troubleshoot there.

Ok, so now we come back to those metrics. Essentially if we have two routes to the same destination then the router with choose the route with the lowest metrics in those brackets []. It will use the first number first and then, in case of a tie, it will go to the next group and so on (we’ll talk about this more later). In the case of RIP we only use hop count. With IGRP we use Bandwidth and Delay. Oh sure, there are more metrics like load, MTU, and reliability but these are the DEFAULT metrics for the IGRP protocol. The first number in the bracket is the administrative distance and the next number is the overall metric based upon the routing protocol in use. Let’s look at the statement again. IGRP has the administrative distance of 100 and we see a metric of 10976. Since we used default metrics with IGRP (Bandwidth and delay) which, when
calculated (go out to Cisco’s website to see how they are calculating them…I double-dog
dare you), come up to a total of 10976. You can use the “show interface” command to see each metric.

`routerB#sh int s0/0`
Serial0/0 is up, line protocol is up
   Hardware is PowerQUICC Serial
   Description: serial line to routerC
   Internet address is 11.1.1.1/24
   MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec, rely 255/255, load 1/255
   Encapsulation HDLC, loopback not set, keepalive set (10 sec)
   Last input 00:00:08, output 00:00:09, output hang never
   Last clearing of "show interface" counters never

Ok, so let’s try creating another show ip route table for another network. Give me the outputs for each router based upon this diagram using RIP and the charts on the next page.

```plaintext
routerA               routerB
                        10.0.0.0/8
Loop 1) 200.0.0.1/24   Loop 4) 203.0.0.1/24
Loop 2) 201.0.0.1/24   Loop 5) 204.0.0.1/24
Loop 3) 202.0.0.1/24   Loop 6) 205.0.0.1/24
```
routerA#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-1, IS-IS level-2,
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

routerB#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-1, IS-IS level-2,
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set
And then we will check the answers

**routerA#sh ip route**

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
      U - per-user static route, o - ODR

Gateway of last resort is not set

I  204.0.0.0/24 [100/8976] via 10.1.1.2, 00:00:07, Serial0/0
I  205.0.0.0/24 [100/8976] via 10.1.1.2, 00:00:08, Serial0/0
C  200.0.0.0/24 is directly connected, Loopback1
C  201.0.0.0/24 is directly connected, Loopback2
C  202.0.0.0/24 is directly connected, Loopback3
I  203.0.0.0/24 [100/8976] via 10.1.1.2, 00:00:08, Serial0/0
C  10.0.0.0/8 is directly connected, Serial0/0

**routerB#sh ip route**

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
      U - per-user static route, o - ODR

Gateway of last resort is not set

C  204.0.0.0/24 is directly connected, Loopback5
C  205.0.0.0/24 is directly connected, Loopback6
I  200.0.0.0/24 [100/80625] via 10.1.1.1, 00:00:33, Serial0/1
I  201.0.0.0/24 [100/80625] via 10.1.1.1, 00:00:33, Serial0/1
I  202.0.0.0/24 [100/80625] via 10.1.1.1, 00:00:33, Serial0/1
C  203.0.0.0/24 is directly connected, Loopback4
C  10.0.0.0/8 is directly connected, Serial0/1

routerB#

Your output may vary a bit if you picked different addresses for your serial lines so don’t freak if it is a bit different. The big thing to look for is an identical number of directly connected and routing statements (count the little “I’s” and “C’s”).
You should have this down pretty good by now but let’s add a third router and some more loopback networks in for good measure. Don’t forget to try doing this in the lab if you get confused.

routerA#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

routerB#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set
routerC# show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

Turn to the next page to check your answer.
Let’s check your answers:

**routerA#sh ip route**

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is not set

```
I   204.0.0.0/24 [100/8976] via 10.1.1.2, 00:00:11, Serial0/0
I   205.0.0.0/24 [100/8976] via 10.1.1.2, 00:00:11, Serial0/0
I   206.0.0.0/24 [100/82625] via 10.1.1.2, 00:00:11, Serial0/0
I   207.0.0.0/24 [100/82625] via 10.1.1.2, 00:00:11, Serial0/0
C   200.0.0.0/24 is directly connected, Loopback1
C   201.0.0.0/24 is directly connected, Loopback2
C   202.0.0.0/24 is directly connected, Loopback3
I   203.0.0.0/24 [100/8976] via 10.1.1.2, 00:00:11, Serial0/0
C   10.0.0.0/8 is directly connected, Serial0/0
I   11.0.0.0/8 [100/82125] via 10.1.1.1, 00:00:14, Serial0/1
I   200.0.0.0/24 is directly connected, Loopback4
C   203.0.0.0/24 is directly connected, Loopback5
C   204.0.0.0/24 is directly connected, Loopback6
I   205.0.0.0/24 [100/80625] via 11.1.1.2, 00:00:19, Serial0/0
I   206.0.0.0/24 [100/80625] via 11.1.1.2, 00:00:19, Serial0/0
I   207.0.0.0/24 [100/80625] via 11.1.1.2, 00:00:19, Serial0/0
I   200.0.0.0/24 [100/80625] via 10.1.1.1, 00:00:14, Serial0/1
I   201.0.0.0/24 [100/80625] via 10.1.1.1, 00:00:14, Serial0/1
I   202.0.0.0/24 [100/80625] via 10.1.1.1, 00:00:14, Serial0/1
C   203.0.0.0/24 is directly connected, Loopback4
C   10.0.0.0/8 is directly connected, Serial0/1
   11.0.0.0/24 is subnetted, 1 subnets
C   11.1.1.0 is directly connected, Serial0/0
I   208.0.0.0/24 [100/80625] via 11.1.1.2, 00:00:19, Serial0/0
routerB#
```
Now, let’s go backwards like I promised…given the `sh ip route` outputs I want you to draw me the diagram of the network.

**routerA#**

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
    D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
    U - per-user static route, o - ODR

Gateway of last resort is not set

C  10.0.0.0/8 is directly connected, Loopback1
C  20.0.0.0/8 is directly connected, Loopback2
C  30.0.0.0/8 is directly connected, Loopback3
I  40.0.0.0/8 [100/8976] via 192.168.1.2, 00:00:54, Serial0/0
I  50.0.0.0/8 [100/8976] via 192.168.1.2, 00:00:54, Serial0/0
I  60.0.0.0/8 [100/8976] via 192.168.1.2, 00:00:54, Serial0/0
I  70.0.0.0/8 [100/82625] via 192.168.1.2, 00:00:54, Serial0/0
I  80.0.0.0/8 [100/82625] via 192.168.1.2, 00:00:54, Serial0/0
I  90.0.0.0/8 [100/82625] via 192.168.1.2, 00:00:54, Serial0/0
C  192.168.1.0/24 is directly connected, Serial0/0
I  192.169.1.0/24 [100/82125] via 192.168.1.2, 00:00:55, Serial0/0
routerB#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

I   1.0.0.0/8 [100/80625] via 192.168.1.1, 00:01:05, Serial0/1
I   2.0.0.0/8 [100/80625] via 192.168.1.1, 00:01:05, Serial0/1
I   3.0.0.0/8 [100/80625] via 192.168.1.1, 00:01:05, Serial0/1
C   4.0.0.0/8 is directly connected, Loopback4
C   5.0.0.0/8 is directly connected, Loopback5
C   6.0.0.0/8 is directly connected, Loopback6
I   7.0.0.0/8 [100/80625] via 192.169.1.2, 00:01:18, Serial0/0
I   8.0.0.0/8 [100/80625] via 192.169.1.2, 00:01:18, Serial0/0
I   9.0.0.0/8 [100/80625] via 192.169.1.2, 00:01:18, Serial0/0
C   192.168.1.0/24 is directly connected, Serial0/1
C   192.169.1.0/24 is directly connected, Serial0/0

routerC#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

I   1.0.0.0/8 [100/82625] via 192.168.1.1, 00:01:11, Serial0/1
I   2.0.0.0/8 [100/82625] via 192.168.1.1, 00:01:11, Serial0/1
I   3.0.0.0/8 [100/82625] via 192.168.1.1, 00:01:11, Serial0/1
I   4.0.0.0/8 [100/8976] via 192.168.1.1, 00:01:11, Serial0/1
I   5.0.0.0/8 [100/8976] via 192.168.1.1, 00:01:11, Serial0/1
I   6.0.0.0/8 [100/8976] via 192.168.1.1, 00:01:11, Serial0/1
C   7.0.0.0/8 is directly connected, Loopback7
C   8.0.0.0/8 is directly connected, Loopback8
C   9.0.0.0/8 is directly connected, Loopback9
I   192.168.1.0/24 [100/82125] via 192.169.1.1, 00:01:11, Serial0/1
C   192.169.1.0/24 is directly connected, Serial0/1

routerC#
Let’s check your answer

IP addresses:

<table>
<thead>
<tr>
<th></th>
<th>routerA</th>
<th>routerB</th>
<th>routerC</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0/0</td>
<td>192.168.1.1/24</td>
<td>192.169.1.1/24</td>
<td>-----</td>
</tr>
<tr>
<td>S0/1</td>
<td>-----</td>
<td>192.168.1.2/24</td>
<td>192.169.1.2/24</td>
</tr>
</tbody>
</table>

Loop1  1.1.1.1/8  Loop4  4.4.4.4/8  Loop7  7.7.7.7/8
Loop2  2.2.2.2/8  Loop5  5.5.5.5/8  Loop8  8.8.8.8/8
Loop3  3.3.3.3/8  Loop6  6.6.6.6/8  Loop9  9.9.9.9/8

Routing protocol:
Ok...let’s try another one:

RouterA#sh ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
     D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
     i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
     U - per-user static route, o - ODR

Gateway of last resort is not set

C  1.0.0.0/8 is directly connected, Loopback1
C  2.0.0.0/8 is directly connected, Loopback2
C  3.0.0.0/8 is directly connected, Loopback3
R  4.0.0.0/8 [120/1] via 192.168.1.2, 00:00:29, Serial0/0
R  5.0.0.0/8 [120/1] via 192.168.1.2, 00:00:29, Serial0/0
R  6.0.0.0/8 [120/1] via 192.168.1.2, 00:00:29, Serial0/0
R  7.0.0.0/8 [120/1] via 192.170.1.1, 00:00:05, Serial0/1
R  8.0.0.0/8 [120/1] via 192.170.1.1, 00:00:05, Serial0/1
R  9.0.0.0/8 [120/1] via 192.170.1.1, 00:00:05, Serial0/1
C 192.170.1.0/24 is directly connected, Serial0/1
C 192.168.1.0/24 is directly connected, Serial0/0
R 192.169.1.0/24 [120/1] via 192.168.1.2, 00:00:00, Serial0/0
     [120/1] via 192.170.1.1, 00:00:06, Serial0/1
RouterA#

RouterB#sh ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
     D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
     i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
     U - per-user static route, o - ODR

Gateway of last resort is not set

R  1.0.0.0/8 [120/1] via 192.168.1.1, 00:00:08, Serial0/1
R  2.0.0.0/8 [120/1] via 192.168.1.1, 00:00:08, Serial0/1
R  3.0.0.0/8 [120/1] via 192.168.1.1, 00:00:08, Serial0/1
C  4.0.0.0/8 is directly connected, Loopback4
C  5.0.0.0/8 is directly connected, Loopback5
C  6.0.0.0/8 is directly connected, Loopback6
R  7.0.0.0/8 [120/1] via 192.169.1.2, 00:00:03, Serial0/0
R  8.0.0.0/8 [120/1] via 192.169.1.2, 00:00:03, Serial0/0
R  9.0.0.0/8 [120/1] via 192.169.1.2, 00:00:03, Serial0/0
R 192.170.1.0/24 is possibly down, routing via 192.168.1.1, Serial0/1
C 192.168.1.0/24 is directly connected, Serial0/1
C  192.169.1.0/24 is directly connected, Serial0/0
routerB#

routerC#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is not set

R  1.0.0.0/8  [120/1]  via 192.170.1.2, 00:00:17, Serial0/0
R  2.0.0.0/8  [120/1]  via 192.170.1.2, 00:00:17, Serial0/0
R  3.0.0.0/8  [120/1]  via 192.170.1.2, 00:00:17, Serial0/0
R  4.0.0.0/8  [120/1]  via 192.169.1.1, 00:00:18, Serial0/1
R  5.0.0.0/8  [120/1]  via 192.169.1.1, 00:00:18, Serial0/1
R  6.0.0.0/8  [120/1]  via 192.169.1.1, 00:00:18, Serial0/1
C  7.0.0.0/8 is directly connected, Loopback7
C  8.0.0.0/8 is directly connected, Loopback8
C  9.0.0.0/8 is directly connected, Loopback9
C  192.170.1.0/24 is directly connected, Serial0/0
R  192.168.1.0/24  [120/1]  via 192.169.1.1, 00:00:19, Serial0/1
  [120/1]  via 192.170.1.2, 00:00:18, Serial0/0
C  192.169.1.0/24 is directly connected, Serial0/1

routerC#

Turn to the next page to check your answer…no peaking!
Let’s check your answer

![Network diagram](image)

IP addresses:

<table>
<thead>
<tr>
<th></th>
<th>routerA</th>
<th>routerB</th>
<th>routerC</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0/0</td>
<td>192.168.1.1/24</td>
<td>192.169.1.1/24</td>
<td>-----</td>
</tr>
<tr>
<td>S0/1</td>
<td>-----</td>
<td>192.168.1.2/24</td>
<td>192.169.1.2/24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Loop</th>
<th></th>
<th></th>
<th>Loop</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.1.1.1/8</td>
<td></td>
<td>4.4.4.4/8</td>
<td></td>
<td>7.7.7.7/8</td>
</tr>
<tr>
<td>Loop2</td>
<td>2.2.2.2/8</td>
<td>Loop5</td>
<td>5.5.5.5/8</td>
<td>Loop8</td>
<td>8.8.8.8/8</td>
</tr>
<tr>
<td>Loop3</td>
<td>3.3.3.3/8</td>
<td>Loop6</td>
<td>6.6.6.6/8</td>
<td>Loop9</td>
<td>9.9.9.9/8</td>
</tr>
</tbody>
</table>

Routing protocol:

Don’t forget to add dce/dte too
Have you noticed anything strange or any patterns with my addressing schemes so far? I have been keeping them “in-bounds” with our addressing patterns. Remember way, way back when you learned about address classes? Let’s review really quick:

<table>
<thead>
<tr>
<th>Class</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>Network.Host.Host.Host</td>
</tr>
<tr>
<td>Class B</td>
<td>Network.Network.Host.Host</td>
</tr>
</tbody>
</table>

We learned that each address class has a network portion and a host portion. The network portion is assigned to us and cannot be “tinkered” with by us. The host portion, on the other hand, is at our mercy. We can do what ever we want with it. That little “demarcation” point between the network and host portion in each class is commonly referred to as the “classful boundary.” Think about it…classful (where the network portion ends) and boundary (a point where something end)...it’s a really geeky sounding thing but it really makes sense. So where does the classful boundary exist in a class A address?

```
Class A   Network.Host.Host.Host
          ↓
Classful boundary
```

Why is this important to us here? Because our choice of routing protocol will dictate whether our addresses will be “truncated at the classful boundary” (chopped off). If the routing information is truncated at the classful boundary then you may encounter routing “issues” with your networks. For example, if we look at our previous example where we used serial lines with addresses of 10.0.0.0/8 and 11.0.0.0/8. Our “network numbers” were retained because they existed before the classful boundary. (Remember we used RIP and IGRP and they do not pass subnet mask information but truncate at the classful boundary). So let’s look at a three router example using IGRP (which truncates at the classful boundary) and some addressing that can cause “issues” for your network.

---

Ok, we can see these addresses are legitimate. They will work when used in certain circumstances. Go ahead and figure out the show ip route tables based upon this information.
routerA#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
* - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

routerB#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
* - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

routerC#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
* - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set
Here is what you actually would see:

**routerA**

```
# show ip route
```

Codes: C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
* - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

---

**routerB**

```
# show ip route
```

Codes: C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
* - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

---

**routerC**

```
# show ip route
```

Codes: C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
* - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

---

573
What?  Darn all that theory stuff.  It just should not have worked.  Isn’t this enough to
drive you completely nuts?  Ok…so technically it should not have worked, but we have
stumbled upon the “exception” to the rule.  For this reason CISCO has included the “ip
classless” command by default with IOS 12 or greater to help make this work.  It really is
good with variable length subnet masking (that’s in a bit) so it is kind of nice.  How to
“break” that exception would be to use several routers with different “major” class
addresses (the network portions).  Let’s look at one (use IGRP or RIP) and how it really
should work:

```
10.1.1.0/16    9.0.0.0/16      10.2.1.0/16
net           net  net
S0/1  S0/0          S0/1       S0/0
S0/0              S0/1       S0/0
routerA  routerB  routerC routerD
```

What would you expect to see?

```
routerA#show ip route
Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
* - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set
```

```
routerB#show ip route
Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
* - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set
```
routerC#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

routerD#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

Turn to the next page to check your answers when you are finished.
Here is what you actually would see:

```
routerA#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

```

```
routerB#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

```

```
routerC#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```
Why did this happen? Because IGRP does not pass subnet mask information beyond the classful boundary. Since Class A IP addresses naturally default to an 8-bit subnet mask then an 8-bit mask is the only portion that is passed with the routing information even though we wanted to use a 16-bit mask. When this does happen the routes are said to be “summarized” or “summarized at the classful boundary.” Geek stuff through and through. Try to see what happens when you ping through the network. Interesting stuff there.

Does this mean we cannot use addresses that are summarized? Heck no, we would just need to modify our choice of routing protocol. Both EIGRP or OSPF pass the entire subnet mask information. In fact, let’s see what happens to our table when we switch to EIGRP.

routerC#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

577
See? Now the routes are being advertised, not truncated at the classful boundary? Your choice of routing protocol is extremely important when designing subnets in your network. Can you think of reasons why you would want routes advertised only at the classful boundary?

Given the following IP addresses that will be used in your network, which routing protocols would you most likely choose to use with which addressing scheme?

1. 184.43.17.9/24
2. 201.201.201.201/26
3. 14.23.43.65/8
4. 114.43.212.84/19
5. 199.4.123.12/25

Now check your answers on the next page.
Variable Length Subnet Masking (VLSM)

When designing networks it is preferable to be as efficient as possible when assigning IP addresses. As we have seen sometimes we even need to use contiguous (sequential) numbers for our subnet schemes even with classless addresses. As your skills in networking and networking design increase you will need to know how to efficiently utilize VLSM (RFC 1219).

Tools and Materials:
Paper and pencils
Super VLSM chart (http://www.henninger.net/downloads/ccna/tools/subnettable.pdf)

Lab Diagram:
Let’s go through one example using the above network design and a class “C” network address given as 212.14.17.x/24.

10. Determine largest network needed: 57 IP’s. This will fit into a network in our first column (62 hosts max). So we put down 212.14.17.64/26 for that network and color out the ip address ranges from .64 to .124 on our chart (all the way across the chart). Our actual usable addresses are .65 to .126…the columns all the way on the left are not that specific.

11. Determine the next largest network needed: 39 IP’s. This will fit into a network in our first column (62 hosts max). So we put down 212.14.17.128/26 for that network and color out the ip address ranges from .128 to .188 on our chart (all the way across the chart). Our actual usable addresses are .129-.190.

12. Determine the next largest network needed: 28 IP’s. This will fit into a network in our second column (30 hosts max). So we put down 212.14.17.32/27 for that network and color out the ip address ranges from .32 to .60 on our chart (all the way across the chart). Our actual usable addresses are .33-.62.

13. Determine the next largest network needed: 24 IP’s. This will fit into a network in our second column (30 hosts max). So we put down 212.14.17.192/27 for that network and color out the ip address ranges from .192 to .220 on our chart (all the way across the chart). Our actual usable addresses are .193-.222.

14. Determine the next largest network needed: 14 IP’s. This will fit into a network in our third column (14 hosts max). So we put down 212.14.17.16/28 for that network and color out the ip address ranges from .16 to .28 on our chart (all the way across the chart). Our actual usable addresses are .17-.30.

15. Determine the next largest network needed: 12 IP’s. This will fit into a network in our third column (14 hosts max). So we put down 212.14.17.224/28 for that network and color out the ip address ranges from .224 to .236 on our chart (all the way across the chart). Our actual usable addresses are .225-.238.

16. Determine the next largest network needed: 6 IP’s. This will fit into a network in our fourth column (6 hosts max). So we put down 212.14.17.8/29 for that network and color out the ip address ranges from .8 to .12 on our chart (all the way across the chart). Our actual usable addresses are .9-.14.

17. Determine the next largest network needed: 2 IP’s. This will fit into a network in our fifth column (2 hosts max). So we put down 212.14.17.4/30 for that network and color out the ip address ranges from .4 to .8 on our chart (all the way across the chart). Our actual usable addresses are .5-.6.

18. Don’t forget about those serial lines between our routers! They need subnets with IP’s too. For those we picked, basically what is left. 212.14.17.240/30 (useable .241-.242), 212.14.17.244/30 (useable .245-.246), and 212.14.17.248/30 (useable .249-.250).
These are the addresses for this lab…can you “see” the variable length subnet mask?

212.14.17.x/24  
212.14.17.64/26  
212.14.17.128/26  
212.14.17.32/27  
212.14.17.192/27  
212.14.17.16/28  
212.14.17.248/30

212.14.17.224/28  
212.14.17.8/29  
212.14.17.4/30  
212.14.17.240/30  
212.14.17.244/30  
212.14.17.128/26  
212.14.17.8/26

More Problems:
For the network diagrammed design an IP addressing scheme using VLSM to be as efficient as possible with IP address distribution.

4. You have been assigned the class “C” private IP address by the upper-level IT staff. Other divisions have other Class “C” IP addresses. For now, you only need to know you have the 192.168.70.0/24 network to design.

5. You have been assigned the class “B” private IP address by the upper-level IT staff. Other divisions have other Class “B” IP addresses. For now, you only need to know you have the 172.168.128.0/18 network to design.

6. You have been assigned the class “A” private IP address by the upper-level IT staff. Other divisions have other Class “A” IP addresses. For now, you only need to know you have the 10.16.0.0/12 network to design.

RIP Version 2 and Redistribution with RIP

Earlier you learned about route summarization and choosing your routing protocols carefully. In that lab you learned what routes are passed with RIP (and the lack of passing masks) and which ones are not. Another way to solve that problem would have been to switch to a routing protocol that allowed subnet masks to be passed. One such protocol that does it is RIP version 2. Sometimes we do not always have the luxury to change routing protocols. For those situations we can just redistribute the routes.

Tools and Materials:
(2) PC/workstations  
(2) Routers  
(2) Switches  
(4) Straight-through cables  
(1) DCE serial cable  
(1) DTE serial cable  
(2) rollover cables
**Lab Design:**

![Diagram of network setup]

**Workstation “A”**

**Workstation “B”**

**Addressing:**

<table>
<thead>
<tr>
<th>Routers</th>
<th>Hostnames</th>
<th>S0</th>
<th>S1</th>
<th>L0</th>
<th>E0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phiber</td>
<td>161.20.4.1/30 (DCE)</td>
<td>n/a</td>
<td>161.20.3.1/30</td>
<td>161.20.2.1/24</td>
<td>161.20.2.1</td>
</tr>
<tr>
<td>Optik</td>
<td>n/a</td>
<td>161.20.1.1/24</td>
<td>161.20.5.1/30</td>
<td>161.20.1.1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workstations</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>161.20.2.2</td>
<td>161.20.1.2</td>
</tr>
<tr>
<td>SM</td>
<td>255.255.255.0</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>GW</td>
<td>161.20.2.1</td>
<td>161.20.1.1</td>
</tr>
</tbody>
</table>

**Step-by-Step Instructions:**

8. Cable and set up the lab as shown.
9. Complete the basic router setup on each router.
10. Configure the interfaces on each router.
11. Configure the workstations. You should NOT be able to ping from workstation A to anywhere. Silly billy…we haven’t configured a routing protocol yet.
12. So let’s fix that little problem here using RIP version 2. Configure the routing protocol and advertise the router’s networks using RIP version 2 by doing this:

```
phiber(config)#router rip
phiber(config-router)#network 161.20.0.0
phiber(config-router)#version 2
```

And on the other router:
optik(config)#router rip
optik(config-router)#network 161.20.0.0
optik(config-router)#version 2

13. Now you should be able to ping from each workstation to the other workstation, to the loopbacks on both routers and everywhere. BAM! Problem solved much easier than with static routes. Yeah…it’s that easy. 😊

14. Now let’s take it up another level and add some more routers to our network (look for the lab diagram at the end of this lab). One router will act as an ISP and the other will be a new company we just acquired. They are running RIP on their network. The boss their likes RIP because she is familiar with it so you decide to leave it intact. But you need to be able to pass your routing information over your network so you use your knowledge of the CISCO website and find out information about two commands you can use to “redistribute” your routing protocol:

   ip rip send version 1
   ip rip receive version 1

8. Also you do not want your ISP to have the information about your network so you decided to stop all routing table broadcasts out the serial interface on phiber. You enter this command:

   phiber(config)#router rip
   phiber(config-router)#passive-interface serial0/1

9. Use your knowledge of debug commands, both before and after implementing the passive interface command, to verify it is working properly. Heck, even a show ip route would work too.

10. Did you remember to statically connect your network to the ISP? Tsk, tsk.
Addressing:

<table>
<thead>
<tr>
<th>Routers</th>
<th>Hostnames</th>
<th>ISP</th>
<th>RIPv1</th>
</tr>
</thead>
<tbody>
<tr>
<td>E0</td>
<td>n/a</td>
<td>192.168.1.1/24</td>
<td></td>
</tr>
<tr>
<td>L0</td>
<td>172.16.1.1/16</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>S0</td>
<td>220.221.222.253/30(DCE)</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>n/a</td>
<td>161.20.6.2/24 (DTE)</td>
<td></td>
</tr>
</tbody>
</table>

So What Have I Learned Here?
In this lab you learned about RIP version 2. It does pass the subnet masks so we can use that VLSM that we learned just a bit ago…See, a place for everything and everything in its place. Isn’t that nice? Now, let’s learn about redistributing IGRP and RIP.

Redistribution of IGRP and RIP

Tools and Materials:
(4) PC/workstations
(4) Routers
(4) Switches
(7) Straight-through cables
(2) rollover cables

Lab Diagram:
Workstation “A”       Workstation “C”       Workstation “B”

Addressing:

Routers

<table>
<thead>
<tr>
<th>Hostnames</th>
<th>Address A</th>
<th>Address B</th>
</tr>
</thead>
<tbody>
<tr>
<td>E0</td>
<td>200.150.100.1/24</td>
<td>202.150.100.1/24</td>
</tr>
<tr>
<td>E1</td>
<td>n/a</td>
<td>203.150.100.1/24</td>
</tr>
<tr>
<td>S0 (DCE)</td>
<td>201.150.100.1/24</td>
<td>n/a</td>
</tr>
<tr>
<td>S1</td>
<td>n/a</td>
<td>201.150.100.2/24</td>
</tr>
</tbody>
</table>

Routers

<table>
<thead>
<tr>
<th>Hostnames</th>
<th>Address A</th>
<th>Address B</th>
</tr>
</thead>
<tbody>
<tr>
<td>E0</td>
<td>202.150.100.2/24</td>
<td>200.150.101.1/24</td>
</tr>
<tr>
<td>E1</td>
<td>203.150.100.2/24</td>
<td>n/a</td>
</tr>
<tr>
<td>S0</td>
<td>n/a</td>
<td>201.150.101.1/24</td>
</tr>
<tr>
<td>S1</td>
<td>201.150.101.2/24</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Workstations

<table>
<thead>
<tr>
<th>Address A</th>
<th>Address B</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>200.150.100.2</td>
</tr>
<tr>
<td>SM</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>GW 1</td>
<td>200.150.100.1</td>
</tr>
<tr>
<td>GW 2</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Workstations

<table>
<thead>
<tr>
<th>Address A</th>
<th>Address B</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>202.150.100.3</td>
</tr>
<tr>
<td>SM</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>GW 1</td>
<td>202.150.100.1</td>
</tr>
<tr>
<td>GW 2</td>
<td>202.150.100.2</td>
</tr>
</tbody>
</table>

Background:

Picture this…your company is running IGRP with an autonomous system number of 38. You have 17 routers in your network spread out over 4 states. Your company buys out another company with IGRP and an autonomous system number of 18 and 15 routers spread out over 2 other states. It would literally take you several days to convert the new network over to work with your network but your boss wants it up and running yesterday. No problem. You can redistribute those other autonomous system numbers into your own on only the “border router” with several simple commands. You can be done in minutes! In this lab you will learn how to redistribute IGRP with IGRP and IGRP with RIP.

Step-By-Step Instructions:
6. Set up and configure the lab as shown.
7. Now let’s redistribute those IGRP areas on the border router:
ritchie(config)#router igrp 38
ritchie(config-router)#redistribute igrp 18

ritchie(config-router)#router igrp 18
ritchie(config-router)#network 201.150.100.0
ritchie(config-router)#redistribute igrp 38

dennis(config)#router igrp 18
dennis(config-router)#network 201.150.100.0
dennis(config-router)#network 200.150.100.0

8. Now we can see how this affects our ip routes. On each router you will see:

dennis#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR
Gateway of last resort is not set

I 202.150.100.0/24 [100/8576] via 201.150.100.2, 00:00:29, Serial0/0
I 203.150.100.0/24 [100/8576] via 201.150.100.2, 00:00:30, Serial0/0
I 201.150.101.0/24 [100/10576] via 201.150.100.2, 00:00:30, Serial0/0
C 200.150.100.0/24 is directly connected, Ethernet0/0
C 201.150.100.0/24 is directly connected, Serial0/0
I 200.150.101.0/24 [100/10676] via 201.150.100.2, 00:00:30, Serial0/0
dennis#

ritchie#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR
Gateway of last resort is not set

C 202.150.100.0/24 is directly connected, Ethernet0/0
C 203.150.100.0/24 is directly connected, Ethernet0/1
I 201.150.101.0/24 [100/8576] via 203.150.100.2, 00:00:20, Ethernet0/1
9. Now let’s change our igrp 18 network over to a RIP network. First let’s get rid of the igrp 18 information:
ritchie(config)#router igrp 38
ritchie(config-router)#no redistribute igrp 18
ritchie(config)#no router igrp 18
dennis(config)#no router igrp 18

10. Now let’s change over to RIP and redistribute it in our network with IGRP:

ritchie(config)#router igrp 38
ritchie(config-router)#redistribute rip 1

ritchie(config-router)#router rip
ritchie(config-router)#network 201.150.100.0
ritchie(config-router)#redistribute igrp 38
dennis(config)#router rip
dennis(config-router)#network 201.150.100.0
dennis(config-router)#network 200.150.100.0

You should be able to ping from router to router without too much problem. However, from workstation A to B will not work because the Time To Live will be exceeded. This is a known problem when redistributing RIP into IGRP where the potential for a routing loop exists. For now just disconnect the straight through cables on Ethernet 0 on both ritchie and ken. This will eliminate the routing loop problem. Relax. Remember RIP takes a while to converge so you might not see the routes or be able to ping for a few minutes. Also, clearing the ip routes a few times couldn’t hurt either:

dennis#clear ip route *
dennis#clear ip route *
dennis#clear ip route *
dennis#clear ip route *

ritchie#clear ip route *
ritchie#clear ip route *
ritchie#clear ip route *
ritchie#clear ip route *

ken#clear ip route *
ken#clear ip route *
ken#clear ip route *
ken#clear ip route *

thompson#clear ip route *
thompson#clear ip route *
thompson#clear ip route *
thompson#clear ip route *
6. Once we have done this then now we can see how this affects our ip routes. On each router you will see:

dennis#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

R  203.150.100.0/24 [120/1] via 201.150.100.2, 00:00:11, Serial0/0
R  201.150.101.0/24 [120/1] via 201.150.100.2, 00:00:12, Serial0/0
C  200.150.100.0/24 is directly connected, Ethernet0/0
C  201.150.100.0/24 is directly connected, Serial0/0
R  200.150.101.0/24 [120/1] via 201.150.100.2, 00:00:12, Serial0/0

ritchie#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

C  203.150.100.0/24 is directly connected, Ethernet0/1
I  201.150.101.0/24 [100/8576] via 203.150.100.2, 00:01:00, Ethernet0/1
R  200.150.100.0/24 [120/1] via 201.150.100.1, 00:00:22, Serial0/1
C  201.150.100.0/24 is directly connected, Serial0/1
I  200.150.101.0/24 [100/8676] via 203.150.100.2, 00:01:00, Ethernet0/1

ken#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

589
U - per-user static route, o - ODR
Gateway of last resort is not set

C  203.150.100.0/24 is directly connected, Ethernet0/1
C  201.150.101.0/24 is directly connected, Serial0/1
I  200.150.100.0/24 [100/1000101] via 203.150.100.1, 00:00:10, Ethernet0/1
I  201.150.100.0/24 [100/8576] via 203.150.100.1, 00:00:10, Ethernet0/1
I  200.150.101.0/24 [100/8576] via 201.150.101.1, 00:01:11, Serial0/1

thompson#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
     D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
     i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
     U - per-user static route, o - ODR

Gateway of last resort is not set

I  203.150.100.0/24 [100/8576] via 201.150.101.2, 00:01:13, Serial0/0
C  201.150.101.0/24 is directly connected, Serial0/0
I  200.150.100.0/24 [100/10002101] via 201.150.101.2, 00:01:13, Serial0/0
I  201.150.100.0/24 [100/10576] via 201.150.101.2, 00:01:13, Serial0/0
C  200.150.101.0/24 is directly connected, Ethernet0/0

thompson#

Notice our our RIP (R) routes are “distributed” as IGRP (I) routes to the right of
the ritchie router.

Supplemental Lab or Challenge Activity:
3. When redistributing IGRP with IGRP what happens if you only redistribute on one
   side (redistribute igrp 38 within 18 but not redistributing igrp 18 within 38)?
4. Repeat this lab with a 26 bit subnet mask. Why does it or doesn’t it work very well
   now?
5. Go out to cisco.com and figure out how to redistribute BGP with IGRP and RIP and
   EIGRP with RIP and IGRP.

So What Did I Learn Here?
In this lab you started to learn the basics about redistribution with routing protocols.
Sorry to tell you this is just the tip of the iceberg. Very few networks use the exact same
routing protocol throughout the entire network (more likely in large networks). In fact
later when you redistribute other protocols you will also have to put metrics in as well.
Whew! RIP…done. IGRP…done. There are three other routing protocols we need to
discuss in later classes or labs: EIGRP, OSPF, and BGP. These three are covered in-
depth in the upper-level CISCO courses but you should be aware of the basics regarding these protocols and for what they are used.

**Static and Dynamic Routes with Discontiguous RIP Networks**

You should have learned that we could use a static route to be able to “route” between what was once “un-routable.” This was known as “auto-summarization” and, by default it is enabled with RIP (and cannot be disabled). We also learned that too many static routes cause problems for administrators. In our earlier lab you learned about route summarization. In that lab you learned what routes are passed with RIP and which ones are not. Just suppose we inherited our network with some given IP addresses and re-assigning IP addresses was not an option. We could use a static route to be able to “route” between what was once “un-routable.” Remember our “exception” to the rule scenario? We could put a static route across it to make it work.

```
10.1.1.0/16    9.0.0.0/16      10.2.1.0/16
net           net  net
S0/1  S0/0          S0/1  S0/0
S0/0  S0/1

routerA  routerB  routerC  routerD
```

```
routerB(config)#ip route 10.2.1.0 255.255.0.0 9.0.0.2
routerC(config)#ip route 10.2.2.0 255.255.0.0 9.0.0.1
```

If you try to ping from end to end it should work now.

*Tools and Materials:*
1. (2) PC/workstations
2. (2) Routers
3. (2) Switches
4. (4) Straight-through cables
5. (1) DCE serial cable
6. (1) DTE serial cable
7. (2) rollover cables
Workstation “A”
Workstation “B”

Addressing:

<table>
<thead>
<tr>
<th>Routers</th>
<th>Hostnames</th>
<th>Phiber</th>
<th>Optik</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>161.20.4.1/30 (DCE)</td>
<td>n/a</td>
<td>161.20.4.2/30 (DTE)</td>
</tr>
<tr>
<td>S1</td>
<td>n/a</td>
<td>161.20.3.1/30</td>
<td>161.20.5.1/30</td>
</tr>
<tr>
<td>L0</td>
<td>161.20.3.1/30</td>
<td>161.20.5.1/30</td>
<td></td>
</tr>
<tr>
<td>E0</td>
<td>161.20.2.1/24</td>
<td>161.20.1.1/24</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workstations</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>161.20.2.2</td>
<td>161.20.1.2</td>
</tr>
<tr>
<td>SM</td>
<td>255.255.255.0</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>GW</td>
<td>161.20.2.1</td>
<td>161.20.1.1</td>
</tr>
</tbody>
</table>

*Step-by-Step Instructions:*
13. Cable and set up the lab as shown.
14. Complete the basic router setup on each router.
15. Configure the interfaces on each router.
16. Configure the routing protocol and advertise/associate/publish the router’s networks. Configure the workstations. You should NOT be able to ping from workstation A to workstation B or vice versa. You should be able to ping from workstation A or B to either loopback. And then try showing the route from …you should see the loopback interface for Phiber (learned via RIP) in the routing table for Optik:

```
optik#sh ip route
```

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
      U - per-user static route, o - ODR
17. So let’s fix that little problem here:

```
optik(config)#ip route 161.20.2.0 255.255.255.0 161.20.4.1
```

What this line says to the router is “to get to the network 161.20.2.0/24 use the interface with the address of 161.20.4.1.” (note: it’s the address on the far side of the serial line…more on that in a bit). Now a request from workstation B to the Ethernet interface has directions on how to get to that address. We have provided them to the router with manual (static) instructions. Our router has summarized our networks because of the addresses we used but, ha-ha!, we are one step ahead of that router because we let it know who’s the boss by slapping a static route in there…take that!

18. Now you should be able to ping from workstation B to the Ethernet interface on Phiber and to workstation A. Now try to ping from workstation A to B. You should not be able to ping. This is because Phiber has no way to direct traffic, even though we did it on Optik. We must add another static route from Phiber to Optik to allow workstation A to be able to ping workstation B. Go ahead and add the route. (Can’t tell you everything step-by-step, otherwise you wouldn’t learn much…ok…if you get stuck you can check the answers.)

19. Static routes are really good for troubleshooting. Later on when you learn about setting up routers with multiple routes to a destination you will learn to use static routes to “force” communication over one path in particular to test that specific path. Suppose the route given by the “-----” in the picture above was chosen by the source router to be the “best path” to the destination router. But we
wanted to test the capabilities of a “lesser path” (given as “                        ”) to the
destination router. We could force the route with a static route.
20. We can actually specify the interface, rather than using the IP address for setting
up a static route (told you we’d come back to it!). So instead of this:

```
optik(config)#ip route 192.168.1.0 255.255.255.0 179.40.6.1
```

For the same thing we could use this:

```
optik(config)#ip route 192.168.1.0 255.255.255.0 serial0/0
```

If you forget your options then use your help command:

```
optik(config)#ip route 192.168.1.0 255.255.255.0 ?
A.B.C.D        Forwarding router's address
FastEthernet  FastEthernet IEEE 802.3
Loopback      Loopback interface
Null          Null interface
Serial        Serial
```

Now let’s explore some of the other options for static routes:

```
optik(config)#ip route 1.0.0.0 255.0.0.0 serial0/1 ?
<1-255>    Distance metric for this route
A.B.C.D     Forwarding router's address
name        Specify name of the next hop
permanent   permanent route
tag         Set tag for this route
```

The first option we see a distance metric for this route. Each routing protocol has
a different default distance metric assigned to it. RIP has a default static route
distance of 120. So actually we already put that in our command, even though it
does not appear in our ip route command. What this is used for is when we want
to put in more than one static route on our router. The router will automatically
select the static route with the lowest distance metric first then, if that route is not
available, go to the route with the second lowest distance metric and then so on.
Distance metrics, as you can see, vary from 1 to 255. Here are some common
metrics for you to know about here at this time:

- Connected interface 0
- Static route 1
- RIP 120
- Unknown 255

If we were to add another router in then we would need to add in another static
route. Using that methodology if we had a network with many routers we could
bury ourselves in static routes which has the possibility of causing major
problems. In our example we just did instead of setting a static route between the
two routers we could set a default network route on optik. This will essentially allow us to add routers at will without all those static routes. Setting many static routes essentially defeats the purposes of having routers make decisions anyways. So there. In the next couple of labs you will learn more about different types of routes and their uses. In the meantime let’s try to do some more exercises and learn by doing!

21. Ok. Let’s try putting a loop back into our network. Connect another serial line from s0/1 (DTE) on phiber to s0/0 (DCE) on optik. Use 56000 for the clockrate. We know from our routing loop labs that our split-horizon is set by default to prevent routing loops, but if we have two paths wouldn’t we want to take advantage of that? Absolutely! If all of our metrics are equal, then our routers will perform load-balancing across the equal lines. Now, of course, you know we can change that. The command to change load-balancing is “variance.” Use your knowledge of the CISCO technical support site and router help features to find out more about this command and how to use it. What we are more concerned with in this lab is static routing. Set your new serial connection to have a different administrative distance than the main line so it will act as a backup line.

22. Ping and trace the route between workstation A and B.

23. Take the main line down (just unplug one end of the serial cable) and ping and trace the route again. Remember RIP may take a few seconds to “catch” up. Your traffic should now be re-routed across the back up line.

24. Bring the main line back up and re-ping and re-trace the route. Unless you used the “permanent” suffix to the ip route command the back up line should still be the preferred line. But…you know how to fix that too.

**Supplemental Labs or Challenge Activities:**

3. Set a whole network with 4-5 routers with routes that are summarized and use static lines to enable “routing.” Now you can see why we don’t always prefer using them.

4. Find out what the other administrative distances are for the other routing protocols. Hint: look on CISCO’s website.

**So What Have I Learned Here?**

In this lab you learned that, while useful, static routes can become a pain in the admin. It is best to do dynamic routing only when absolutely necessary. Later, as you progress in your studies you will better learn when and where to use static routes. But for now just forget about them.

**Gateway of last resort**

You have learned how to use static routes and dynamic routes to accomplish things. But we have another type of static route called the “quad zero” or “gateway of last resort.” Basically this command tells the router “if you have a packet and you don’t know where to send it, then send it to the gateway of last resort address.” Let’s look at one. Make everything IGRP AS43
Loop1--172.16.1.1/16

In this scenario it is good for us to set a gateway of last resort to the ISP router (routerA). Test your network. You should be able to ping from routerD to the Loopback1. So let’s start by looking at our “default” ip routes before we add in this command:

```
routingA#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
routingB#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
        i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
        * - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```
routerC#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
* - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

routerD#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
        D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
        N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
        E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
* - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Ok…now let’s remove any routing protocol between the ISP and routerB. We wouldn’t want to advertise those routes to the Internet, right? We can even put the serial interface into passive mode for good measure. The ping should not work now to Loop 1 from routerD.

Now let’s add a quad zero to routerD:

    RouterD(config):ip route 0.0.0.0 0.0.0.0 220.1.1.2

This says “anything you cannot find in the router table send to 220.1.1.2.” Since the routes for the 220 network are in the table, then the information knows how to get to the “gateway of last resort.” Eh voila! It should work like a champ. But what a second! Let’s see what this does to our show ip routes:

routerA#show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP

597
Gateway of last resort is not set

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

routerB# show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
* - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

routerC# show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
* - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

routerD# show ip route

Codes:  C – connected, S – Static, I – IGRP, R – RIP, M – Mobile, B – BGP
D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
N1 – OSPF NSSA external type 1, N2 – OSPF external type 2,
E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
i – IS-IS, L1 – IS-IS level-2, IS-IS level-2,
*. - candidate default, U – per-user static route, o-ODR

Gateway of last resort is not set

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

All kinds of crazy stuff in this lab. We covered more stuff from the CCNA course that we just didn’t have time to cover and peeked at a bit of CCNP stuff. You know how CISCO is with their tests: to pass a test over 1-4 you need to know 5 and 6 too. By now you should have a good understanding of the design aspects for networks with CISCO equipment.

Future additions:
  What’s Next?
  IPX, appletalk, more protocols (BGP, IS-IS, et al.)
  Frame and ISDN
  Switching and VLAN’s
  Broken Labs
  Make that stuff up lab
  Troubleshooting
  WECIL’s

Notes for me:
  Floating static (permanent command)
Routing Review

Routing Protocols
1. What are three attributes of distance vector routing protocols?
2. Know how to do VLSM.
3. What is the difference between RIPv1 and RIPv2?
4. What do RIPv1 and IGRP have in common?
5. What does EIGRP use for establishing and maintaining neighbors?
6. How do you configure EIGRP on a router?
7. What are three features of using EIGRP on a router?
8. When should EIGRP’s auto summarization feature be disabled?
9. How can you “force” the election of a DR?
10. How does a router advertise an IGRP route that has been redistributed into an EIGRP network in a routing table?
11. Can you use the same autonomous system number for all areas in a network that uses both EIGRP and IGRP (in other words use #38 for EIGRP and IGRP at the same time)?
12. What is route summarization and how does it work?
13. What is a metric? How can you see a metric in a routing table?
14. What is a static route and how is it configured?
15. What is a dynamic route and how is it configured?
16. What is a gateway of last resort and how is it configured?
17. What is a quad-zero route and how is it configured?
18. What are the basic commands for OSPF configuration?
19. Why would you use authentication with OSPF information exchanges?

RIP
1. What are the limit(s) of using RIPv1 that are overcome by using RIPv2?
2. What are some of the advantages and disadvantages to using RIP?
3. How often does RIP send a broadcast or multicast?
4. What are the steps to configuring RIP on a router?

Routing
1. What is the difference between a unicast, a broadcast, and a multicast?
2. What are the minimum steps for establishing routing?
3. There is a command that is “enabled” or “configured” by default on Cisco routers IOS 12.x and higher called “no ip subnet-zero.” What does this command provide, what happens when it is disabled, and how do you disable it?
4. Draw the “VLSM” cheat sheets for using subnet masks borrowing 3-bits and 4-bits.
5. Why do we have VLSM?
6. What is the Cisco command for displaying a “routing table?”
7. What is sent during an update or route broadcast or multicast?
8. How does a router know how to “send” information to other places?
9. What is route summarization?
10. In a class “C” IP address what is the largest and smallest subnet masks that can be had only when using VLSM (not the default)?
Troubleshooting scenarios for Part 3

Here is just a “small” list of the items I might mess with on a troubleshooting test related to this section:

- Bad straight through cable
- Bad console cable
- Unhooked straight through cable
- Unhooked console cable
- Reversed DCE/DTE cable
- Change passwords
- Change RIP to RIPv2
- Change RIPv2 to RIP
- Remove RIP
- Remove IP host list
- No clockrate on Serial DCE
- Remove IP from Serial Interface
- Remove IP from Eth Interface
- Change mask on serial interface
- Change mask on eth interface
- Change ip on workstation
- Remove gateway from host
- Remove Loopback
- Change RIP metrics
- Remove ip split horizon
- Remove ip subnet-zero
- Change baud of router
- Change ip hostname (for ping)
- Remove static line
- Change subnet mask to summarize
- Remove ip helper address
**Objective:**
To list all commands utilized in Part3 of this textbook.

**Step-by-Step Instructions:**
1. For each of the commands give a description of the command, the prompt for configuration, and any abbreviations for that command. Put it any commands not already in here too. The more, the better.

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Command</th>
<th>Shortcut</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>setup</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>help</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>enable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>disable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>exit</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>history</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>show</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>show configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>terminal</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>hostname</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>copy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>running-configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>start-configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>write memory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>show buffers</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>show flash</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>show interface</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>show memory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>show protocols</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>show processes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>show run</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>show start</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>show stacks</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>show tech</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>show version</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>password cisco</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>login</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>line vty</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>line con</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>line aux</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>logging synchronous</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>exec-timeout</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>enable password</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prompt</td>
<td>Command</td>
<td>Shortcut</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>enable secret</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip host</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no shut</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface e0/0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>clockrate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>router rip</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>network</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ping</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>traceroute</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>show cdp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sh ip route</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>debug ip icmp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>debug ip rip</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>undebug</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>show controller s0/0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface loopback 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>telnet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>show sessions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip dhcp pool</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip dhcp excluded-address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip helper-address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>default-router</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip route</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>version 2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Whole Enchilada/Crazy Insano Lab #1 (WECIL): Routing

Objectives:
To give you an idea of what a practical exam may be designed like to encompass all of the objectives from this part.

Lab Design:

You are the network administrator for a medium-sized manufacturing company in Atlanta. They house all of their operations in three buildings on a city block. Each building has a router and six switches. There is a connection from one building to the ISP that is also used as a DHCP router/server. The ISP has assigned you to the 212.14.39.253/30. The ISP serial interface provides clocking and has an IP address of 212.14.39.254/30. Your task is to design an addressing scheme using private IP addresses. You will need to implement a routing protocol that allows workstation A to be able to ping to workstation B. As an extra measure of security you should not allow your networks to advertise themselves outside of your network. Both workstations should receive their IP address from the DHCP router/server. Since your equipment is limited use a logical interface to emulate the other switches on each router. Both workstations should be able to ping to every logical interface and to 172.16.1.1

Variants:
Class “A,” “B,” or “C” private IP addresses only.
Mixed “A,” “B,” or “C” private IP addresses.
Class “A,” “B,” or “C” public IP addresses only.
Mixed “A,” “B,” or “C” public IP addresses.
Mixed public and private addressing.
Design a network using VLSM and then anyone of the above scenarios. Design a network addressing scheme that summarizes addresses on one of the routers. Change metrics on routers.

*Lab Design:*

![Network Diagram]

**ISP**

WWW 172.16.1.1/16

dhcp

Workstation “A”

Workstation “B”

More options with this design:
- Force path selection with static routes.
- Add a routing loop.
- Force path selection on a network with a routing loop using dynamic routes.
Make part of your network RIP and part of it RIPv2.
Whole Enchilada/Crazy Insano Lab #3 (WECIL): IGRP/RIP

Objective:
To put all or most of the concepts together into one large lab. In this lab we will be mixing IGRP and RIP. Basically picture yourself working for a company with a large IGRP network on two VLAN’s. Recently your company just acquired a company with several hundred hosts using static RIP addressing on the 192.168.x.x private network. You don’t have time to change all those static addresses so you decide to just redistribute everything. You would like to restrict those RIP workstations from being able to telnet and ping to your network though. Don’t forget about your good planning by making redundant backup lines between your switches. Your IGRP network receives its addresses via DHCP from your border router. Hang several loopback interfaces on the back side of the ISP addressed with 172.16.1.1 to 172.16.1.10. Think of the odd-numbered loopbacks as evil workstations smurfing your network. Write an ACL to keep those odd ones from being able to smurf your network. Oh yeah. You will need to make up your own addresses.

Tools and Materials:
(3) routers
(6) switches
(11) straight-through cables
(10) cross-over cables
(8) workstations

Lab Design:
Whole Enchilada/Crazy Insano Lab #4 (WECIL): IP/IPX

Objective:
To put all or most of the concepts together into one large lab. In this lab we will be mixing IP and IPX. Basically picture yourself working for ABC company with a large IGRP network on two VLAN’s. Set up your company to use static RIP addressing on the 192.168.x.x private network. You would like to restrict those all workstations from being able to telnet and ping except for one subnet for you (network administrator). Don’t forget about your good planning by making redundant backup lines between your switches. Hang a loopback interface with a 172.16.1.1 address to test ping from the workstations. Oh yeah. You will need to make up your own IPX addresses that are in use on VLAN2. Those are the accountants using Novell 4.11.

Tools and Materials:
(2) routers
(5) switches
(10) straight-through cables
(10) cross-over cables
(8) workstations

Lab Design:
Part 4: 
WAN Routing Fundamentals 

They said “I was crazy!”

How do you like the idea of free and low-cost textbooks? Well here is how you can help me in the effort to bring more of these types of books to you. They say there is safety in numbers, well I can promote this type of textbook production way better if I hear from you personally. Send me an email at Basham.Matt@spcollege.edu just to tell me you are using the book, you like the idea of cheaper high-quality books, or to just say “hello.” Together we will change education for the better!
Remote Access to a Router with AUX (and Banners)

**Objective:**
To be able to access a router using dial-up networking (DUN).

**Tools and Materials:**
(2) Workstations  
(2) modems  
(1) DB-9 to RJ-45 adapter  
(2) RS-232 to RJ-45 adapter  
(1) Straight-through cable  
(2) rollover cables  
(2) RJ-11 (phone lines)  
(1) Adtran 550 with Octal ports

**Lab Diagram:**

```
com1:DB-9 to RJ-45  
RJ-45 to RS-232  
RJ-11  
PSTN  
Matt  
ST  
555-6001  
Router  
555-6002  
```

**Step-By-Step Instructions:**
1. Set up and cable the lab as shown.  
2. Check to see which line number is used for dial-in connections:

   ```
   Router>sh line
   
   You should see(I cut-off the stuff on the right):
   Tyt Typ Tx/Rx A Modem Roty
   *  0  CTY    --  --  --
   *  65  AUX 19200/19200  inout  --  --
   66 VTY    --  --  --
   67 VTY    --  --  --
   68 VTY    --  --  --
   69 VTY    --  --  --
   70 VTY    --  --  --
   Line(s) not in async mode -or- with no hardware support: 1-64
   ```
3. Configure the router to receive incoming calls.

   Router(config)#line aux 0  (or line 65)
   Router(config-line)#login
   Router(config-line)#password auxpass
   Router(config-line)#speed 115200
   Router(config-line)#flowcontrol hardware
   Router(config-line)#stopbits 1
   Router(config-line)#transport input all
   Router(config-line)#modem inout
   Router(config-line)#modem autoconfigure discovery

   The last line will attempt to discover your modem type automatically. Probably not needed but nice to have.

4. To troubleshoot a connection use “debug modem” on the router and establish the connection.

5. On the PC dial into the router using Hyperterminal. You will be prompted for a password. If you are successful then you should see the user mode prompt.

6. You may want to have a message appear when someone accesses your router. Some people are very friendly and make a banner like:

   “Welcome to the ABC network.”

   Wrong answer recruit…a banner like this is like a welcome mat being thrown out. In fact a case where a “defendant” hacked into a router was thrown out because the administrator had a banner like the one above. In short, don’t welcome me in, if I am not supposed to be there. You will probably want to make one more like:

   “WARNING: Authorized admittance only. Unauthorized entrance will be prosecuted to the fullest extent of the law.”

   Or something like that…if you have a corporate lawyer then have them come up with one…they live for that stuff.

7. So let’s get by the legal mumbo-jumbo and put up a login banner. You have many different ways to do this…let’s find out…

   Router#banner ?

   You should see:
   LINE         c banner-text c, where ‘c’ is a delimiting character
   Exec         set EXEC process creation banner
   Incoming     set incoming terminal line banner
   Login        set login banner
   Motd         Set message of the day banner
8. We simply type our command, subcommand, the letter ‘c,’ our message, then another ‘c’ to end it:

   Router(config)#banner login c stay out or get prosecuted c

9. So which subcommand do we pick? Login? Motd? Right now it really does not matter...they will all just about do the same thing.

Supplemental Lab or Activities:
1. Try setting up a dial-in connection on a serial port. You will need a different cable from your modem to router and some commands on the serial port. Try it.
2. Try using DUN to access the router. Where and why does it crap out?

So What Did I Learn Here?
In this lab you learned how to dial into the AUX port of a router from home (or somewhere). This lab is a good transition into the remote access class later. There you will learn about reverse telnet and modem strings with routers and stuff like that. For now let’s call it quits with dial up and move over to the serial interfaces and WAN connections.
Remote Access Lab

Objective: To learn how to add a modem to a workstation and how to set up a dial up network. To make this lab work will require a three-step process: (1) to install a modem, (2) to create a dial up connection, and (3) to allow in-coming calls. You may also have to disable any security settings, like the Windows firewall to make this lab work.

Materials:
2 Modems
3 telephones
5 telephone cables
2 Modem-to-PC cables
Adtran Atlas 550 with POTS card

PC ----modem----adtran----modem----PC

| telephone | telephone |
| telephone |

in the classroom set up:

server 1----modem----adtran----(student connection)
server 2----modem----/ /
server 3----modem------/

Step 1 to install the Modem:
(skip this step if you already Install the modem software onto the PC if you haven’t already done so. If you have, or are not sure, just right click on the “My Network Places” icon on the desktop and select “Properties.” If you have one it will show up as a “Dial up” connection. Here is my PC with a complete lab set up:
Here are the steps for adding in a modem (in case your PC doesn’t detect it and auto-install it). First open the control panel. Then select “Add hardware.” You should see the wizard pop-up (Click on “next”):

Click on install the hardware from a list and then select Modems.

Let the PC auto-detect the modem and then install the software manually.
Choose the don’t search option and select modems.

Select the Hayes Accura 56k Ext. Fax Modem 4703US (or which ever brand you are using).

Select “Yes” to continue. Check with your instructor first.
The modem auto install will run and then pick “finish.”

Step 2 to Create a Connection:

The next step in firing this puppy up is to create a connection that will allow you to dial out from the PC. Right click on the “My Network Places” icon and select properties. Then select create a new connection and the new connection wizard should appear. Select “next” to begin.

Select connect to the network at my workplace and dial up connection.
Type in a name you wish to give the connection, then put it the phone number (do not use any dashes, hyphens, etc.)

Click “finish” to end the wizard. The new connection should pop up next. You have now created your connection out of the PC.

**Step 3 to Create an Incoming Connection:**
The last part here is to allow calls to be received. Otherwise you will only hear your modem dialing the other modem, and then ringing and ringing and ringing. So, we have to set your PC up to receive calls. Right click on the “My Network Places” icon and select properties. Then select create a new connection and the new connection wizard should appear. Select “next” to begin.
This time select set up an advanced connection and then accept incoming connections.

Select your modem and, if you want to, select allowing virtual private connections.
Now, here is the tricky part…you have to select which users you wish to be allowed to dial into your PC. Your school might have all kinds of stuff locked down. I find it best to create your own user and password combo to make this lab run smoother:

Then select next and finish to end this part. You now have three components, that together, will allow you to dial out and receive calls (from selected individuals) on your Windows 2000 PC.
**Troubleshooting tips:**
If you run into any problems along the way you can always test your dial up connections from phone-to-phone. If that portion dials and rings correctly then you can eliminate the modems, adtrans, and telephone cables as a source of the problem. Nine times out of ten the PC is the culprit. You may also need to shut down everything, boot up the Adtran (wait for it), and then start powering up everything else.

In case you get stuck on any of these steps do not hesitate to try using the help feature in Microsoft. Now, if I can only get you to stop and ask for directions.
Asynchronous Connections to a Router
By Michael Gordon

Directions: In the following lab, you will attach an external modem to your router, enable modem autoconfigure, reverse telnet into the modem, and complete a dial up connection to the AUX port of your router from your PC. The dial up environment you are configuring corresponds with the internetworking model detailed below.

Connecting the Modem:
In this procedure, you will connect an external modem to the AUX port of your router.

1. Plug in to modems power supply and power on the modem.
2. Connect the DB-9 male interface of the serial cable to the DB-9 female connector on the back of the modem.
3. Connect the RJ-45-to-DB-25 male connector (labeled “MODEM”) to the DB-25 female connector on the other end of the serial cable.
4. Connect an RJ-45 rollover cable between the connector (labeled “MODEM”) and the AUX port of your router.
5. Connect an RJ-11 patch cable between the external modem’s line port and any empty port on the POTS line simulator.
6. Connect an RJ-11 patch cable between the PC's internal modem’s line port and any empty port on the POTS line simulator.

Configuring the AUX Port for Use of a Modem:
In this procedure, you will configure the characteristics of the AUX port (line 1) on the router, including authentication to protect the router from unauthorized dial in access.

7. Telnet into your router.
8. In config mode, type **line aux 0** to enter line config mode.
9. Type **modem inout** to configure the line for both incoming and outgoing use of a modem.
10. Type **flowcontrol hardware** to use RTS/CTS for flow control.
11. Type **stopbits 1** to set the number of stopbits transmitted per byte.
12. Type **login**.
13. Type **password cisco** to set a password on the AUX port.
14. Exit config mode.
**Configuring the External Modem:**
In this procedure, you will enable modem autoconfigure discovery on your router, which will allow your router to identify the modem and issue it the appropriate AT command string. Once this process is completed, you will save the configuration to the modems NVRAM and disable modem autoconfigure on the router to reduce overhead, as it is no longer necessary.

15. Type `debug confmodem` to display the modem configuration process.
16. Type `terminal monitor` to display debugging output on the terminal.
17. Enter `config` mode and type `int loopback 0` to create a loopback interface.
18. Type `ip address 1.1.1.1 255.255.255.255` to address the loopback interface.
19. Type `line aux 0` to enter line config mode.
20. Type `transport input all` to allow all protocols to be passed to the access server across this line.
21. Type `modem autoconfigure discovery` to enable modem autoconfigure.
22. Exit config mode.

Be patient. It may take a minute for debugging should yield the following output:

Notice the following:
- the line speed is detected as 38400 (max speed for AUX port).
- the modem type is detected as `default`.
- the AT command string for this modem is issued to the modem.
- modem configuration is successful.

TTY1: detection speed **38400** response --OK---
TTY1: Modem type is **default**
TTY1: Modem command: --AT&F&C1&D2S0=1H0--
TTY1: Modem configuration **succeeded**
TTY1: Detected modem speed 38400
TTY1: Done with modem configuration

22. Type `u all` to turn off all debugging.
23. Type `show line` to identify the line used by the AUX port.

Notice the following:
- the AUX port is using line 1.

TTY Typ Tx/Rx A Modem Roty AccO AccI Uses Noise Overruns Int
0 CTY - - - - - - 0 0 0/0 -
1 AUX 38400/38400 - - - - - - 0 1 0/0 -
*2 VTY - - - - - - 1 0 0/0 -
3 VTY - - - - - - 0 0 0/0 -
4 VTY - - - - - - 0 0 0/0 -
5 VTY - - - - - - 0 0 0/0 -
6 VTY - - - - - - 0 0 0/0 -

Keep track of updates and changes at [http://www.spcollege.edu/star/cisco](http://www.spcollege.edu/star/cisco) Scroll to the bottom of the page and click on the “Lab Manual Edits.”
24. Type `telnet 1.1.1.1 2001` to reverse telnet into your modem.

You should get a response similar to the output below:

```
Trying 1.1.1.1, 2001 ... Open
```

25. At the Password: prompt, type `cisco`.

You should get a response of `Password OK`.

26. Type `at&w` and press enter to save the current configuration to modem NRVAM.

You should get a response of `OK`.

27. Type `(ctrl+shift+6` followed by `x) to exit the reverse Telnet session.

Your router’s prompt should reappear.

28. Type `disconnect` to end the reverse Telnet session.

29. When prompted to confirm closing the connection, press Enter.

30. Enter `config` mode and type `line aux 0` to enter line config mode.

31. Type `no modem autoconfigure` to disable modem autoconfigure, as it is no longer necessary.

32. Close the Telnet session.

**Completing a Call to the Router:**

In this procedure, you will dial into the router from your PC. This will give you remote access to manage your router from any dialup location.

33. Disconnect the Ethernet cable connecting your PC to the router over the LAN.

34. Start Hyperterminal.

35. Enter your name in the Connection Description dialog box and click OK.

36. The Connect to dialog box will appear:

   - In the Phone number: box, enter the phone number of the line that the router’s external modem is connected to. (either 101, 102, 103, or 104)
   - In the Connect using: box, make sure the modem is selected.
   - Click OK.

37. The Connect dialog box will appear. Click Dial.

The PC should dial your router’s modem. You should hear the modems negotiate the connection and a password prompt should appear.

```
If the connection succeeds but no prompt appears, make sure you have disconnected your previous reverse Telnet session to the external modem.
```

38. Type `cisco` to enter user mode. The router user mode prompt should appear.

39. Proceed as usual to enter privileged mode. You have full access to the router via the asynchronous connection.

40. When you are ready to disconnect, just close the Hyperterminal session.
Configuring a Terminal Server
By Michael Gordon

**Directions:** In the following lab, you will configure remote access to multiple routers through a single asynchronous connection to a router providing *terminal services*, such as a Cisco 2509 or 2511. After completing a dial up connection to the terminal server, you will be able to use reverse Telnet to access all other connected routers via the TTY (asynchronous) ports on the terminal server. Without a terminal server, management of each individual router would require a separate analog line and modem for each.

The dial up environment you are configuring corresponds with the internetworking model detailed below.

---

**Connecting the Modem and Terminal Server:**
In this procedure, you will connect an external modem to the terminal server and connect the asynchronous lines from the terminal server to the other routers.

1. Connect the modem to the AUX port of the terminal server, as you have done in the previous lab.
2. Using RJ-11 patch cable, connect the terminal server’s external modem to the first empty port on the POTS line simulator.
3. Using RJ-11 patch cable, connect your PC’s internal modem to any other empty port on the POTS line simulator.
4. Connect the octal cable to ASYNC 1-8 port on the access server.
5. Connect ASYNC – Line 1 of the octal cable to the console port of RouterA.
6. Connect ASYNC – Line 2 of the octal cable to the console port of RouterB.
7. Connect ASYNC – Line 3 of the octal cable to the console port of RouterC.

**Configuring the AUX Port and TTY Ports:**
In this procedure, you will configure the characteristics of the AUX port (line 17) and TTY ports (lines 1,2,3) on the access server.

8. Console into the access server via Hyperterminal.
9. In `config` mode, type `line aux 0` to enter line config mode.
10. Type `transport input telnet` to allow the Telnet protocol to be passed to the access server across this line.
11. Type `modem inout` to configure the line for both incoming and outgoing use of a modem.
12. Type **flowcontrol hardware** to use RTS/CTS for flow control.
13. Type **stopbits 1** to set the number of stopbits transmitted per byte.
14. Type **login**.
15. Type **password cisco** to set a password on the AUX port.
16. Type **line 1 3** to enter line config mode for TTY ports 1-3.
17. Type **no exec** to prevent the port from receiving unsolicited data, which could make the line unavailable. This command will allow only outgoing connections for the line.
18. Type **transport input telnet**.

**Configuring a Loopback Interface and Host Table:**

When you have multiple devices connected to a terminal server, it can be simpler to remember host names than the line numbers each device is connected to. In this procedure, you will statically map hostnames to IP addresses, to ease router access through the terminal server.

19. Type **int loopback 0** to create a loopback interface.
20. Type **ip address 1.1.1.1 255.255.255.255** to address the loopback interface.
21. Type **exit** to exit interface config mode.
22. Type **ip host routera 2001 1.1.1.1** to create a host table entry for RouterA.
23. Type **ip host routerb 2002 1.1.1.1** to create a host table entry for RouterB.
24. Type **ip host routerc 2003 1.1.1.1** to create a host table entry for RouterC.
25. Exit config mode.

**Configuring the Console Ports:**

In this procedure, you will configure the console port on each router to allow the Telnet protocol to be used on the line.

26. Telnet into your router.
27. Enter **config** mode, and type **line con 0** to enter line config mode.
28. Type **transport input telnet** to allow the Telnet protocol to be passed to the access server across this line.
29. Exit config mode and end the Telnet session.
Completing a Call to the Terminal Server:
In this procedure, you will dial into the terminal server from your PC. From there, you will reverse Telnet into the individual connected routers.

- Since there is only one line to the terminal server (101), only one group can dial in at a time. You will have to take turns completing these last sections.

30. Disconnect the Ethernet cable connecting your PC to the router over the LAN.
31. Start Hyperterminal.
32. Enter your name in the Connection Description dialog box and click OK.
33. The Connect to dialog box will appear:
   - In the Phone number: box, enter the phone number of 101.
   - In the Connect using: box, make sure the modem is selected.
   - Click OK.
34. The Connect dialog box will appear. Click Dial. The PC should dial your router’s modem. You should hear the modems negotiate the connection and a password prompt should appear.
35. Type cisco to enter user mode. The user mode router prompt should appear.
36. Type `telnet routera` to reverse Telnet into RouterA. You should get a response similar to the output below:

   Trying routera (1.1.1.1, 2001) ... Open

38. Type `ctrl+shift+6 followed by x` to exit the reverse Telnet session. The router prompt should reappear for the terminal server.
39. Type `telnet routerb` to reverse Telnet into RouterB.
41. Type `ctrl+shift+6 followed by x` to exit the reverse Telnet session. The router prompt should reappear for the terminal server.
42. Type `telnet routerc` to reverse Telnet into RouterC.
43. Press Enter. A prompt for RouterC should appear.
44. Type `ctrl+shift+6 followed by x` to exit the reverse Telnet session. The router prompt should reappear for the terminal server.

Switching Between Sessions:
At this point, you have several reverse Telnet sessions open at once. In this procedure, you will easily switch between sessions by indicating the line number.

45. Type `sh sessions` to view the open reverse Telnet sessions. All open sessions associated with the terminal line will be displayed.
46. Type `1` to reenter the session on line 1. You should get a response similar to the output below:

   [Resuming connection 1 to routera ... ]

47. Press Enter. A prompt for RouterA should appear.
48. Type (ctrl+shift+6 followed by x) to exit the reverse Telnet session. The router prompt should reappear for the terminal server.
49. Type disconnect 2 to close the connection to RouterB.
50. Type show sessions. Only the open sessions to RouterA and RouterC remain.
51. Close Hyperterminal.

At this point, the next group may dial in to the terminal server and complete the lab.

Looking for college courses?
We deliver.

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education.

Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

St. Petersburg College’s eCampus
http://e.spcollege.edu

Your time. Your place. Your future.
Objective:
To learn more about serial line encapsulation types:

Tools and Materials:
(2) Workstations
(2) Console cables
(2) DTE cables
(2) DCE cables
(3) Routers

Lab Diagram:

<table>
<thead>
<tr>
<th>Name</th>
<th>S0/0</th>
<th>S0/1</th>
<th>L0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leftist</td>
<td>201.200.200.1/24</td>
<td>201.200.200.2/24</td>
<td>12.0.0.1/8</td>
</tr>
<tr>
<td>Urvile</td>
<td>n/a</td>
<td>n/a</td>
<td>12.0.0.1/8</td>
</tr>
</tbody>
</table>

Background:
Back in part 2 we learned the default encapsulation type on a serial line is HDLC. This is CISCO’s proprietary “Serial HDLC Synchronous” line protocol. Needless to say it does not always work well with non-CISCO devices. For example, IBM routers would need to use SDLC for its serial line encapsulations. So how do we know what encapsulations are available to us? That’s easy…we just need to use our handy-dandy help feature at the right moment on the router. So let’s look:

Router(config)# int s0/0
Router(config-if)# enc ?
Encapsulations on serial lines are easy. What you have set on one end, you must have the same set on the other otherwise no communication can take place.

**Step-By-Step Instructions:**

1. Set up the lab and cable it as shown. Use EIGRP as your routing protocol. Use the same autonomous number for each network.
2. Ping from the router prompt of Terminus to Leftist and then to Urvile. It should work jiffy spiffy-like. Do a trace route between them.
3. Now change the encapsulation on Terminus S0/0 to PPP:

   terminus(config)#int s0/0
   terminus(config-if)#enc ppp

4. Ping from the router prompt of Terminus to Leftist and then from Terminus to Urvile (loopback). It should not work so jiffy spiffy-like. You should see:

   terminus#ping 200.200.200.2
   Type escape sequence to abort.
   Sending 5, 100-byte ICMP Echos to 200.200.200.2, timeout is 2 seconds:
   ....
   Success rate is 0 percent (0/5)

Do a trace route between them. You should not get anywhere because the encapsulation types have to be the same on both ends in order to communicate. Then change the encapsulation on S0/1 of leftist. Let’s change the encapsulation type on s0/1 on leftist. Verify your encapsulation with “show interface.” You should see:

```
Serial0/0 is up, line protocol is up
Hardware is PowerQUICC Serial
Internet address is 200.200.200.1/24
MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
   reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, loopback not set
Keepalive set (10 sec)
LCP Open
```
Open: IPCP, CDPCP
Last input 00:00:01, output 00:00:04, output hang never
Last clearing of "show interface" counters 00:03:45
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
81 packets input, 6663 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
1 input errors, 0 CRC, 1 frame, 0 overrun, 0 ignored,
0 abort
85 packets output, 7435 bytes, 0 underruns
0 output errors, 0 collisions, 2 interface resets
0 output buffer failures, 0 output buffers swapped out
8 carrier transitions
DCD=up  DSR=up  DTR=up  RTS=up  CTS=up

5. Now that the encapsulation types match on each end of the serial line, ping from the router prompt of Terminus to Leftist. It should work just fine. You should see:

    terminus# ping 200.200.200.2
    Type escape sequence to abort.
    Sending 5, 100-byte ICMP Echos to 200.200.200.2, timeout is 2 seconds:
    !!!!!
    Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms

Ping from Terminus to Urvile. Initially you may think it should not work so jiffy spiffy-like because you have PPP as an encapsulation on one serial line and HDLC as the encapsulation on the other line. But since we have the same encapsulation on each end of the serial line we can mix and match encapsulations over the entire network. Geeeze. If we could not then the entire Internet would have to run on only one encapsulation type.

You should see:

    terminus# ping 12.0.0.1
    Type escape sequence to abort.
    Sending 5, 100-byte ICMP Echos to 12.0.0.1, timeout is 2 seconds:!!!!!
    Success rate is 100 percent (5/5), round-trip min/avg/max = 64/64/68 ms

Do a trace route between the three. You should see:

    terminus# traceroute 12.0.0.1
    Type escape sequence to abort.
    Tracing the route to 12.0.0.1
    1  200.200.200.2 16 msec 16 msec 16 msec
    2  201.200.200.2 32 msec 32 msec *
Let’s also look at our ip route:

```
terminus# sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
       inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

200.200.200.0/24 is variably subnetted, 2 subnets, 2 masks
C       200.200.200.0/24 is directly connected, Serial0/0
C       200.200.200.2/32 is directly connected, Serial0/0
D    201.200.200.0/24 [90/21024000] via 200.200.200.2, 00:02:23, S0/0
C    10.0.0.0/8 is directly connected, Loopback0
D    11.0.0.0/8 [90/20640000] via 200.200.200.2, 00:02:23, Serial0/0
D    12.0.0.0/8 [90/21152000] via 200.200.200.2, 00:02:23, Serial0/0
```

6. To change the encapsulation back we would just reverse the process and use HDLC:

```
terminus(config)# int s0/0
terminus(config-if)# enc hdlc
```

7. You can change all serial interfaces to PPP for its encapsulation and it should work just fine. Remember: it’s got to be the same on both ends to work.

**Challenge Lab or Supplemental Activities:**

1. Go to the web and find out for what all those other encapsulation types are primarily used.
2. Can SDLC-primary on one end of a serial line communicate with SDLC-secondary on the other end?

**So what have I learned here?**

In this lab you have learned there are many different encapsulation types on a serial interface and that CISCO routers use HDLC by default. Other manufactures use different encapsulations, for example IBM routers use SDLC for their encapsulations by default. Why is this lab here and not in part 2? PPP allows us to set authentication parameters (ew! Geek-speak). In “real-people” talk this means we can set user names and passwords for people to “dial-in” (aha! Remote access=WAN technology) to our serial lines. Remember our serial lines typically run over the web or telephone lines over great distances. This usually means security is very important (refer to guest names below). In the next lab you will learn how to set up those user names and passwords with PPP.
Guest Router Name Derivation

Terminus, Leftist, and Urville were three hackers from the Legion of Doom, who lived in Georgia, that were busted in 1990 by the U.S. Secret Service in connection with the Martin Luther King Day AT&T long distance network crash. They were known as “switching gurus” and as “heavy hitters” within the LoD because they frequently accessed BellSouth’s network. Apparently BellSouth, at that time, did not have very strict security in place.

PPP Frame Format

<table>
<thead>
<tr>
<th>Preamble</th>
<th>SOF</th>
<th>Flag</th>
<th>Address</th>
<th>Control</th>
<th>Protocol</th>
<th>Data</th>
<th>FCS</th>
<th>Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up to 63</td>
<td>1/4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Var.</td>
<td>2 or 4</td>
<td>1</td>
</tr>
</tbody>
</table>

Preamble always a sequence of 01010101 (not seen with protocol inspectors)
SOF always 11 to indicate the “start of frame” (not seen)
Flag always 0111110 to indicate the start of a “PPP frame” (0x7e in hex)
Address always 1111111 (we are dealing with direct connections; 0xff in hex)
Control used in HDLC; with PPP it is always 00000011 (0x03 in hex)
Protocol used to identify the network protocol in use over PPP link
Data variable
FCS error control mechanism
Flag always 0111110 to indicate the end of a “PPP frame” (0x7e in hex)
Authentication with PPP

**Objective:**
To learn more about PPP’s authentication methods: PAP and CHAP

**Tools and Materials:**
(2) Workstations
(2) Console cables
(2) DTE cables
(2) DCE cables
(3) Routers

**Lab Diagram:**

![Lab Diagram](image)

<table>
<thead>
<tr>
<th>Name</th>
<th>S0/0</th>
<th>S0/1</th>
<th>L0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminus</td>
<td>200.200.200.1/24</td>
<td>201.200.200.1/24</td>
<td>n/a</td>
</tr>
<tr>
<td>Leftist</td>
<td>n/a</td>
<td>200.200.200.2/24</td>
<td>201.200.200.2/24</td>
</tr>
<tr>
<td>Urville</td>
<td>10.0.0.1/8</td>
<td>11.0.0.1/8</td>
<td>12.0.0.1/8</td>
</tr>
</tbody>
</table>

**Background:**
In the last lab you learned about different encapsulations on serial lines. In this lab you will delve more deeply into the PPP encapsulation. PPP can use passwords and user names for authentication over serial lines before communication can take place. Here you will learn how PPP works, how to configure PPP authentication, and troubleshooting tools for PPP. During the establishment of PPP five things can take place:

1. First, the serial line establishment will take place. This is where any negotiation will take place. (LCP—Link Control Protocol)
2. Second, if user names and passwords are used, authentication of those names and passwords will take place.
3. Next, the network layer will negotiate which protocols will be in use during the session. (NCP-Network Control Protocol).
4. Then the line comes up and communication can take place.
5. Finally the link will be terminated after all communication is finished.

You will “see” each of these steps during this lab. When configuring authentication with user names and passwords we have two methods to accomplish this: PAP or CHAP.
PAP (Password Authentication Protocol) uses passwords that are sent in clear text during a two-way handshake process (how secure is that? What is the point?) Basically a remote user requests a connection by sending a username and password request (one part of the two-way handshake) the device to be accessed then processes the information and either accepts or rejects the username and password (the other part of the two-way handshake). PAP only requests username and passwords once.

CHAP (Challenge Handshaking Authentication Protocol) is similar to PAP except the username and passwords are encrypted (much better), a three-way handshake is used, and periodically CHAP re-requests usernames and passwords for authentication. With CHAP a remote user requests a connection (one part of the three-way handshake), the device to be accessed then requests a username and password (the second part of the three-way handshake), the remote user responds with the username and password (still the second part of the three-way handshake), and the device to be accessed then accepts or rejects the username and password (the third part).

You will configure and “see” each of these in this lab.

Step-By-Step Instructions:
1. Set up the lab and cable it as shown. Use EIGRP as your routing protocol. Use the same autonomous number for each network. Use PPP for encapsulation on the serial lines.
2. Ping from the router prompt of Terminus to Leftist and then to Urvile. It should work jiffy spiffy-like. Do a trace route between them to verify connectivity.
3. Now that we know everything works lets look at the default state of PPP (without any user names or passwords):

   terminus#debug ppp tasks

   Then disconnect the serial line for about 10 seconds and then re-connect it. You will see the LCP task negotiation and the line come back up. You should see something like:

   terminus#debug ppp tasks
   (line is disconnected)
   00:52:38: %LINK-3-UPDOWN: Interface Serial0/0, changed state to down
   00:52:39: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to down
   (line is reconnected)
   00:52:49: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
   00:52:49: Se0/0:  AAA_PER_USER  LCP_UP (0x81483B3C) id 0
   (0s.) queued 1/1/1
   00:52:49: Se0/0:  AAA_PER_USER  LCP_UP (0x81483B3C) id 0
   (0s.) busy/0 started 1/1/1
   00:52:49: Se0/0:  AAA_PER_USER  LCP_UP (0x81483B3C) id 0
   (0s.) busy/0 done in 0 s. 0/0/1
   00:52:50: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
Let’s look at what is happening here by “cleaning up our debug” a bit:

```
terminus#debug ppp tasks
(line is disconnected)
Line protocol on Interface Serial0/0, changed state to down
(line is reconnected)
Interface Serial0/0, changed state to up
AAA_PER_USER     LCP_UP
AAA_PER_USER     LCP_UP
AAA_PER_USER     LCP_UP
Line protocol on Interface Serial0/0, changed state to up
```

We can see that when our line is disconnected no “task” packets are communicated over the ppp line. But we do have LCP task packets being communicated when the line comes back “up.” Remember our PPP five step process: line is reconnected, LCP is negotiated, any username/passwords are verified, NCP is negotiated, line comes up and communication takes place, and the session is terminated. With the debug tasks we can only see LCP packets.

Next we can look at the actual negotiation steps with debug. Be sure to turn off all debugging so we get a “clear” debug ppp negotiation. You should see:

```
terminus#undebug ppp tasks
PPP background processing debugging is off
terminus#debug ppp negotiation
PPP protocol negotiation debugging is on
(line is disconnected)
00:54:27: %LINK-3-UPDOWN: Interface Serial0/0, changed state to down
00:54:27: Se0/0 IPCP: State is Closed
00:54:27: Se0/0 CDPCP: State is Closed
00:54:27: Se0/0 PPP: Phase is TERMINATING
00:54:27: Se0/0 LCP: State is Closed
00:54:27: Se0/0 PPP: Phase is DOWN
00:54:27: Se0/0 IPCP: Remove route to 200.200.200.2
00:54:28: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to down
(line is reconnected)
00:54:36: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
00:54:36: Se0/0 PPP: Treating connection as a dedicated line
00:54:36: Se0/0 LCP: O CONFREQ [Closed] id 3 len 10
00:54:36: Se0/0 LCP: MagicNumber 0x04158C08 (0x050604158C08)
00:54:36: Se0/0 LCP: I CONFREQ [REQsent] id 4 len 10
00:54:36: Se0/0 LCP: MagicNumber 0x01C88BB2 (0x050601C88BB2)
00:54:36: Se0/0 LCP: O CONFACK [REQsent] id 4 len 10
00:54:36: Se0/0 LCP: MagicNumber 0x01C88BB2 (0x050601C88BB2)
00:54:36: Se0/0 LCP: I CONFACK [ACKsent] id 3 len 10
```
Ok…all those numbers and stuff can be confusing. Let’s strip that output down to just the information in yellow (the greyed in part if you have the book) and see what is happening (I put the numbers in for easier explanation):

1. debug ppp negotiation
2. (line is disconnected)
3. Interface Serial0/0, changed state to down
4. IPCP: State is Closed
5. CDPCP: State is Closed
6. PPP: Phase is TERMINATING
7. LCP: State is Closed
8. PPP: Phase is DOWN
9. IPCP: Remove route to 200.200.200.2
10. Line protocol on Interface Serial10/0, changed state to down
11. (line is reconnected)
12. Interface Serial10/0, changed state to up
13. PPP: Phase is ESTABLISHING, Active Open
14. CONFREQ [Closed]
15. LCP: I CONFREQ [REQsent]
16. LCP: O CONFACK [REQsent]
17. LCP: I CONFACK [ACKsent]
18. LCP: State is Open
19. PPP: Phase is UP
20. IPCP: O CONFREQ [Closed]
21. IPCP: Address 200.200.200.1
22. CDPCP: O CONFREQ [Closed]
In lines 4-9 we can see what is involved with “tearing down” a connection session. Obviously we would expect to have to re-create those to establish a new PPP session. We see that IPCP went down, then CDPCP, then PPP and finally LCP. Lastly the route was removed. We would expect to see the creation in the reverse order.

In line 13 we see, after our serial line is re-connected, the beginning step of establishing a PPP session.

In 15-18 we see our LCP negotiation phase:
1. a request from s0/0 to s0/1 (line 15),
2. the acknowledgement of that request from s0/1 that s0/0 wants to establish LCP (line 16),
3. then the acknowledgement of s0/0 that s0/1 received the request from s0/0 to establish an LCP (line 17).
4. Then LCP is “open” (line 18).

Then we see our PPP “phase” is set to up in line 19. Then we see our IPCP and CDPCP being brought back up just about the same time in the remainder of our script. During the IPCP:
1. We see s0/0 send it’s ip address (200.200.200.1) to s0/1 (lines 20-21)
2. Then s0/1 sends it’s ip address (200.200.200.2) to s0/0 (lines 23-24)
3. Then s0/0 sends an acknowledgement to s0/1 that it received the ip address from s0/1 (lines 25-26)
4. Then s0/1 sends an acknowledgement to s0/0 that it received the ip address from s0/0 (lines 29-30)
5. Finally the route is established (line 34)

During the CDPCP:
1. a CDPCP configuration request is sent from s0/0 to s0/1 (line 22/27)
2. an acknowledgement of receipt of that request is sent from s0/1 to s0/0 (line 28).
3. an acknowledgement of receiving that acknowledgement is sent from s0/0 to s0/1 (line 32).
4. The CDPCP state is set to “open.”
Our order has reversed itself and our connection, via PPP encapsulation, is now ready to communicate!

4. Let’s turn off debugging. Use “undebug all” or “undebug ppp.”

5. Now let’s set up PPP with PAP authentication. Just remember with our encapsulations on serial lines what we do on one end we must do on the other end too. If you just use “ppp authentication pap” you will not be able to have a ppp connection because no username/password authentication will be able to take place.

```plaintext
terminus(config)#int s0/0
terminus(config-if)#enc ppp
terminus(config-if)#ppp authentication pap
terminus(config-if)#ppp pap sent-username prophet password legodoom
terminus(config-if)#exit
terminus(config)#username prophet password legodoom
```

Before we change the other end of the line let’s look at a “failed” PPP negotiation process. Here we will see s0/1 refusing the connection because we have not set up authentication on it yet (notice how we never make it past the LCP negotiation phase):

```plaintext
terminus#debug ppp negotiation
PPP protocol negotiation debugging is on
terminus#
01:18:42: Se0/0 LCP: I CONFREQ [Listen] id 208 len 10
01:18:42: Se0/0 LCP: MagicNumber 0x01DE98E6
01:18:42: Se0/0 LCP: O CONFREQ [ACKsent] id 109 len 14
01:18:42: Se0/0 LCP: AuthProto PAP (0x0304C023)
01:18:42: Se0/0 LCP: MagicNumber 0x042BA23D
01:18:42: Se0/0 LCP: I CONFREJ [ACKsent] id 109 len 8
01:18:42: Se0/0 LCP: AuthProto PAP (0x0304C023)
01:18:42: Se0/0 PPP: Closing connection because remote won't authenticate
01:18:42: Se0/0 LCP: O TERMREQ [ACKsent] id 111 len 4
01:18:42: Se0/0 LCP: O CONFREQ [TERMsent] id 112 len 14
01:18:42: Se0/0 LCP: AuthProto PAP (0x0304C023)
01:18:42: Se0/0 LCP: MagicNumber 0x042BA23D
01:18:42: Se0/0 LCP: I TERMAK [TERMsent] id 111 len 4
01:18:42: Se0/0 LCP: State is Closed
01:18:42: Se0/0 PPP: Phase is DOWN
01:18:42: Se0/0 PPP: Phase is ESTABLISHING, Passive Open
01:18:42: Se0/0 LCP: State is Listen
01:18:44: Se0/0 LCP: TIMEout: State Listen
```
Watch out! This one can really be tough to stop on your router. Remember your up arrow to quickly find the “undebug all” to try stopping this. You may even have to disconnect the line again to help slow down the debug messages even after you have turned off all debugging. Then do it on the other end of the serial line (router “leftist”):

```
leftist (config)#int s0/1
leftist (config-if)#enc ppp
leftist (config-if)#ppp authentication pap
leftist (config-if)#ppp pap sent-username prophet password legodoom
leftist (config-if)#exit
leftist(config)#username prophet password legodoom
```

As soon as you put in the ppp pap username/passwords you should see something like this:

```
leftist(config)#int s0/1
leftist(config-if)#ppp pap sent-username prophet password legodoom
leftist(config-if)#01:23:27: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed state to up
leftist(config-if)#01:23:29: %LINK-3-UPDOWN: Interface Serial0/1, changed to state to up
leftist(config-if)#
```

Notice the process here…the line protocol comes up then the state comes up.

6. Let’s look at our PAP negotiation process:

1. `leftist#debug ppp negotiation`
2. PPP protocol negotiation debugging is on
3. `leftist#`
4. `01:24:37:%LINK-3-UPDOWN: Interface Serial0/1, changed state to down`
5. `01:24:37: Se0/1 IPCP: State is Closed`
6. `01:24:37: Se0/1 CDPCP: State is Closed`
7. `01:24:37: Se0/1 PPP: Phase is TERMINATING`
8. `01:24:37: Se0/1 LCP: State is Closed`
9. `01:24:37: Se0/1 PPP: Phase is DOWN`
10. `01:24:37: Se0/1 IPCP: Remove route to 200.200.200.1`
11. `01:24:38: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed state to down`
12. `01:24:46: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up`
13. `01:24:46: Se0/1 PPP: Treating connection as a dedicated line`
14. `01:24:46: Se0/1 PPP: Phase is ESTABLISHING, Active Open`
15. `01:24:46: Se0/1 LCP: O CONFREQ [Closed] id 76 len 14`
16. `01:24:46: Se0/1 LCP: AuthProto PAP (0x0304C023)`
17. `01:24:46: Se0/1 LCP: MagicNumber 0x01E48949 (0x050601E48949)`
18. `01:24:46: Se0/1 LCP: I CONFREQ [REQsent] id 187 len 14`
19. `01:24:46: Se0/1 LCP: AuthProto PAP (0x0304C023)`
20. `01:24:46: Se0/1 LCP: MagicNumber 0x04318B4D (0x050604318B4D)`
We see our differences now in lines 28-33 with our username and acknowledgements being displayed.

7. Let’s turn off debugging. Use “undebug all” or “undebug ppp negotiation.”
8. Finally let’s look at our ppp authentication process.

```bash
leftist#debug ppp authentication
PPP authentication debugging is on
04:26:10: %LINK-3-UPDOWN: Interface Ser0/1, changed state to down
04:26:11: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial10/1, changed state to up
04:26:16: Se0/1 PPC: Treating connection as a dedicated line
04:26:16: %LINK-3-UPDOWN: Interface Ser0/1, changed state to up
04:26:16: Se0/1 PPC: Phase is AUTHENTICATING, by both
04:26:16: Se0/1 PAP: O AUTH-REQ id 32 len 21 from "prophet"
04:26:16: Se0/1 PAP: I AUTH-REQ id 32 len 21 from "prophet"
04:26:16: Se0/1 PAP: Authenticating peer prophet
04:26:16: Se0/1 IPCP: I CONFACK [REQsent] id 8 len 10
04:26:16: Se0/1 IPCP: Address 200.200.200.2 (0x0306C8C8C802)
04:26:16: Se0/1 IPCP: I CONFACK [ACKsent] id 8 len 4
04:26:16: Se0/1 IPCP: Address 200.200.200.1 (0x0306C8C8C801)
04:26:16: Se0/1 IP: Install route to 200.200.200.1
04:26:16: Se0/1 PPP: Phase is UP
04:26:16: Se0/1 IPCP: State is Open
04:26:16: Se0/1 IPCP: Addr 200.200.200.2 (0x0306C8C8C802)
04:26:16: Se0/1 IPCP: State is Open
04:26:16: Se0/1 CP: Install route to 200.200.200.1
04:26:16: Se0/1 PPP: Phase is AUTHENTICATING, by both
04:26:16: Se0/1 PAP: O AUTH-REQ id 32 len 21 from "prophet"
04:26:16: Se0/1 PAP: I AUTH-REQ id 32 len 21 from "prophet"
04:26:16: Se0/1 PAP: Authenticating peer prophet
04:26:16: Se0/1 IPCP: State is Open
04:26:16: Se0/1 CP: Install route to 200.200.200.1
04:26:16: Se0/1 PPP: Phase is UP
04:26:16: Se0/1 IPCP: State is Open
04:26:16: Se0/1 CP: Install route to 200.200.200.1
04:26:16: Se0/1 PPP: Phase is AUTHENTICATING, by both
04:26:16: Se0/1 PAP: O AUTH-REQ id 32 len 21 from "prophet"
04:26:16: Se0/1 PAP: I AUTH-REQ id 32 len 21 from "prophet"
04:26:16: Se0/1 PAP: Authenticating peer prophet
```
From the leftist router this time we see a request from “prophet” on s0/1 and then an authorization request (meaning “Ok I found you, I accept your username and password”). Then a couple of acknowledgements and acknowledgement of acknowledgements and the line comes up ready to communicate!

9. Let’s turn off debugging. Use “unde bug all” or “undebug ppp authentication.”
10. Let’s switch to CHAP. First, start by removing the PAP stuff:

```
leftist(config)#int s0/0
leftist(config-if)#ppp authentication chap
leftist(config-if)#exit
leftist(config)#no username prophet password legodoom
```

```
urvile(config)#int s0/1
urvile(config-if)#ppp authentication chap
urvile(config-if)#exit
urvile(config)#no username prophet password legodoom
```

One way we could do this is to use the hostnames of the routers and the enable passwords for easy access.

```
leftist(config)#int s0/0
leftist(config-if)#enc ppp
leftist(config-if)#ppp authentication chap
leftist(config-if)#exit
leftist(config)#username urvile password cisco
```

```
urvile(config)#int s0/0
urvile(config-if)#enc ppp
urvile(config-if)#ppp authentication chap
urvile(config-if)#exit
urvile(config)#username leftist password cisco
```

But, generally we want to have remote users whose names we can input for CHAP access to the router. This actually makes more sense and is more of a “real-world” scenario. Undo all of the last steps. This time use similar commands except the username and passwords are set a bit differently. The username must match the hostname of the destination router. Use the line between leftist and urvile to set up chap.

```
leftist(config)#int s0/0
leftist(config-if)#enc ppp
leftist(config-if)#ppp authentication chap
leftist(config-if)#exit
leftist(config)#username prophet password cisco
```
This will set up a username to “dial-in” and be “authenticated” to the urvile router. We chose to use the username prophet and are obligated to use the password cisco since we already set it up in our router basics. Next, on urvile, we use similar commands except that we set the username to the router which will be calling in. We must also include the hostname that will be calling in to urvile.

```
urvile(config)#int s0/0
urvile(config-if)#enc ppp
urvile(config-if)#ppp authentication chap
urvile(config-if)#ppp authentication chap callin
urvile(config-if)#ppp chap hostname prophet
urvile(config-if)#exit
urvile(config)#username leftist password cisco
```

Don’t forget to change the settings on both sides! (Use S0/1 on urvile.) Notice how we now have to use the hostname of the other router and the “enable secret” of “cisco” (the encrypted one). You will know when you have the right combination of user names and passwords when the line and protocol both come up.

11. Then view the CHAP with the same debugs...debug tasks, debug negotiation, and debug authentication. They should be similar to the PAP ones except that there is a three-way handshake and our passwords are encrypted. Can you see it?

**Challenge Lab or Supplemental Activities:**

1. Try switching the order of which router will be called into and which one will do the calling. Why would this be important? Why would you want to do this?
2. Try configuring PAP and CHAP on the same router. Why would you want to do this?
3. Can we do any authentication with HDLC? Try it and find out. When would you want to use PPP with authentication and when would you want to use HDLC?
4. What are the other debug options available with PPP? What does each of them do?
5. What options are available for PPP on a serial interface? (hint: ppp ?) For what is each used?
6. Use a protocol inspector to try “stealing” passwords over PAP and CHAP lines.
7. What the heck is a “magic number”? Go and find out.
8. What are those acronyms in our debug ppp negotiation? What do they mean? What is a IPCP and CDPCP?
9. When would you use Microsoft-chap?
10. Does our username/passwords set up under our interface have to match those put on our router?

**So what have I learned here?**

In this lab you have learned some of the options available with PPP authentication. You have seen the five steps in PPP negotiation. You should now be able to define and
differentiate between PAP and CHAP and when you would want to use each. You have seen that CHAP is better, from a security perspective, because the username and passwords are encrypted. This means they cannot as easily be “stolen” with a protocol inspector and used illegally. Do you remember what PAP and CHAP stand for? I would want to know if I was taking a test on it…hint, hint, wink, wink. In the next lab you will learn how to use another serial line encapsulation: frame relay.

Guest Router Name Derivation

Terminus, Leftist, and Urvile were three hackers from the Legion of Doom, who lived in Georgia, that were busted in 1990 by the U.S. Secret Service in connection with the Martin Luther King Day AT&T long distance network crash. They were known as “switching gurus” and as “heavy hitters” within the LoD because they frequently accessed BellSouth’s network. Apparently BellSouth, at that time, did not have very strict security in place.

Pro-Cote Inc.
Specialty Coatings Applicators
877-588-4300

Certified Conductive and Static Dissipative Floor applications for every need:
- Facilities with highly sensitive electronic equipment
- Computer facilities
- Electronic manufacturing & assembly
- Facilities containing AGV lines
- Clean rooms
- Manufacturing/processing flammable or combustible materials
- Munitions plants

Pro-Cote Inc. is a proud installer of Valspar Flooring Systems
www.pro-cote-usa.com

Pro-Cote Inc.
P. O. Box 1766
Largo, FL  33779
Toll Free (877) 588-4300
Remote Access DUN with PPP Encapsulation

Objective:
To learn how to set up a dial-up networking connection into a serial port to allow the use of the PPP protocol.

Tools and Materials:
(5) Workstations
(2) routers (with slow speed serial wic’s only…2AS, not 2T)
(4) modems
(3) DB-9 to RJ-45 adapters (PC Com1)
(4) RS-232 to RJ-45 adapters
(4) RJ-11 cables
(4) Straight-through cables
(1) cross-over cable
(2) console cables
(file servers and switches will not be used…you will need the IPX numbers for configuring the routers. We will use loopbacks to emulate the networks.)

Lab Diagram:
Step-By-Step Instructions:
The key here is to break the lab down into “baby steps.” Forget about the IPX stuff completely… save it for last. Our plan of attack will be to set up our internal network and test it. Then configure the dial up networking and test it. And then finish it off with IPX.

1. Set up the lab and cable it as shown. Check it twice!
2. Set up the basics on each router.
3. Configure the interfaces and loopbacks.
4. Pick a routing protocol and advertise the networks.
5. Test your connectivity by using ping between loopbacks on the routers. Use trace route and sh ip route also.
6. Configure the dial-up networking on the workstations if they already have not been done.
7. Configure the serial interface on “dark” to accept dial-up networking. Oh? You say you haven’t done that before? Sure you have… sort of. Use the same commands you used to set up the AUX port. The only difference between the two is now we can use PPP as an encapsulation type (with usernames and passwords if we want---not shown below). We could not easily do that with an AUX port DUN. You can add a banner or MOTD if you wish.

```
dark(config)#int s0/0               (or line 65)
dark(config-if)#login
dark(config-if)#password dark
dark(config-if)#speed 115200
dark(config-if)#flowcontrol hardware
dark(config-if)#stopbits 1
dark(config-if)#transport input all
dark(config-if)#modem inout
dark(config-if)#modem autoconfigure discovery
dark(config-if)#enc ppp
```

8. Test your dial up networking from each workstation into the network. When dialed in each workstation should be able to ping all of the connections including the loopbacks.
9. Add in the IPX stuff. We only put the file servers in the picture because that information needs to be included in the router programs. Ok…I will make it easy for you:

```
dark(config)#ipx routing 0000.AAAA.0001
dark(config-router)#exit
dark(config)#int loop 0
dark(config-if)#ipx network 100 encapsulation Novell-Ether
dark(config-if)#int s0/1
dark(config-if)#ipx network 192
dark(config-if)#ipx sap-interval 0

lord(config)#ipx routing 0000.BBBB.0002
lord(config-router)#exit
lord(config)#int loop 0
lord(config-if)#ipx network 200 encapsulation Novell-Ether
```
Then you can decide if you want to do your ipx routing statically (with static routes) or dynamically (using router ipx with advertised networks).

**Statically:**
```
dark(config)#ipx route 200 192.0000.BBBB.0002
dark(config)#ipx route 2000 192.0000.BBBB.0002
dark(config)#ipx sap 4 2000.0000.0000.0002 451 2
dark(config)#ipx router rip
dark(config-router)#no network 192
```
```
lord(config)#ipx route 100 192.0000.AAAA.0001
lord(config)#ipx route 1000 192.0000.AAAA.0001
lord(config)#ipx sap 4 1000.0000.0000.0001 451 2
lord(config)#ipx router rip
lord(config-router)#no network 192
```

**Or dynamically:**
```
dark(config)#router rip
dark(config-router)#version 2
dark(config-router)#network 192.168.1.0
dark(config-router)#network 1.0.0.0
```
```
lord(config)#router rip
lord(config-router)#version 2
lord(config-router)#network 192.168.1.0
lord(config-router)#network 2.0.0.0
```

**Challenge Lab or Supplemental Activities:**
1. Try changing IPX numbers.
2. Try changing to different IPX frame types (ie., from 802.3 to 802.2 or SNAP, etc).
3. Try the lab once with static IPX routing and then with the dynamic IPX routing. Which one do you prefer?

**So what have I learned here?**
This is actually a mini-whole enchilada/crazy insane lab for the remote access part. Eh, what the heck we even through in some IPX for good measure. The biggest reason why this is here is to get you to start thinking about breaking down networks into “baby steps” when configuring them. They do not look as intimidating then. You will find those people who have problems setting up their networks are also the same people who “skip” steps or “lump several things together” in order to save time. Hmpf! Take your time because you are getting paid by the hour anyway.
Jason Allen Diekman, a.k.a. “Shadow Knight” or “Dark Lord,” was charged with hacking into NASA, Oregon State University, and a San Francisco area ISP in 2002. He was sentenced to 21 months in Federal Prison, ordered to pay restitution of $87,736.29, and will have 3 years of probation, which includes no computer accessing. Apparently he used stolen credit card numbers to transfer money through Western Union and to try buying equipment from NASA’s Jet Propulsion Laboratory. While free on bail from charges the “defendant” (use whatever word you want there) hacked into several other university computer systems. Boy this one is a case study in stupidity 101. Even “geniuses” do not always have “common sense.” Won’t shower time be fun for him too?
Setting up a Router to be a Frame Relay Switch

Objective:
In this lab you learn how to “change” a router into a frame relay switch.

Tools and Materials:
(1) router
(1) workstation
(1) console cable

Lab Diagram:

Background:
Many people do not have the luxury of having an Adtran Atlas 550 for frame relay simulation. This lab will show you how you can transform a router into a frame relay switch. You can then use this for some of the basic frame relay experiments that only require a frame relay switch between two routers. If you have a 4000 series router then you can make a frame switch with 3 or more fully-meshed frame lines. What? You only have 2500’s or 2600’s? Oh well, you can only set up the router as a static frame relay switch between two routers.

Step-By-Step Instructions:
1. Set up the basics on the router.

   router>en
   router#config t
   router(config)#hostname Frswitch
   Frswitch(config)#en secret cisco
   Frswitch(config)#en password class
   Frswitch(config)#line con 0
   Frswitch(config-line)#exec-timeout 0 0
   Frswitch(config-line)#logging synchronous
   Frswitch(config-line)#line vty 0 4
   Frswitch(config-line)#login
   Frswitch(config-line)#password cisco

2. Enable the router to become a frame relay switch:

   Frswitch(config)#frame-relay switching

3. Set up the interfaces on the router.

   Frswitch(config)#int s0/0
   Frswitch(config-if)#enc frame-relay
4. Then configure a dlc-switching table on EACH interface.

```plaintext
Frswitch(config)#int s0/0
Frswitch(config-if)#frame-relay route 18 interface s0/1 16
Frswitch(config)#int s0/1
Frswitch(config-if)#frame-relay route 16 interface s0/0 18
```

Think of this like: “From, Through, and To”

5. You are now ready to start using your frame-relay switch. Don’t forget to save the configuration.

**Supplemental Lab or Challenge Activities:**

1. Go out and research how many serial lines can be put into a 2610, 2611, 2620, or 2620 router. Why are we limited to just 2 serial lines or can we have more? Surely we might need more than two routers hooked together with frame relay. For example:

   ![Network Diagram]

   - Seattle (s0/3, dlc#19)
   - Detroit (s0/2, dlc#17)
   - Los Angeles (s0/0, dlc#16)
   - Tampa (s0/1, dlc#18)

**So what have I learned here?**

In this lab you learned how to turn a router into a frame relay switch. Would you do this inside a company? Almost always no. Your serial lines with HDLC can provide clocking to move the information. In the next couple of labs you will learn about configuring different topologies with frame relay networking.
Objective:
To learn how to set up a basic frame relay network with 2 routers.

Tools and Materials:
(3) routers
(2) DCE-to-DTE cables
(1) console cable
(1) workstation

Lab Diagram:

<table>
<thead>
<tr>
<th>Router</th>
<th>Make</th>
<th>Make</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0/0</td>
<td>192.168.1.2/24</td>
<td>192.168.1.1/24</td>
</tr>
<tr>
<td>Dce/dte</td>
<td>dte</td>
<td>dte</td>
</tr>
<tr>
<td>Dlci</td>
<td>16</td>
<td>18</td>
</tr>
<tr>
<td>Loop 0</td>
<td>2.2.2.2/8</td>
<td>1.1.1.1/8</td>
</tr>
</tbody>
</table>

Step-By-Step Instructions:
1. Set up a frame relay switch with two routes (see last lab).
2. Set up the basics on one router.

```
router#config t
router(config)#hostname turner
turner(config)#enable secret cisco
turner(config)#enable password class
turner(config)#line con 0
turner(config-line)#logging synchronous
turner(config-line)#exec-timeout 0 0
turner(config-line)#line vty 0 4
turner(config-line)#password cisco
turner(config-line)#login
```
Add interface configurations.

```plaintext
turner(config)# int s0/0
turner(config-if)# ip address 192.168.1.1 255.255.255.0
turner(config-if)# enc frame-relay
turner(config-if)# no shut
turner(config)# int loop 0
turner(config-if)# ip address 1.1.1.1 255.0.0.0
turner(config-if)# no shut
```

Add routing protocol.

```plaintext
turner(config)# router eigrp 38
turner(config-router)# network 192.168.1.0
turner(config-router)# network 1.0.0.0
```

3. Now, do the same for the other router.
4. Verify connectivity. When you check your routes, ping, and view the frame relay circuit status you should see:

```plaintext
turner# sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
       default
       U - per-user static route, o - ODR

Gateway of last resort is not set

C   1.0.0.0/8 is directly connected, Loopback0
D   2.0.0.0/8 [90/2297856] via 192.168.1.2, 00:11:14, Serial0/1
C   192.168.1.0/24 is directly connected, Serial0/1
```

```plaintext
turner# ping 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/65/68 ms
```

You can check the status of the frame relay circuit connection with “show frame-relay pvc.”

turner#sh frame-relay pvc

PVC Statistics for interface Serial0/1 (Frame Relay DTE)

DLCI = 16, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/1

  input pkts 40 output pkts 38 in bytes 3302
  out bytes 3130 dropped pkts 0 in FECN pkts 0
  in BECN pkts 0 out FECN pkts 0 out
  BECN pkts 0 in DE pkts 0 out DE pkts 0
  out bcast pkts 15 out bcast bytes 926
  pvc create time 00:12:39, last time pvc status changed 00:12:19

Challenge Lab or Supplemental Activities:
1. Switch address schemes to a pure Class “B” network.
2. Switch address schemes to a pure Class “A” network.

So what have I learned here?
In this lab you have learned how to set up the bare minimum requirements for a Frame relay main connection using a router set up as a frame relay switch. In the next couple of labs you will learn some more intermediate-level commands for setting up multiple router frame relay networks.

Guest Router Name Derivation

Patrice Williams was sent to prison in 2002 after she, and a partner (Makeebra Turner), hacked into the Chase Financial Corporation. Apparently this dastardly duo stole credit card numbers and used them to purchase about $600,000 worth of merchandise on 68 different accounts. They also “distributed” some of those numbers to someone else in Georgia who, in turn, purchased about $100,000. The brain trust plea-bargained down to a one-year and a day prison term in return.
Frame Relay: Hub and Spoke with 3 routers

Objective:
To be able to configure a “hub and spoke” frame relay network using 3 routers.

Background:
You can configure frame relay as a “hub and spoke” topology. Essentially one router acts as a “master” or “primary” route controller (the “hub”). All others act as “slaves” or “secondary” routes (the “spokes”) with configurations leading to the “master” or “primary” controller. If this were to be a “fully-meshed” frame relay network then each would have routes to all others. In our example below see how router “Lloyd” and “Allen” map back to “Timothy” while Timothy routes to both Lloyd and Allen. We use hub and spokes for control over the network and, sometimes, to reduce costs.

Materials Needed:
(3) routers
(3) DTE cables
(1) Adtran atlas 550
(1) PC/workstation
(1) console cable

Lab Diagram:

Router | Timothy | Allen | Lloyd
--- | --- | --- | ---
S0/0 | 192.168.20.1/24 | 192.168.20.2/24 | 192.168.20.3/24
Loop 0 | 192.168.1.1/24 | 192.168.2.1/24 | 192.168.3.1/24
Adtran | 1/1 | 1/2 | 2/1
Dlci | 16 | 18 | 17
Master? | Yes | No | No
Maps | 192.168.20.3-dlci 17 | 192.168.20.1-dlci 16 | 192.168.20.1-dlci16

192.168.20.2-dlci 18
Step-By-Step Instructions:
1. Cable the lab as shown and set up the basics on each router. Choose a routing protocol and set it up on each router (don’t forget to advertise your networks). Add the loopback interface configurations.
2. To set up the “master” or “primary” router as a “hub:”

```plaintext
timothy(config)#int s0/0
timothy(config-if)#ip address 192.168.20.1 255.255.255.0
timothy(config-if)#enc frame-relay
 timothy(config-if)#frame-relay map ip 192.168.20.2 18
 broadcast
 timothy(config-if)#frame-relay map ip 192.168.20.3 17
 broadcast
 timothy(config-if)#frame-relay lmi-type ansi
 timothy(config-if)#no shut
```

Basically you are setting the ip, changing the encapsulation type to frame-relay and then making maps to the other routers and broadcasting the maps (with ip’s and dlci numbers). The Adtran’s have been configured for lmi-type ansi.

3. To set up the “‘slaves” or “secondary” routers:

```plaintext
allen(config)#int s0/0
allen(config-if)#ip address 192.168.20.2 255.255.255.0
allen(config-if)#enc frame-relay
allen(config-if)#frame-relay map ip 192.168.20.1 16
 broadcast
allen(config-if)#frame-relay lmi-type ansi
allen(config-if)#no shut

lloyd(config)#int s0/0
lloyd(config-if)#ip address 192.168.20.3 255.255.255.0
lloyd(config-if)#enc frame-relay
lloyd(config-if)#frame-relay map ip 192.168.20.1 16
 broadcast
lloyd(config-if)#frame-relay lmi-type ansi
lloyd(config-if)#no shut
```

4. Test your configuration using “sh frame pvc,” “ping,” and “sh ip route.” You should see:

```plaintext
timothy#sh frame pvc
PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 17, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE,
INTERFACE = Serial0/0

    input pkts 42 output pkts 45 in bytes 3464
    out bytes 3676 dropped pkts 1 in FECN pkts 0
    in BECN pkts 0 out FCEN pkts 0 out BECN pkts 0
    in DE pkts 0 out DE pkts 0
    out bcast pkts 18 out bcast bytes 1152
pvc create time 00:23:24, last time pvc status changed
00:16:49
```

655
**DLCI = 18, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0**

input pkts 54  output pkts 53  in bytes 4472  
out bytes 4364  dropped pkts 0  in FECN pkts 0  
in BECN pkts 0  out FECN pkts 0  out BECN pkts 0  
in DE pkts 0  out DE pkts 0  
out bcast pkts 21  out bcast bytes 1344  
pvc create time 00:23:18, last time pvc status changed 00:19:50

Notice you are on “timothy” which uses dcli #16 to connect to the Adtran. When you do a sh frame pvc you see the status of dcli #17 and #18…the other two dcli’s.

**Challenge Lab or Supplemental Activities:**

1. Change the map statements to reflect a “full-mesh” topology. Note any differences in pvc’s, ip routes, etc.
2. Why do we need to use the same subnet over all three frame relay interfaces? I thought we needed separate subnet numbers on each?
3. Put an error on the serial interface on the hub router (timothy—shut down the interface, remove the lmi type, etc). See if you can still get connectivity between all three routers. What about connectivity between allen and Lloyd?

**So what have I learned here?**

So far we have learned that frame relay is just another encapsulation that we can use on a serial interface (which is HDLC by default). In this lab you have learned to set up a “hub and spoke” frame relay network. Each router is connected with a circuit to a master router that contains maps to all others. We do this to save money because each circuit connection costs money. Obviously if we can purchase two frame relay circuits instead of three then we would be saving money. In the next lab you will learn how to configure a full-mesh frame relay network using subinterfaces.

**Guest Router Name Derivation**

Timothy Lloyd Allen was a chief network program designer for Omega Engineering Corp (New Jersey) who was sentenced to 41 months in prison for unleashing a $10 million “time bomb” within a manufacturing software program he helped design. After 11 years with the company he was “suddenly laid off,” but, ha-ha, he would “get his revenge.” And boy did he. Now he’s got to hope he finds a bigger boy friend than everyone else. Won’t shower time be fun?
Fully-Meshed Frame Relay with 3 Routers and Sub-interfaces

Objective:
To be able to configure a “fully-meshed” frame relay network using 3 routers and sub-interfaces.

Background:
After learning how to configure “hub and spoke” networks it is now time to configure a fully-meshed frame relay network. Unlike the hub and spoke network which had all serial interfaces on one subnet we will have to use a different subnet for every connection in our meshed network. This will require two or more ip addresses on every serial interface used. Since we cannot use more than one ip address on an interface we will be using sub-interfaces with different ip addresses. To geek it up the sub-interfaces are “logical” sub-interfaces on our “physical” main interface. You will also begin to see why we identified our DLCI’s in the manner we have been using.

Materials Needed:
(3) routers
(3) DTE cables
(1) Adtran atlas 550
(1) PC/workstation
(1) console cable

Lab Diagram:

<table>
<thead>
<tr>
<th>Router</th>
<th>Diekman</th>
<th>Shadow</th>
<th>Knight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial int.</td>
<td>in text</td>
<td>in text</td>
<td>in text</td>
</tr>
<tr>
<td>Loop 0</td>
<td>1.1.1.1/8</td>
<td>2.2.2.2/8</td>
<td>3.3.3.3/8</td>
</tr>
<tr>
<td>Adtran</td>
<td>1/1</td>
<td>1/2</td>
<td>2/1</td>
</tr>
<tr>
<td>Dlci</td>
<td>16</td>
<td>18</td>
<td>17</td>
</tr>
</tbody>
</table>
Step-By-Step Instructions:

1. Cable the lab as shown and set up the basics on each router. Choose a routing protocol and set it up on each router (don’t forget to advertise your networks). Add the loopback interface configurations.

2. To set up the routers for subinterfaces. The important thing here is to understand which IP addresses and which DLCI numbers to use. Let’s look at the three PVC’s we will be creating here first:

The design key is to remember the DLCI connection number. If you are configuring “knight” router (almost funny, huh?) which has a connection to DLCI 17 then you will need to set up connection with DLCI 16 and 18. The drawings used by CISCO are difficult to understand unless you have developed your own step-by-step method (what a coincidence! That is what you are getting here!).

First we need to pick out some subnet numbers for our three circuits. Let’s use these:

- circuit #1  192.168.1.0 network (use .1 and .2)
- circuit #2  192.168.2.0 network (use .1 and .2)
- circuit #3  192.168.3.0 network (use .1 and .2)

Next we need to associate them with the DLCI numbers. Usually you will see them given in a picture as shown on the top of the next page. You can see why these things can be confusing. The DLCI’s really help. It is easier to
understand if you add the dlci connection (in this case to our Adtran) within a table format. Start by putting “none” in the sub-interface configuration space (notice the format s0/0.16 for dlci #16) for the dlci to which the serial line connects (Use the first drawing not the one above…it can be confusing):

<table>
<thead>
<tr>
<th>Router</th>
<th>Diekman</th>
<th>Shadow</th>
<th>Knight</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0/0</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Adtran</td>
<td>1/1</td>
<td>1/2</td>
<td>2/1</td>
</tr>
<tr>
<td>Dlci</td>
<td>16</td>
<td>18</td>
<td>17</td>
</tr>
<tr>
<td>S0/0.16</td>
<td>none</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S0/0.17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S0/0.18</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Next, add in the ip address for the subnet/sub-interface… Let’s start with circuit #1 (192.168.1.0 network using .1 and .2):

<table>
<thead>
<tr>
<th>Router</th>
<th>Diekman</th>
<th>Shadow</th>
<th>Knight</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0/0</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Adtran</td>
<td>1/1</td>
<td>1/2</td>
<td>2/1</td>
</tr>
<tr>
<td>Dlci</td>
<td>16</td>
<td>18</td>
<td>17</td>
</tr>
<tr>
<td>S0/0.16</td>
<td>none</td>
<td></td>
<td>192.168.1.2</td>
</tr>
<tr>
<td>S0/0.17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S0/0.18</td>
<td>192.168.1.1</td>
<td>none</td>
<td></td>
</tr>
</tbody>
</table>

See? Our circuit #1 has a connection on dlci #18 on Diekman and dlci #16 on Shadow. Since we are using the 192.168.1.0 network we arbitrarily pick which one has which address. We will be using the sub-interface number that corresponds with the dlci number. We don’t have too it is just easier that way.
Next, let’s fill in the information for circuit #2 (192.168.2.0 network using .1 and .2):

<table>
<thead>
<tr>
<th>Router</th>
<th>Diekman</th>
<th>Shadow</th>
<th>Knight</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0/0</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Adtran</td>
<td>1/1</td>
<td>1/2</td>
<td>2/1</td>
</tr>
<tr>
<td>Dlci</td>
<td>16</td>
<td>18</td>
<td>17</td>
</tr>
<tr>
<td>S0/0.16</td>
<td>none</td>
<td>192.168.1.2</td>
<td></td>
</tr>
<tr>
<td>S0/0.17</td>
<td>192.168.2.1</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td>S0/0.18</td>
<td>192.168.1.1</td>
<td>none</td>
<td>192.168.2.2</td>
</tr>
</tbody>
</table>

Finally, let’s fill in the information for circuit #3 (192.168.3.0 network using .1 and .2):

<table>
<thead>
<tr>
<th>Router</th>
<th>Diekman</th>
<th>Shadow</th>
<th>Knight</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0/0</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Adtran</td>
<td>1/1</td>
<td>1/2</td>
<td>2/1</td>
</tr>
<tr>
<td>Dlci</td>
<td>16</td>
<td>18</td>
<td>17</td>
</tr>
<tr>
<td>S0/0.16</td>
<td>none</td>
<td>192.168.1.2</td>
<td>192.168.3.2</td>
</tr>
<tr>
<td>S0/0.17</td>
<td>192.168.2.1</td>
<td>192.168.3.1</td>
<td>none</td>
</tr>
<tr>
<td>S0/0.18</td>
<td>192.168.1.1</td>
<td>none</td>
<td>192.168.2.2</td>
</tr>
</tbody>
</table>

Eh, voila! We are ready to configure our routers. Let’s configure our serial interface and sub-interfaces on “knight” (dlci #17):

```
knight(config)#int s0/0
knight(config-if)#enc frame-relay
knight(config-if)#frame-relay lmi-type ansi
knight(config-if)#no shut

knight(config-if)#int s0/0.16 point-to-point
knight(config-subif)#ip address 192.168.3.2 255.255.255.0
knight(config-subif)#no shut
knight(config-subif)#frame-relay interface-dlci 16

knight(config-if)#int s0/0.18 point-to-point
knight(config-subif)#ip address 192.168.2.2 255.255.255.0
knight(config-subif)#no shut
knight(config-subif)#frame-relay interface-dlci 18
```

do not forget to use your “cut and paste” utilities…these are a time-saver! The last line (frame-relay interface-dlci 18) just identifies the dlci connection (in this case to our Adtran). Now isn’t that nice that we already have it in our drawing. Notice how there is no ip address on S0/0. A good way to double-check yourself: Knight connects using dlci #17 so we should have sub-interface connections for #16 and #18. You can never be too careful during configuration. So now we can configure the next router: Shadow (dlci #18).
shadow(config)#int s0/0
shadow(config-if)#enc frame-relay
shadow(config-if)#frame-relay lmi-type ansi
shadow(config-if)#no shut

shadow(config-if)#int s0/0.16 point-to-point
shadow(config-subif)#ip address 192.168.1.2 255.255.255.0
shadow(config-subif)#no shut
shadow(config-subif)#frame-relay interface-dlci 16

shadow(config-if)#int s0/0.17 point-to-point
shadow(config-subif)#ip address 192.168.2.1 255.255.255.0
shadow(config-subif)#no shut
shadow(config-subif)#frame-relay interface-dlci 17

So now we can configure the next router: Diekman (dlci# 16)

diekman(config)#int s0/0
diekman(config-if)#enc frame-relay
diekman(config-if)#frame-relay lmi-type ansi
diekman(config-if)#no shut
diekman(config-if)#int s0/0.17 point-to-point
diekman(config-subif)#ip address 192.168.3.1 255.255.255.0
diekman(config-subif)#no shut
diekman(config-subif)#frame-relay interface-dlci 17
diekman(config-if)#int s0/0.18 point-to-point
diekman(config-subif)#ip address 192.168.1.1 255.255.255.0
diekman(config-subif)#no shut
diekman(config-subif)#frame-relay interface-dlci 18

3. Test your configuration using “sh frame pvc,” “ping,” and “sh ip route.”
   You should see:

diekman#sh frame pvc
PVC Statistics for interface Serial0/0 (Frame Relay DTE)

**DLCI = 17, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE,**
INTERFACE = Serial0/0
  input pkts 42  output pkts 45  in bytes 3464
  out bytes 3676  dropped pkts 1  in FECN pkts 0
  in BECN pkts 0  out FECN pkts 0  out BECN pkts 0
  in DE pkts 0  out DE pkts 0
  out bcast pkts 18  out bcast bytes 1152
  pvc create time 00:23:24, last time pvc status changed 00:16:49

**DLCI = 18, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE,**
INTERFACE = Serial0/0
  input pkts 54  output pkts 53  in bytes 4472
  out bytes 4364  dropped pkts 0  in FECN pkts 0
  in BECN pkts 0  out FECN pkts 0  out BECN pkts 0
  in DE pkts 0  out DE pkts 0
  out bcast pkts 21  out bcast bytes 1344
  pvc create time 00:23:18, last time pvc status change
Notice you are on “diekman” which uses dlci #16 to connect to the Adtran. When you do a sh frame pvc you see the status of dcli #17 and #18…the other two dlci’s.

**Challenge Lab or Supplemental Activities:**

1. Why did we use “point-to-point?” Our other option when configuring our sub-interface was “multi-point.” Find out when we use each.
2. Put an error on the serial interface on one of the routers (shut down the interface, remove the lmi type, etc). See if you can still get connectivity between all three routers.

**So what have I learned here?**

So far we have learned that frame relay is just another encapsulation that we can use on a serial interface (which is HDLC by default). You have also learned how to set up a “hub and spoke” frame relay network which only has partial meshed connectivity. We did that to save money. In this lab you learned how to configure a fully meshed frame relay network. You learned that you need multiple ip addresses on your physical serial interface (which you cannot do) so you used logical sub-interfaces to set up your circuits. My, my, my so much to do! Next we will start expanding upon your knowledge of hub and spoke networks and fully-meshed networks by adding in some networking “twists.”

---

**Guest Router Name Derivation**

Jason Allen Diekman, a.k.a. “Shadow Knight” or “Dark Lord,” was charged with hacking into NASA, Oregon State University, and a San Francisco area ISP in 2002. He was sentenced to 21 months in Federal Prison, ordered to pay restitution of $87,736.29, and will have 3 years of probation, which includes no computer accessing. Apparently he used stolen credit card numbers to transfer money through Western Union and to try buying equipment from NASA’s Jet Propulsion Laboratory. While free on bail from charges the “defendant” (use whatever word you want there) hacked into several other university computer systems. Boy this one is a case study in stupidity 101. Even “geniuses” do not always have “common sense.” Won’t shower time be fun for him too?

---

**Take college courses on a new schedule--yours.**

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education. Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

[http://e.spcollege.edu](http://e.spcollege.edu)

St. Petersburg College's eCampus. Your time. Your place. Your future.
Frame Relay Operation and Troubleshooting

Objective:
To learn how to troubleshoot frame relay problems.

Theory of operation:
Frame relay is a layer 2 technology. Troubleshooting it is simple. It is when frame is combined with other stuff that it becomes complicated. In a simple, basic frame relay connection you only need to configure:

1. the ip addresses on the same subnet with the proper mask
2. set the encapsulation type to frame relay
3. bring the interface up
4. set the lmi-type, if needed.

That is it. To troubleshoot we can use our OSI layer-by-layer technique:

Snapshot of activity: show frame-relay pvc

```plaintext
show controller s0/0       layer 1
show interface s0/0        layer 1
show frame-relay pvc       layer 2
show frame-relay lmi       layer 2
debug frame-relay events   layer 2
show frame-relay map       layer 2/3
show ip route              layer 3
```

Step-By-Step Instructions:
1. Frame relay is one of the easier problems to troubleshoot because there really is not much that can go wrong with frame relay: it either works or it doesn’t. Let’s start by viewing our available frame relay show and debug commands. I have high-lighted some of the more commonly-used commands:

```plaintext
router#sh frame ?
end-to-end      Frame-relay end-to-end VC information
fragment        show frame relay fragmentation info.
ip             show frame relay IP statistics
lapf       show frame relay lapf status/statistics
lmi               show frame relay lmi statistics
map           Frame-Relay map table
pvc            show frame relay pvc statistics
qos-autosense     show frame relay qos-autosense info.
route           show frame relay route
svc             show frame relay SVC stuff
traffic       Frame-Relay protocol statistics
vofr       Show frame-relay VoFR statistics
```
router#debug frame-relay ?

detailed          Detailed Debug: Only for Lab use
dlsw              Frame Relay dlsw
end-to-end        Frame-relay end-to-end VC info.
events            Important Frame Relay packet events
foresight         Frame Relay routr ForeSight support
fragment          Frame Relay fragment
hpr               Frame Relay APPN HPR
ip                Frame Relay Internet Protocol
l3cc              Frame Relay Layer 3 Call Control
l3ie              Frame Relay IE parsing/construction
lapf              Frame Relay SVC Layer 2
llc2              Frame Relay llc2
lmi               LMI packet exch w/ service provider
nli               Network Layer interface
packet            Frame Relay packets
ppp               PPP over Frame Relay
rsrb              Frame Relay rsrb
verbose           Frame Relay

2. Next let’s use some of those more common commands to see “good” traffic, packets and statistics on a frame relay connection between two routers (one dlci #16 the other dlci#18). As always we like to start with an overall “snapshot” of our connection. We use show frame pvc to show our permanent virtual circuit statistics (layer 2):

router#sh frame pvc

PVC Statistics for interface Serial0/1 (Frame Relay DTE)

DLCI = 18, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE,
INTERFACE = Serial0/1

input pkts 18        output pkts 23     in bytes 1758
out bytes 2114       dropped pkts 0     in FECN pkts 0
in BECN pkts 0       out FECN pkts 0   out BECN pkts 0
in DE pkts 0         out DE pkts 0
out bcast pkts 1     out bcast bytes 30
pvc create time 00:05:19, last time pvc status changed 00:00:30

Which dlci is this router connected to? If you said 18 then you were incorrect. The frame status we see is for the other one. If we have more than one dlci in our network we will see all but our own dlci number. For example, if we were connected to dlci #16 and our other routers were connected to dlci#17, 18, and 19, then a show frame pvc command would show us the statistics for dlci #17, 18, and 19. Now let’s look at our LMI statistics. This does not show us much except our LMI type is CISCO. Obviously I didn’t use an ADTRAN because the LMI type is set to ANSI on those.
We can see our frame relay map. Since we only are using two routers we should only see one map statement:

```
router#sh frame map
Serial0/1 (up): ip 192.168.1.2 dlci 18(0x12,0x420), dynamic, broadcast,, status defined, active
```

3. Some of the more common problems you will encounter in a basic frame relay connection will be:

- Wrong type of serial cable (dce/dte) layer 1
- No serial connection layer 1/2
- Incorrect serial line encapsulation layer 2
- Wrong ip address/mask layer 3
- No routing protocol (dynamic or static) layer 3

Let’s take a few pages to look at what will happen to your frame relay connection and what your troubleshooting commands will show you.

4. Wrong type of serial cable (dce/dte). Let’s start with an overview of our frame relay connection:

```
make#sh controller s0/0
Interface Serial0/0
Hardware is PowerQUICC MPC860
DCE V.35, no clock
idb at 0x80F3DD50, driver data structure at 0x80F43264
```
Whammo! Nailed that one quicker than a new bucket of chicken at an all you can eat buffet! The show controllers command tells us we have the dce with no clock, which is wrong. We must use dte.

5. No serial connection. This one is just as easy.

    router#sh frame pvc

    PVC Statistics for interface Serial0/0 (Frame Relay DTE)

    router#sh controller s0/0
    Interface Serial0/0
    Hardware is PowerQUICC MPC860
    No serial cable attached

6. Incorrect serial line encapsulation

    router#sh frame pvc

    router#sh controller s0/0
    Interface Serial0/0
    Hardware is PowerQUICC MPC860
    DTE V.35 TX and RX clocks detected.

    make#sh int s0/0
    Serial0/0 is up, line protocol is up
    Hardware is PowerQUICC Serial
    Internet address is 192.168.1.2/24
    MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely
    255/255, load 1/255
    Encapsulation HDLC, loopback not set, keepalive set
    (10 sec)

    Bingo! We go back and change our encapsulation type to frame relay and it works.

7. Wrong ip address/mask  Here I just changed the network number on one side. We start with our show frame pvc and work through the commands:

    router#sh frame pvc

    PVC Statistics for interface Serial0/0 (Frame Relay DTE)

    DLCI = 16, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE,
    INTERFACE = Serial0/0

    input pkts 3 output pkts 6 in bytes 158
    out bytes 334 dropped pkts 0 in FECN pkts 0
    in BECN pkts 0 out FECN pkts 0 out BECN pkts 0
    in DE pkts 0 out DE pkts 0
    out bcast pkts 3 out bcast bytes 124

    pvc create time 00:01:53, last time pvc status changed
    00:01:53
A hint! We can check our ip address against the ip address on the other end of our frame-relay line with show frame map:

```sh
router# sh frame map
Serial0/0 (up): ip 192.168.1.1 dlc 16 (0x10,0x400), dynamic, broadcast,, status defined, active
```

We can see it in our frame relay map and our show interface s0/0. So we fix it (back to 192.168.1.2/24) and it works.

8. No routing protocol (dynamic or static).

```
router# sh frame pvc
```

```
PVC Statistics for interface Serial0/0 (Frame Relay DTE)
DLCI = 16, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE,
INTERFACE = Serial0/0

  input pkts 1    output pkts 6           in bytes 30
  out bytes 550   dropped pkts 0         in FECN pkts 0
  in BECN pkts 0  out FECN pkts 0       out BECN pkts 0
  in DE pkts 0    out DE pkts 0
  out bcast pkts 1 out bcast bytes 30

  pvc create time 00:00:23, last time pvc status changed 00:00:23
```

```
router# sh controllers s0/0
Interface Serial10/0
Hardware is PowerQUICC MPC860
DTE V.35 TX and RX clocks detected.
```
router#sh int s0/0
Serial0/0 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 192.168.1.2/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255
  Encapsulation FRAME-RELAY, loopback not set, keepalive set (10 sec)
  LMI enq sent  45, LMI stat recv'd 47, LMI upd recv'd 0, DTE LMI up
  LMI enq recv'd 0, LMI stat sent  0, LMI upd sent  0
  LMI DLCI 1023 LMI type is CISCO frame relay DTE

make#sh frame lmi

LMI Statistics for interface Serial0/0 (Frame Relay DTE)
LMI TYPE = CISCO
  Invalid Unnumbered info 0   Invalid Prot Disc 0
  Invalid dummy Call Ref  0   Invalid Msg Type 0
  Invalid Status Message 0    Invalid Lock Shift 0
  Invalid Information ID 0    Invalid Report IE Len 0
  Invalid Report Request 0    Invalid Keep IE Len 0
  Num Status Enq. Sent 52     Num Status msgs Rcv'd 54
  Num Update Status Rcv'd 0    Num Status Timeouts 0

router#sh frame map
Serial0/0 (up): ip 192.168.1.1 d1ci 16(0x10,0x400),
  dynamic, broadcast,, status defined, active

router#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP,EX - EIGRP external,0 - OSPF,IA - OSPF inter area
N1-OSPF NSSA external type 1,N2 - OSPF NSSA external type 2
E1-OSPF external type 1, E2 - OSPF external type 2, E - EGP
i-IS-IS,LI-IS-IS lev -1,L2-IS-IS lev -2,*-candidate default
U - per-user static route, o - ODR

Gateway of last resort is not set

C    2.0.0.0/8 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, Serial0/0

Ok…we can take a hint here…with our show frame map we see we are expecting dynamic routing to take place. But we don’t see any routes learned via dynamic routing in our sh ip route. So let’s check our router config. We find no routing protocol. Therefore we add in the same routing protocol that is enabled on the other side:

router(config)#router eigrp 38
router(config-router)#network 192.168.1.0
router(config-router)#network 1.0.0.0
So what have I learned here?
Nothing really glamorous just frame relay operation and troubleshooting. Again it is easy when you know how. Unlike the other troubleshooting labs this one relies more upon over-all troubleshooting commands (like show controllers and sh ip route) than upon specific frame-relay only troubleshooting commands. Keep in mind that your problem may not be frame-relay related at all…it could be host names, ip addresses, etc. If you really, really get stuck, then save your configs, turn everything off and go do something for an hour or so. A clear head can really help in troubleshooting.

Looking for college courses?
We deliver.

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education.

Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

St. Petersburg College’s eCampus
http://e.spcollege.edu

Your time. Your place. Your future.
Basic ISDN Configuration with BRI interface (MERGE)

Objective:
To learn how to set up a basic ISDN connection, using BRI interfaces, between two routers. In this lab you will be using a MERGE box for ISDN emulation.

Tools and Materials:
(2) routers  
(2) straight-through cables  
(1) workstation  
(1) MERGE ISDN emulator  
(2) PC/workstations  
(2) console cables

Lab Design:

Step-By-Step Instructions:
1. Configure the basics on each router. Set up and cable the lab as shown. Pick a routing protocol to use and advertise the networks.
2. Set the ISDN switch type on each router:

   smarts(config)#isdn switch-type basic-5ess  
   smarts(config)#dialer-list 1 protocol ip permit  

   kissane(config)#isdn switch-type basic-5ess  
   kissane(config)#dialer-list 1 protocol ip permit
3. Configure the ISDN interface on “smarts.” You will be configuring the ip address, “no shut,” dialer group, and a dialer map (how to get from here to there).

```bash
smarts(config)#int bri0/0
smarts(config-if)#ip address 192.168.1.1 255.255.255.0
smarts(config-if)#no shut
smarts(config-if)#dialer-group 1
smarts(config-if)#dialer map ip 192.168.1.2 name kissane 5552000
```

4. Configure the ISDN interface on “kissane” with similar commands. You will be configuring the ip address, the isdn spid (service profile identifiers), and a dialer map (how to get from here to there).

```bash
kissane(config)#int bri0/0
kissane(config-if)#ip address 192.168.1.2 255.255.255.0
kissane(config-if)#no shut
kissane(config-if)#dialer-group 1
kissane(config-if)#dialer map ip 192.168.1.1 name smarts 5551000
```

5. Test the connection using ping, sh ip route, and sh cdp nei from BRI0/0 to BRI0/0. Use “show isdn status” to inspect the status of the BRI interfaces. You should see:

```bash
smarts#sh isdn status
Global ISDN Switchtype = basic-5ess
ISDN BRI0/0 interface
dsl 0, interface ISDN Switchtype = basic-5ess
Layer 1 Status:
ACTIVE
Layer 2 Status:
TEI = 86, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
Layer 3 Status:
1 Active Layer 3 Call(s)
Activated dsl 0 CCBs = 1
CCB:callid=8001, sapi=0, ces=1, B-chan=1, callytype=DATA
The Free Channel Mask: 0x80000002
Total Allocated ISDN CCBs = 1
smarts#
```

Here we can see an “active” good ISDN connection.

6. Try to ping the loopback. You should not be able to see it. It should not have shown up in the ip routing table either. The ISDN line comes up, stays active, and then shuts off pretty quickly. Its actually faster than the routing protocol (I used EIGRP). In order to make this work we need to set up some static routes between the two.

```bash
smarts(config)#ip route 192.168.200.0 255.255.255.0 192.168.1.2
```
This route basically is saying, “in order to get to the 192.168.200.0/24 network use the 192.168.1.2 interface.”

kissane(config)#ip route 192.168.100.0 255.255.255.0 192.168.1.1

This route basically is saying, “in order to get to the 192.168.100.0/24 network use the 192.168.1.1 interface.” You should be able to ping and see all networks. Now you should see:

kissane#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds: !!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 32/32/33 ms

00:21:07: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
00:21:07: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551000
00:21:08: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0/0:1, changed state to up

kissane#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds: !!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms

kissane#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR
Gateway of last resort is 192.168.1.1 to network 0.0.0.0

C 192.168.200.0/24 is directly connected, Loopback0
C 192.168.1.0/24 is directly connected, BRI0/0
S 192.168.100.0/24 [1/0] via 192.168.1.1
S* 0.0.0.0/0 [1/0] via 192.168.1.1
Challenge Lab or Supplemental Activities:
1. Repeat the lab using Class “B” addresses.
2. Repeat the lab using Class “A” addresses.
3. Use the help features to find out all commands for isdn and what they mean.

So what have I learned here?
In this lab you have learned how to set up the bare minimum requirements for an ISDN BRI connection using the MERGE ISDN simulators. In later labs you will learn more about “real-world” applications using PPP, ISDN SPID’s and Dial on Demand Routing (DDR).

Guest Router Name Derivation

Timothy Kissane was a software developer for a company called System Management Arts Incorporated (“SMARTS”). When he was hired he signed a confidentiality agreement that he would never reveal any of the source code he developed for a program called “InCharge” (a network monitoring program). After he was fired a couple of the competitors to SMARTs received email messages from “Joe Friday” via Hotmail offering the source code for InCharge for sale. These were forwarded from the competitors back to SMARTS (aha! They are all in it together!). He was arrested in February 2002, released on bail and is awaiting trial on charges of “theft of a trade secret” in connection with his prior employment.

Hungry for more???

You are almost there for the CompTIA Network+ and Security+ Certifications too!
For more information:

http://www.comptia.org/certification/default.aspx

Building successful careers for students everywhere…
Basic ISDN Configuration with BRI interface (ADTRAN)

**Objective:**
To learn how to set up a basic ISDN connection, using BRI interfaces, between two routers. In this lab you will be using an ADTRAN box for ISDN emulation.

**Tools and Materials:**
(2) routers
(2) straight-through cables
(1) workstation
(1) ADTRAN Atlas 550
(2) PC/workstations
(2) console cables

**Lab Design:**

```
                     st | 1 | 2 | st
```

<table>
<thead>
<tr>
<th>Router</th>
<th>Smarts</th>
<th>Kissane</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loop 0</td>
<td>192.168.100.1/24</td>
<td>192.168.200.1/24</td>
</tr>
<tr>
<td>BR10</td>
<td>192.168.1.1/24</td>
<td>192.168.1.2/24</td>
</tr>
<tr>
<td>Adtran Port</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Phone number</td>
<td>555-1234</td>
<td>555-4000</td>
</tr>
</tbody>
</table>

**Before we begin...**
The Adtran units are quirky as all get out with ISDN. If you are having problems then save all of your configurations, power everything down, bring up the Adtran, wait about 30 seconds after it is through cycling, then turn on everything. When doing ISDN troubleshooting we have found that 9 times of 10 it is just a proof-reading/fat-finger error somewhere. Have a friend proofread for you. Remember that user names and passwords are case sensitive. Did you skip putting in the router basics? Do you have the correct IP addresses and masks? Also, we have also found the Adtran to have all kinds of problems with IOS 12.3...everything works fine but your show and debug outputs will show tons of errors and weird things. If you can pass the traffic, then assume it is working fine.
Step-By-Step Instructions:
1. Configure the basics on each router. Set up and cable the lab as shown. Pick a routing protocol to use and advertise the networks.
2. Set the ISDN switch type on each router:

   smarts(config)#isdn switch-type basic-ni
   smarts(config)#dialer-list 1 protocol ip permit
   
   kissane(config)#isdn switch-type basic-ni
   kissane(config)#dialer-list 1 protocol ip permit

3. Configure the ISDN interface on “smarts.” You will be configuring the ip address, the isdn spid (service profile identifiers), and a dialer map (how to get from here to there).

   smarts(config)#int bri0/0
   smarts(config-if)#ip address 192.168.1.1 255.255.255.0
   smarts(config-if)#no shut
   smarts(config-if)#dialer-group 1
   smarts(config-if)#dialer map ip 192.168.1.2 name kissane 5554000

4. Configure the ISDN interface on “kissane” with similar commands. You will be configuring the ip address, the isdn spid (service profile identifiers), and a dialer map (how to get from here to there).

   kissane(config)#int bri0/0
   kissane(config-if)#ip address 192.168.1.2 255.255.255.0
   kissane(config-if)#no shut
   kissane(config-if)#dialer-group 1
   kissane(config-if)#dialer map ip 192.168.1.1 name smarts 5551234

5. Test the connection using ping, sh ip route, and sh cdp nei from BRI0/0 to BR10/0. Use “show isdn status” to inspect the status of the BRI interfaces. You should see:

   kissane#ping 192.168.100.1

   Type escape sequence to abort.
   Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is
   2 seconds:
   ....
   Success rate is 0 percent (0/5)
   kissane#sh isdn status
   Global ISDN Switchtype = basic-ni
   ISDN BR10/0 interface
       dsl 0, interface ISDN Switchtype = basic-ni
       Layer 1 Status: ACTIVE
       Layer 2 Status:
Layer 2 NOT Activated
Layer 3 Status:
  0 Active Layer 3 Call(s)
Activated dsl 0 CCBs = 1
  CCB:callid=0x8002, sapi=0x0, ces=0x1, B-chan=1
The Free Channel Mask: 0x80000002
Total Allocated ISDN CCBs = 1

Here we can see a problem with our ISDN connection. Unlike the MERGE box we need to include service profile identifiers, ppp and authentication.

```
smarts(config-if)#isdn spid1 51055512340001 5551234
smarts(config-if)#isdn spid2 51055512350001 5551235
smarts(config-if)#enc ppp
smarts(config-if)#ppp authentication chap

smarts(config)#username kissane password 0 cisco
smarts(config)#ip host kissane 192.168.1.2

kissane(config-if)#isdn spid1 51055540000001 5554000
kissane(config-if)#isdn spid2 51055540010001 5554001
kissane(config-if)#enc ppp
kissane(config-if)#ppp authentication chap

kissane(config)#username smarts password 0 cisco
kissane(config)#ip host smarts 192.168.1.1
```

6. Try to ping the loopback. You should not be able to see it. It should not have shown up in the ip routing table either. The ISDN line comes up, stays active, and then shuts off pretty quickly. Its actually faster than the routing protocol (I used EIGRP). In order to make this work we need to set up some static routes and a quad-zero (“gateway of last resort”) between the two.

```
smarts(config)#ip route 192.168.200.0 255.255.255.0 192.168.1.2

This route basically is saying, “in order to get to the 192.168.200.0/24 network use the 192.168.1.2 interface.”

kissane(config)#ip route 192.168.100.0 255.255.255.0 192.168.1.1
kissane(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.2

This route basically is saying, “in order to get to the 192.168.100.0/24 network use the 192.168.1.1 interface.” You should be able to ping and see all networks. Now you should see:

kissane#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2
seconds: .!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max =
  32/32/33 ms
```
00:21:07: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
00:21:07: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551000
00:21:08: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0/0:1, changed state to up

kissane#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds: !!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms

kissane#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -BGP
     D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
     i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
     U - per-user static route, o - ODR
Gateway of last resort is 192.168.1.1 to network 0.0.0.0

C    192.168.200.0/24 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, BRI0/0
S    192.168.100.0/24 [1/0] via 192.168.1.1
S*   0.0.0.0/0 [1/0] via 192.168.1.1

Challenge Lab or Supplemental Activities:
1. Repeat the lab using Class “B” addresses.
2. Repeat the lab using Class “A” addresses.
3. Try using PAP as an encapsulation for PPP. Try HDLC.

So what have I learned here?
In this lab you have learned how to set up the bare minimum requirements for an ISDN BRI connection using the ADTRAN ISDN simulators. In later labs you will learn more about “real-world” applications using PPP, ISDN SPID’s and Dial on Demand Routing (DDR).

Guest Router Name Derivation

Timothy Kissane was a software developer for a company called System Management Arts Incorporated (“SMARTS”). When he was hired he signed a confidentiality agreement that he would never reveal any of the source code he developed for a program called “InCharge” (a network monitoring program). After he was fired a couple of the competitors to SMARTs received email messages from “Joe Friday” via Hotmail offering the source code for InCharge for sale. These were forwarded from the competitors back to SMARTs (aha! They are all in it together!). He was arrested in February 2002, released on bail and is awaiting trial on charges of “theft of a trade secret” in connection with his prior employment.
Electronic technology continues to infiltrate industry, electrostatic discharge threatens operations causing costly damage, injury and downtime. Pro-Cote Inc. can protect your work environment against destructive static shock with Valspar ESD Epoxy Flooring Systems.

Certified Conductive and Static Dissipative applications for every need:

- Facilities with highly sensitive electronic equipment
- Computer facilities
- Electronic manufacturing & assembly
- Facilities containing AGV lines
- Clean rooms
- Manufacturing/processing flammable or combustible materials
- Munitions plants

"On Site Certification"

Specialty Applications:

- Non-skid coatings
- Chemical resistant urethanes
- Epoxy coatings
- Moisture cured urethane
- Seamless epoxy floor systems
- Waterborne systems
- VOC compliant systems
- Concrete leveling
- Concrete resurfacing
- Control joint repair

The Management at Pro-Cote Inc. wants to put its expertise to work for you. With over 55 years of combined experience, our goal is to be the best specialty coatings applicator company, period.

**Application Experience**

- Industrial floor coatings
- Manufacturing production areas
- Distribution and warehouse floors
- Loading docks and ramps
- Safety zones and work areas
- Battery charging areas
- UPS rooms
- Parking garages

Let us help with your specification needs.

**“Safety Solutions”**

Pro-Cote Inc. is a proud installer of Valspar Flooring Systems

www.valsparflooring.com

Pro-Cote Inc.
P. O. Box 1766
Largo, FL 33779
Toll Free (877) 588-4300

www.pro-cote-usa.com

ISDN Operation and Troubleshooting
Objective:  
This paper lab explains the fundamentals of ISDN operation. Here we will start with the theory of ISDN operation, then break it down a little more in-depth layer by layer, discuss troubleshooting commands for ISDN, and then finish by looking at how to decipher the debug and show command outputs of working and non-working ISDN lines.

ISDN Theory of Operation:  
ISDN, as a WAN technology, is fairly simple: once you know how to set it up and use it. It is a technology that has been around for a while now and is used for main WAN connections or, more likely, as backup connections for other WAN technologies. Once you understand how ISDN operates you should be more likely to understand what you need to set up on your routers and how to troubleshoot it.

ISDN operation is a simple (I think it is...) three-step operation that correlates nicely with the lower three layers of the OSI model:

1. ISDN DDR generates “interesting” traffic  
2. ISDN call is made  
3. PPP handshaking

Then you are ready to go! Let’s look at each step a bit more in-depth.

Layer-By-Layer ISDN Operation:

Physical Layer
As we discuss the steps they will be numbered and correlated to the router configuration. Use this to correlate the discussion (“the theory”) with the implementation (“learning by doing”). (1) Of course no traffic will pass through a physical interface if it is physically “shut down” so we must also configure our interfaces to be up during this phase. (2) ISDN uses Dial on Demand Routing (DDR) to establish the first phase of connection at the physical layer. We set up access control lists in our configuration that determine “what is” and “what is not interesting traffic.” This will decide whether or not we move on to the second phase. Finally you may see the term “spoofing” used during troubleshooting or checking the status of an ISDN connection. The router “spoofs” (fakes) a connection during the set up phases to imitate an active state, otherwise the next steps could not take place. Some commands that must be used to set up a basic ISDN connection include:

```
router(config)#int bri0/0  
(1)
router(config-if)#ip address 192.168.1.1 255.255.255.0  
(1)
router(config-if)#no shut  
(1)
router(config-if)#dialer-group 1  
(2)
router(config-if)#dialer map ip 192.168.1.2 name routerB 5552000  
(2)
router(config)#dialer-list 1 protocol ip permit  
(2)
```
**Data Link Layer**
Two things happen here: The bearer channel (B-channel) is set up and the data channels (D-channel) are set up. This, essentially, is how a call is made. There is a bit of overlap with the physical layer (dialer strings/maps) much. (1) The D-channel is uses a protocol called “Link Access Protocol-D” or LAPD. This uses Q.921 for establishment. Therefore it makes sense for us to debug Q.921 during troubleshooting. (2) If a protocol is used then it must hand-shake (establish, negotiate, and maintain of LCP-Link Control Protocol). This is where service profile identifiers (SPID) may or may not be used (a.k.a “TEI”)and username/password problems can be found. Also, certain manufacturers of networking equipment do not require specific encapsulations. Nine times out of ten they do require PPP for encapsulation. For example, MERGE boxes do not require PPP but ADTRAN units do require PPP. Good stuff to know when setting them up.

```
router(config-if)#dialer map ip 192.168.1.2 name routerB 5552000 (2)
router(config-if)#isdn spid1 51055512340001 5551234 (2)
router(config-if)#isdn spid2 51055512350001 5551235 (2)
router(config-if)#enc ppp (2)
router(config-if)#ppp authentication chap (2)
router(config)#username joe password cisco (2)
router(config)#ip host joe 192.168.1.1 (2)
```

**Network Layer**
(1) This is where our network layer implementation of PPP takes place (NCP-Network Control Protocol). This is where username and password problems can also be found. (Ok so there is some overlap). Here we will also find the Q.931 protocol to finish our ISDN connection.

```
router(config)#username joe password cisco (2)
router(config)#ip host joe 192.168.1.1 (2)
```

**Troubleshooting ISDN:**
Just like we have done before you will start at the physical layer and work your way up the OSI model:

1. ISDN DDR generates “interesting” traffic  
2. ISDN call is made  
3. PPP handshaking

**Physical layer:**
Step 1: Since ISDN uses interesting traffic to initiate a call we must first generate interesting traffic. The easiest way is to ping the other ISDN interface to see if the line comes up. To check for interesting traffic beyond what happens we can use these commands (use them in this order too):

```
sh controllers bri
sh int bri0/0
sh protocols
debug dialer packets
debug dialer
```
Problems that cause ISDN to not work: no cable, dialer interface shut down, dialer list configured improperly or not at all, or problems with the dialer string/map.

**Data Link layer:**
Step 2: We need to see if our LAP-D and PPP are completing properly. We can use these commands for that:

```plaintext
debug isdn q921  (LAPD)
debug ppp negotiation  (PPP)
```

Problems that cause ISDN to not work: problems with the dialer string/map, problems with the layer 2 ISDN line, or problems with ppp.

**Network layer:**
Step 3: Check for confirmation of a good connection using these commands:

```plaintext
debug isdn q931
show isdn status
```

Problems that cause ISDN to not work: problems with the dialer string/map, problems with the layer 2 ISDN line, or problems with ppp.

**ISDN Network Scenarios:**
Ok. Great. Now you are ready to fire up an ISDN connection and troubleshoot it with no problems, right? Maybe. Let’s take some time to look at some of the output from these debug and show commands. We have already seen how cryptic they can be. For these let’s work from the network layer down. We will show a good connection and then introduce problems and see how the debug/show commands change with problems and what causes them.

Here is the output of each of those commands for a good working ISDN connection using an ADTRAN between two routers. To bring the line up I pinged the BRI interface. You can see the first ping packet does not work. It generates the interesting traffic, the BRI line comes up and the other four succeed.

```plaintext
kissane#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:....
Success rate is 80 percent (4/5), round-trip min/avg/max = 32/32/32 ms
00:13:26: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
00:13:26: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551234
00:13:27: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0/0:1, changed state to up
00:13:27: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551234 smarts
```
Network Layer:
With a good, active connection we can see our ISDN active status:

```
kissane#sh isdn status
Global ISDN Switchtype = basic-ni
ISDN BRI0/0 interface
dsl 0, interface ISDN Switchtype = basic-ni
Layer 1 Status:
ACTIVE
Layer 2 Status:
   TEI = 64, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
   TEI = 65, Ces = 2, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
Spid Status:
   TEI 64, ces = 1, state = 8(established)
      spid1 configured, spid1 sent, spid1 valid
   Endpoint ID Info: epsf = 0, usid = 70, tid = 1
   TEI 65, ces = 2, state = 5(init)
      spid2 configured, spid2 sent, spid2 valid
   Endpoint ID Info: epsf = 0, usid = 70, tid = 2
Layer 3 Status:
   1 Active Layer 3 Call(s)
Activated dsl 0 CCBs = 1
   CCB:callid=0x8004, sapi=0x0, ces=0x1, B-chan=1
   The Free Channel Mask: 0x80000002
   Total Allocated ISDN CCBs = 1
kissane#
```

We can see our layer 1 status is “active.” Our layer 2 has two active “multiple frames established” which is one for each spid. Finally our layer 3 has one active call. To see more details about that call:

```
kissane#sh isdn active
```

```
-----------------------------------------------------------------------
ISDN ACTIVE CALLS
-----------------------------------------------------------------------
History table has a maximum of 100 entries.
History table data is retained for a maximum of 15 Minutes.
-----------------------------------------------------------------------
          Call  Calling   Called    Remote  Seconds Seconds Seconds Charges
Call Type   Number   Number   Name    Used   Left   Idle   Units/Currency
-----------------------------------------------------------------------
Out 5551234    5551234  smarts 9    114    5  0
```

Finally we can see what happens if everything is fine with our debug isdn q931 command. First I waited until the BRI connection was administratively down. Then I enabled the debug command. Finally I pinged the other BRI to bring the line back up.
You should see:

```
kissane# debug isdn q931
ISDN Q931 packets debugging is on
```

```
kissane# ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds: .!!!.
Success rate is 80 percent (4/5), round-trip min/avg/max = 32/33/36 ms
```

```
01:33:55: ISDN BR0/0: TX -> SETUP pd = 8  callref = 0x07
01:33:55:  Bearer Capability i = 0x8890
01:33:55:  Channel ID i = 0x83
01:33:55:  Keypad Facility i = '5551234'
01:33:236232422420: ISDN BR0/0: RX <- CALL_PROC pd = 8  callref = 0x87
01:33:236232201280:  Channel ID i = 0x89
01:33:236232422420: ISDN BR0/0: RX <- CONNECT pd = 8  callref = 0x87
01:33:236232201280:  Channel ID i = 0x89
01:33:55: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
01:33:55: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551234
```

Let’s decipher those bold statements for our Q931 debug.

1. TX SETUP
2. RX CALL_PROC
3. RX CONNECT
4. Interface BRI0/0:1, changed state to up (on the destination side)
5. Interface BRI0/0:1 is now connected to 5551234 (on the destination side)
6. TX CONNECT_ACK
7. Line protocol on Interface BRI0/0:1, changed state to up (on the source side)

In line 1 we see our Q931 requesting (transmission: TX) a handshake procedure to setup an ISDN connection with certain parameters. Here the source is asking “can I connect to you?” Line 2 is the reception of the setup request to allow the “call to proceed.” In other words, the destination is responding with “I am not busy so you can connect to me.” Line 3 is our source actually connecting to the destination. Line 4-5 shows us the line coming up and connected with a number (555-1234). Line 6 is our destination telling the source “the line is established so you can bring up your interface and start sending information.” Line 7 shows us the source BRI is brought up and we can now transmit our data.
Data Link Layer:
Let’s start with our LAP-D negotiation (Q.921). This will negotiate the setting up of our SPID’s. Again, I used a BRI that was not connected, enabled the debug isdn q921, and then pinged the other BRI. (Don’t forget to turn off debug from the last step…it would be too confusing).

kissane# debug isdn q921
ISDN Q921 packets debugging is on
kissane# ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds: .!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 32/32/32 ms

01:52:21: ISDN BR0/0: TX -> INFOc sapi = 0  tei = 64  ns = 22  nr = 20  i = 0x0
8010805040288901801B23CC73535331323334
01:52:21:90194354176: ISDN BR0/0: RX <- RRr sapi = 0  tei = 64  nr = 23
01:52:21:9235625684: ISDN BR0/0: RX <- INFOc sapi = 0  tei = 64  ns = 20  nr = 23  i = 0x08018802180189
01:52:21: ISDN BR0/0: TX -> RRr sapi = 0  tei = 64  nr = 21
01:52:21:90194354176: ISDN BR0/0: RX <- INFOc sapi = 0  tei = 64  ns = 21  nr = 23  i = 0x08018807180189
01:52:21: ISDN BR0/0: TX -> RRr sapi = 0  tei = 64  nr = 22
01:52:21: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
01:52:21: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551234
01:52:21: ISDN BR0/0: TX -> INFOc sapi = 0  tei = 64  ns = 23  nr = 22  i = 0x0
801080F
01:52:21:90194313216: ISDN BR0/0: RX <- RRr sapi = 0  tei = 64  nr = 24
01:52:22: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0/0:1, changed state to up
kissane#
01:52:103079256064: ISDN BR0/0: RX <- RRp sapi = 0  tei = 65  nr = 1
01:52:24: ISDN BR0/0: TX -> RRf sapi = 0  tei = 65  nr = 1
kissane#
01:52:27: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551234 smarts

kissane#
01:52:13 3144027136: ISDN BR0/0: RX <- RRp sapi = 0  tei = 64  nr = 24
01:52:31: ISDN BR0/0: TX -> RRf sapi = 0  tei = 64  nr = 22
Once again, let’s cut out all the mumbo-jumbo and look at the text in “bold.”

1. TX -> INFOc sapi = 0 tei = 64 (from source)
2. ISDN BR0/0: RX <= RRr sapi = 0 tei = 64 (from destination)
3. BR0/0: RX <-> INFOc sapi = 0 tei = 64 (from destination)
4. ISDN BR0/0: TX -> RRr sapi = 0 tei = 64 (from source)
5. ISDN BR0/0: RX <-> INFOc sapi = 0 tei = 64 (from destination)
6. ISDN BR0/0: TX -> RRr sapi = 0 tei = 64 (from source)
7. Interface BRIO/0:1, changed state to up
8. Interface BRIO/0:1 is now connected to 5551234
9. TX -> INFOc sapi = 0 tei = 64 (from source)
10. ISDN BR0/0: RX<= RRr sapi = 0 tei = 64 (from destination)
11. Line protocol on Interface BRIO/0:1, changed state to up
12. Interface BRIO/0:1 is now connected to 5551234 smarts
13. 01:52:13 ISDN BR0/0:RX<-RRp sapi = 0 tei = 64 (from destination)
14. 01:52:31: ISDN BR0/0:TX -> RRf sapi = 0 tei = 64 (from source)

In line 1 we see our source BRI requesting services from the destination using tei=64 (spid 1). Then, in line 2, our destination acknowledges our request from our source. In line 3 the negotiation for services begins when the destination requests user information. In line 4 the requested information is sent to the destination. In line 5 the destination sends acknowledgement of receipt of that information and, in line 6, the source sends acknowledgement of receipt of the destination’s acknowledgement of receipt of that information. In line 7 and 8 our BRI comes up. In line 9 our source sends a message to the destination that they are up and ready. Line 10 shows the destination acknowledging the readiness. Then, in line 11 and 12, the destination BRI’s comes up and we are ready to go. Lines 13 and 14 are packets, which are periodically sent between source and destination to let each other know they are up, and operating. These will continue as long as the BRI line is active. Unlike other broadcasts they are not sent every X seconds. Watch the counters...they tend to decrement exponentially.

Now let’s debug our PPP. If you have already done the PPP with authentication lab then you are already familiar with the process.

```
kissane# debug ppp negotiation
PPP protocol negotiation debugging is on
kissane# ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:....!!
Success rate is 60 percent (3/5), round-trip min/avg/max = 32/32/32 ms
02:04:20: %LINK-3-UPDOWN: Interface BRIO/0:1, changed state to up
02:04:20: %ISDN-6-CONNECT: Interface BRIO/0:1 is now connected to 5551234
02:04:20: BR0/0:1 PPP: Treating connection as a callout
02:04:20: BR0/0:1 PPP: Phase is ESTABLISHING, Active Open
02:04:20: BR0/0:1 LCP: O CONFREQ [Closed] id 27 len 15
```
Then let’s strip it down. If you recall PPP sets up in three stages (hence a three-way hand-shaking process): LCP, IPCP, and CDPCP.

1. PPP: Phase is ESTABLISHING
2. LCP: O CONFREQ  
3. LCP: AuthProto CHAP (from source)
4. LCP: MagicNumber 0x0208BEB7  (from source)
5. LCP: MagicNumber 0x045572BC  (from source)
6. LCP: O CONFACK [REQsent]  (from destination)
7. LCP: AuthProto CHAP  (from destination)
8. LCP: TIMEout: State ACKsent  (from source)
9. LCP: O CONFREQ [ACKsent]  (from source)
10. PPP: Phase is AUTHENTICATING, by both
11. CHAP: O CHALLENGE  (from source)
12. LCP: I CONFREQ [REQsent]  (from source)
13. CHAP: I CHALLENGE  (from source)
14. LCP: O CONFACK [REQsent]  (from source)
15. CHAP: I SUCCESS  (from source)
16. LCP: TIMEout: State ACKsent  (from source)
17. CHAP: O SUCCESS  (from source)
18. PPP: Phase is UP
19. IPCP: O CONFREQ [Closed]  (from source)
20. IPCP: Address 192.168.1.2  (from source)
21. IPCP: I CONFREQ [REQsent]  (from source)
22. IPCP: Address 192.168.1.1  (from source)
23. IPCP: O CONFACK [REQsent]  (from destination)
24. IPCP: I CONFACK [ACKsent]  (from destination)
25. IPCP: O CONFACK [ACKsent]  (from destination)
26. CDPCP: I CONFACK [ACKsent]  (from destination)
27. IPCP: O CONFACK [ACKsent]  (from destination)
28. IPCP: State is Open
29. CDPCP: I CONFACK [ACKsent]  (from destination)
30. IPCP: Install route to 192.168.1.1  (from source)
31. %LINEPROTO-5-UPDOWN: Line protocol on Interface BR10/0:1, changed state to up  
32. IPCP: Address 192.168.1.2  (from source)
33. %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551234 smarts

586
In line 1 we see our PPP request beginning. Line 2 shows us a request from our source to start an LCP session. Line 3 shows our destination requesting CHAP password authentication from the source. Lines 4-7 repeat this process until, in line 8, the CHAP password authentication times out. (See? Nothing is perfect). In line 9 a request from our source to start an LCP session is repeated. Line 10 shows our destination requesting a CHAP password for authentication. Line 11 shows acknowledgement of the CHAP request. Line 12 shows acknowledgement of the acknowledgement that the information requesting CHAP password verification, the LCP state is set to open, and the next phase of PPP establishment starts. Lines 15-20 show us a similar process for verifying the CHAP password. Line 21 sets our PPP phase (LCP) as up. Line 22 starts our IPCP negotiation. (This intermingles with CDPCP so I will break them out separately.) Here a request for the BRI ip address of the destination is requested. Lines 23-28 and 21-33 show the exchange of ip addresses between source and destination. Lines 29-30 and 34-25 show the CDPCP exchange sequence. Finally our route is installed in line 36. Then our state is up and connected in lines 37-38.

Physical Layer:
Now lets look at our dialer events and interface states.

kissane#debug dialer
Dial on demand events debugging is on
1. 02:07:54: BR0/0 DDR: Dialing cause ip (s=192.168.1.2, d=192.168.1.1)
2. 02:07:54: BR0/0 DDR: Attempting to dial 5551234
3. 02:07:54: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
4. 02:07:54: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551234
5. 02:07:56: BR0/0:1 DDR: dialer protocol up
6. 02:07:57: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0/0:1, changed state to up
7. 02:08:00: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551234 smarts

Since there is not a lot here let’s just go line by line. Line 1 shows our DDR dialing with source and destination addresses. Then, in line 2 we dial our destination number set in our dialer map statement. Our state comes up on our source, we are connected, the dialer protocol comes up, our state comes up on our destination, and our BRI line is connected. Not too tough. For a more exacting look combine the debug dialer with debug ppp negotiation.

Debug dialer packet gives us similar information but includes the icmp information. I will let you figure out what is happening here (hints in bold):

```
debug dialer packets
Dial on demand packets debugging is on
ping 192.168.1.1
```

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
Success rate is 60 percent (3/5), round-trip min/avg/max = 32/32/32 ms

03:02:19: BR0/0 DDR: ip (s=192.168.1.2, d=224.0.0.10), 60 bytes, outgoing interesting (ip PERMIT)
03:02:20: BR0/0 DDR: ip (s=192.168.1.2, d=192.168.1.1), 100 bytes, outgoing interesting (ip PERMIT)
03:02:20: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
03:02:20: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551234
03:02:22: BR0/0 DDR: ip (s=192.168.1.2, d=192.168.1.1), 100 bytes, outgoing interesting (ip PERMIT)
03:02:23: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0/0:1, changed state to up
03:02:23: BR0/0 DDR: cdp, 277 bytes, outgoing uninteresting (no list matched)
03:02:23: BR0/0 DDR: cdp, 277 bytes, outgoing uninteresting (no list matched)
03:02:23: BR0/0 DDR: cdp, 277 bytes, outgoing uninteresting (no list matched)
03:02:23: BR0/0 DDR: ip (s=192.168.1.2, d=224.0.0.10), 60 bytes, outgoing interesting (ip PERMIT)
03:02:24: BR0/0 DDR: ip (s=192.168.1.2, d=192.168.1.1), 100 bytes, outgoing interesting (ip PERMIT)
```
We never want to forget about our interface states.

kissane# sh controller bri
BRI unit 0:BRI unit 0 with U interface:
Layer 1 internal state is ACTIVATED
Layer 1 U interface is ACTIVATED.
ISDN Line Information:
  Last C/I from ISDN transceiver:
    AI:Activation Indication
  Last C/I to ISDN transceiver:
    AI:Activation Indication
Current EOC commands:
  RTN - Return to normal

(there is a ton of information with this one...I cut it off here).

kissane# sh int bri0/0
BRI0/0 is up (spoofing), line protocol is up (spoofing)
  Hardware is PQUICC BRI with U interface
  Internet address is 192.168.1.2/24
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set
  Last input 00:04:09, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/34 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations 0/1/256 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  3256 packets input, 13283 bytes, 0 no buffer
  Received 1 broadcasts, 0 runts, 0 giants, 0 throttles
  34 input errors, 34 CRC, 0 frame, 0 overrun, 0 ignored, 19
  abort
  3256 packets output, 13475 bytes, 0 underruns
  0 output errors, 0 collisions, 6 interface resets
  0 output buffer failures, 0 output buffers swapped out
  6 carrier transitions
kissane#
kissane#sh prot
Global values:
Internet Protocol routing is enabled
Ethernet0/0 is administratively down, line protocol is down
Serial0/0 is administratively down, line protocol is down
BRIO/0 is up, line protocol is up
  Internet address is 192.168.1.2/24
BRIO/0:1 is down, line protocol is down
BRIO/0:2 is down, line protocol is down
Serial0/1 is administratively down, line protocol is down
Loopback0 is up, line protocol is up
  Internet address is 192.168.200.1/24
kissane#

Troubleshooting in Action:
Now let’s see what happens to each of these troubleshooting outputs when you have problems in your network.

Dialer interface shut down (layer 1)
No isdn cable connected (layer 1)

No dialer list (layer 2)
Incorrect dialer list (layer 2)

Incorrect spid (layer 2/3)
No ppp (layer 2/3)
Incorrect username/password (layer 2/3)
Missing username/password (layer 2/3)

Bad ip address/mask (layer 3)

The first thing you will want to do is “sh isdn status.” This will pin-point the layer where your trouble lies.

Dialer interface shut down (layer 1)
Here, since we always start at layer 1 when troubleshooting, we want to check our cables and interfaces. Debug dialer and debug dialer packets will show absolutely nothing if you try to ping the other interface.

kissane#sh isdn status
Global ISDN Switchtype = basic-ni
ISDN BRIO/0 interface
dsi 0, interface ISDN Switchtype = basic-ni
Layer 1 Status:
DEACTIVATED
Layer 2 Status:
TEI = 64, Ces = 1, SAPI = 0, State = TEI_ASSIGNED
TEI = 65, Ces = 2, SAPI = 0, State = TEI_ASSIGNED
Spid Status:
TEI 64, ces = 1, state = 5(init)
  spid1 configured, spid1 sent, spid1 valid
Endpoint ID Info: epsf = 0, usid = 70, tid = 1  
  TEI 65, ces = 2, state = 5(init)  
  spid2 configured, spid2 sent, spid2 valid
Endpoint ID Info: epsf = 0, usid = 70, tid = 2
Layer 3 Status:  
  0 Active Layer 3 Call(s)
Activated dsl 0 CCBs = 0
The Free Channel Mask: 0x80000003
Total Allocated ISDN CCBs = 0

kissane#sh controller bri
BRI unit 0:BRI unit 0 with U interface:
Layer 1 internal state is DEACTIVATED
Layer 1 U interface is ACTIVATED.

kissane#sh int bri0/0
BRI0/0 is administratively down, line protocol is down
  Hardware is PQUICC BRI with U interface
  Internet address is 192.168.1.2/24
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set

kissane#sh prot
Global values:
  Internet Protocol routing is enabled
Ethernet0/0 is administratively down, line protocol is down
Serial0/0 is administratively down, line protocol is down
BRI0/0 is administratively down, line protocol is down
  Internet address is 192.168.1.2/24
BRI0/0:1 is administratively down, line protocol is down
BRI0/0:2 is administratively down, line protocol is down
Serial0/1 is administratively down, line protocol is down
  Internet address is 192.168.200.1/24
Loopback0 is up, line protocol is up

To fix this, go into your configuration under the BRI interfaces and type “no shut.”

No ISDN cable connected (layer 1)
If you are trying these one at a time, then verify good proper operation after trying each.
It will keep you from being confused. For this scenario I just unplugged one of the BRI
straight-through cables from the BRI interface on the router. Interestingly sh int an sh
protocols still show the line as up. Only the sh controllers show it being deactivated. If it
is not connected then the debugs will not work. Ok. So you are wondering why not just
look at the connections in front of you? Because sometimes one router will be in Detroit
and the other one will be in Chicago (or something like that) and it is nice to “learn” if a
cable is physically connected without physically being there.

kissane#sh isdn status
Global ISDN Switchtype = basic-ni
ISDN BRI0/0 interface
  dsl 0, interface ISDN Switchtype = basic-ni
Layer 1 Status:
  DEACTIVATED
Layer 2 Status:
   TEI = 64, Ces = 1, SAPI = 0, State = TEI_ASSIGNED
Spid Status:
   TEI 64, ces = 1, state = 5(init)
      spid1 configured, spid1 sent, spid1 valid
   TEI Not Assigned, ces = 2, state = 1(terminal down)
      spid2 configured, spid2 NOT sent, spid2 NOT valid
Layer 3 Status:
   0 Active Layer 3 Call(s)
Activated dsl 0 CCBs = 0
The Free Channel Mask:  0x80000003
Total Allocated ISDN CCBs = 0

Ok. Since we troubleshoot from the bottom-up and we see a layer 1 problem let’s go through the layer 1 troubleshooting tools.

kissane# sh controllers bri
BRI unit 0:BRI unit 0 with U interface:
   Layer 1 internal state is DEACTIVATED
   Layer 1 U interface is DEACTIVATED.

(rest of output omitted—about 5 pages worth!)

Just for good measure I added in layer 2 and 3 outputs from those commands that had some output:

kissane# debug dialer packets
Dial on demand packets debugging is on
kissane# ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:.....
Success rate is 0 percent (0/5)
05:05:33: BR0/0 DDR: ip (s=192.168.1.2, d=192.168.1.1), 100 bytes, outgoing interesting (ip PERMIT)
05:05:33: BR0/0 DDR: ip (s=192.168.1.2, d=224.0.0.10), 60 bytes, outgoing interesting (ip PERMIT)
05:05:34: BR0/0 DDR: cdp, 277 bytes, outgoing uninteresting (no list matched)

kissane# debug dialer
Dial on demand events debugging is on
kissane# ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:.....
Success rate is 0 percent (0/5)
kissane#
05:04:28: BR0/0 DDR: Dialing cause ip (s=192.168.1.2, d=192.168.1.1)
05:04:28: BR0/0 DDR: Attempting to dial 5551234
05:04:35: BR10/0: wait for isdn carrier timeout, call id=0x8014
05:04:35: BR0/0:1 DDR: disconnecting call
05:04:35: BR0/0:2 DDR: disconnecting call
05:04:36: BR0/0 DDR: Dialing cause ip (s=192.168.1.2, d=192.168.1.1)
05:04:36: BR0/0 DDR: Attempting to dial 5551234
kissane#

kissane#debug ppp nego
PPP protocol negotiation debugging is on
kissane#ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:......
Success rate is 0 percent (0/5)
kissane#
05:14:06: BR0/0:1 LCP: State is Closed
05:14:06: BR0/0:1 PPP: Phase is DOWN
05:14:06: BR0/0:2 LCP: State is Closed
05:14:06: BR0/0:2 PPP: Phase is DOWN
kissane#

Obviously to fix this we just plug that cable in to the BRI interface.

No dialer list (layer 2):
For this one I just removed the dialer map statement.

kissane#sh isdn status
Global ISDN Switchtype = basic-ni
ISDN BR0/0 interface
   dsl 0, interface ISDN Switchtype = basic-ni
Layer 1 Status:
   ACTIVE
Layer 2 Status:
   TEI = 64,Ces = 1,SAPI = 0,State =MULTIPLE_FRAME_ESTABLISHED
   Spid Status:
      TEI 64,ces = 1,state = 5(init)  
      spid1 configured, spid1 sent, spid1 valid
      Endpoint ID Info: epsf = 0, usid = 70, tid = 1
      TEI Not Assigned, ces = 2, state = 1(terminal down)
      spid2 configured, spid2 NOT sent, spid2 NOT valid
Layer 3 Status:
   0 Active Layer 3 Call(s)
Activated dsl 0 CCBs = 0
The Free Channel Mask: 0x80000003
Total Allocated ISDN CCBs = 0

Layer 1 is active (normal), only 1 TEI on layer 2 (abnormal), and nothing on layer 3 (abnormal). We still need some more information.
kissane#sh controller bri
BRI unit 0:BRI unit 0 with U interface:
Layer 1 internal state is ACTIVATED
Layer 1 U interface is ACTIVATED.

Not much help here...we expected it because we already learned our layer 1 is fine. We get the bri is “good” with sh int bri0/0 and sh prot too. But show protocols tells us something is wrong. Let’s move to our layer 2 commands. This is where we should find our problems.

kissane#debug dialer packets
Dial on demand packets debugging is on
kissane#sh controller bri
05:31:22: BR0/0 DDR: ip (s=192.168.1.2, d=224.0.0.10), 60 bytes, outgoing interesting (ip PERMIT)
kissane#debug dialer packets
05:31:26: BR0/0 DDR: ip (s=192.168.1.2, d=224.0.0.10), 60 bytes, outgoing interesting (ip PERMIT)

Interesting traffic is being generated but the interface and protocol is not coming up. We need to look a bit further.

kissane#undebug all
All possible debugging has been turned off
kissane#debug isdn q921
ISDN Q921 packets debugging is on
kissane#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:.....
Success rate is 0 percent (0/5)

05:32:53688777932: ISDN BR0/0: RX <- RRp sapi = 0 tei = 64 nr = 5
05:32:12: ISDN BR0/0: TX -> RRf sapi = 0 tei = 64 nr = 4

Again, not much help here. The q.921 is being sent and received. Now let’s look at our PPP negotiation.

kissane#undebug all
All possible debugging has been turned off

kissane#debug ppp nego
PPP protocol negotiation debugging is on
kissane#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:.....
Success rate is 0 percent (0/5)
Aha! No PPP is being negotiated at all. We would expect some problem with PPP not being on the routers at all. Since we have several hints we just need to go back and double-check very closely our configurations for our layer 2 commands:

```
router(config-if)#dialer map ip 192.168.1.2 name routerB 5552000 (2)
router(config-if)#isdn spid1 51055512340001 5551234 (2)
router(config-if)#isdn spid2 51055512350001 5551235 (2)
router(config-if)#enc ppp (2)
router(config-if)#ppp authentication chap (2)
router(config)#username joe password cisco (2)
router(config)#ip host joe 192.168.1.1 (2)
```

To fix this we just add in the dialer map statement.  
**Incorrect dialer list (layer 2):**
For this I just changed the dialer map statement from so the number called would be incorrect (actually to itself). You will have the same outputs as if you did not even have a map.  
**Incorrect spid (layer 2/3):**
If you have more than one spid then you will have to change them all. The other spids are secondary and will be used in case the primary spid does not work. This one is easy to spot—the spid gets rejected.

```
kissane#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:.....
Success rate is 0 percent (0/5)
06:00:43: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0/0, TEI 64 changed to up
06:00:43: %ISDN-4-INVALID_SPID: Interface BR0/0, Spid1 was rejected
```

Just for sake of continuity let’s check our isdn status too.
```
kissane#sh isdn status
Global ISDN Switchtype = basic-ni
ISDN BRI0/0 interface
dsl 0, interface ISDN Switchtype = basic-ni
Layer 1 Status: ACTIVE
Layer 2 Status:
  TEI = 64, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
  Spid Status:
    TEI 64, ces = 1, state = 6(not initialized)
      spid1 configured, spid1 sent, spid1 NOT valid
    TEI Not Assigned, ces = 2, state = 1(terminal down)
      spid2 configured, spid2 NOT sent, spid2 NOT valid
Layer 3 Status: 0 Active Layer 3 Call(s)
Activated dsl 0 CCBs = 0
The Free Channel Mask: 0x80000003
Total Allocated ISDN CCBs = 0
```
Fairly simple to spot...then go back and put in the correct spid numbers.

No ppp (layer 2/3):
Here I just removed PPP from the BRI interface.

```bash
kissane#sh isdn status
Global ISDN Switchtype = basic-ni
ISDN BRI0/0 interface
    dsl 0, interface ISDN Switchtype = basic-ni
Layer 1 Status:
    ACTIVE
Layer 2 Status:
    TEI = 64,Ces = 1,SAPI = 0,State
    MULTIPLE_FRAME_ESTABLISHED
    TEI = 65,Ces = 2,SAPI = 0,
    State=MULTIPLE_FRAME_ESTABLISHED
Spid Status:
    TEI 64, ces = 1, state = 8(established)
    spid1 configured, spid1 sent, spid1 valid
    Endpoint ID Info: epsf = 0, usid = 70, tid = 1
    TEI 65, ces = 2, state = 8(established)
    spid2 configured, spid2 sent, spid2 valid
    Endpoint ID Info: epsf = 0, usid = 70, tid = 2
Layer 3 Status:
    0 Active Layer 3 Call(s)
    Activated dsl 0 CCBs = 0
    The Free Channel Mask: 0x80000003
    Total Allocated ISDN CCBs = 0
kissane#
```

Everything looks good until we get to our Layer 3 status. So we need to go through our commands. Our hunch would have us start at Layer 3.

```bash
kissane#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:....
Success rate is 0 percent (0/5)
kissane#
```

06:19:44: %ISDN-6-CONNECT:
Interface BRI0/0:1 is now connected to 5551234
06:19:44: ISDN BRI0/0: TX -> CONNECT_ACK pd = 8 callref = 0x35
So we see our line come up and go back down right away. Since we have “used” all of our layer 3 commands we need to go back and use some layer 2 commands.

kissane# debug isdn q921
ISDN Q921 packets debugging is on
kissane# ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:.....
Success rate is 0 percent (0/5)
kissane#
06:23:25: ISDN BR0/0: TX -> RRF sapi = 0  tei = 65  nr = 0
06:23:25: %ISDN-6-CONNECT: Interface BR10/0:1 is now connected to 5551234
06:23:124554092544: ISDN BR0/0: RX RRP sapi = 0  tei = 64 nr = 15
06:23:29: ISDN BR0/0: TX -> RRF sapi = 0  tei = 64 nr = 12
06:23:150323896320: ISDN BR0/0: RX RRP sapi = 0  tei = 65 nr = 0
06:23:35: ISDN BR0/0: TX -> RRF sapi = 0  tei = 65 nr = 0
06:23:167503724544: %ISDN-6-DISCONNECT: Interface BR10/0:1 disconnected from 5551234
06:23:39: ISDN BR0/0: TX INFO sapi = 0  tei = 64 ns = 15 nr = 12 i = 0x0
801364508028090
06:23:169652894924: ISDN BR0/0: RX RRR sapi = 0  tei = 64 nr = 16
06:23:39: ISDN BR0/0: TX INFO sapi = 0  tei = 64 ns = 16 nr = 13 i = 0x0
801365A
06:23:167503724544: ISDN BR0/0: RX RRR sapi = 0  tei = 64 nr = 17
06:23:40: %LINEPROTO-5-UPDOWN: Line protocol on Interface BR10/0:1, changed state to down
kissane#
06:23:193273569280: ISDN BR0/0: RX RRP sapi = 0  tei = 65 nr = 0
06:23:45: ISDN BR0/0: TX -> RRF sapi = 0  tei = 65 nr = 0
06:23:210453438464: ISDN BR0/0: RX RRP sapi = 0  tei = 64 nr = 17
06:23:49: ISDN BR0/0: TX -> RRF sapi = 0  tei = 64 nr = 13
06:23:236223242240: ISDN BR0/0: RX RRP sapi = 0  tei = 65 nr = 0
06:23:55: ISDN BR0/0: TX -> RRF sapi = 0  tei = 65 nr = 0

Same stuff with no real information so let's go to our other layer 2 command.

kissane#debug ppp nego
PPP protocol negotiation debugging is on
kissane#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:......
Success rate is 0 percent (0/5)
kissane#
06:26:48: %LINK-3-UPDOWN: Interface BR10/0:1, changed state to up
06:26:48: %ISDN-6-CONNECT: Interface BR10/0:1 is now connected to 5551234
06:26:49: %LINEPROTO-5-UPDOWN: Line protocol on Interface BR10/0:1, changed state to up
Wow! What happened to our PPP negotiation? Remember our LCP, IPCP, and CDPCP phases? Yeah...they are not here anymore. Obviously a PPP problem, even though we can hardly “see” a PPP problem so we need to go back and check our PPP encapsulation.

```
kissane#sh int bri0/0
BRI0/0 is up, line protocol is up (spoofing)
    Hardware is PQIICC BRI with U interface
    Internet address is 192.168.1.2/24
    MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
    Encapsulation H323, loopback not set
```

Got it! We need to change our encapsulation to PPP to get this thing to work.

**Incorrect username/password (layer 2/3):**
For this I changed the username but left the password “as-is.”

```
kissane#sh isdn status
Global ISDN Switchtype = basic-ni
ISDN BR10/0 interface
dsl 0, interface ISDN Switchtype = basic-ni
    Layer 1 Status:
        ACTIVE
    Layer 2 Status:
        TEI = 64, Ces = 1, SAPI = 0, State =MULTIPLE_FRAME_ESTABLISHED
        Spid Status:
            TEI 64, ces = 1, state = 5(init)
                spid1 configured, spid1 sent, spid1 valid
            Endpoint ID Info: epsf = 0, usid = 70, tid = 1
            TEI Not Assigned, ces = 2, state = 1(terminal down)
                spid2 configured, spid2 NOT sent, spid2 NOT valid
    Layer 3 Status:
        0 Active Layer 3 Call(s)
        Activated dsl 0 CCBs = 0
        The Free Channel Mask: 0x80000003
        Total Allocated ISDN CCBs = 0
```

From this we see problems at layer 2 because we only have one TEI and one valid spid. So we go through our layer 2 commands. I took out a lot of text on this one.

```
kissane#debug isdn q921
```
ISDN Q921 packets debugging is on
kissane# **ping 192.168.1.1**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:.....
Success rate is 0 percent (0/5)
kissane#

06:36:53: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
06:36:53: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551234
06:36:53: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to down
06:36:55: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
06:36:55: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551234
06:36:55: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to down
06:36:57: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
06:36:57: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551234
06:36:57: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to down
06:36:59: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
06:36:59: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to down

Here we can see us going up and down a lot again. With no real clues we turn to our other layer 2 command:

kissane# **debug ppp nego**
PPP protocol negotiation debugging is on
kissane# **ping 192.168.1.1**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:.....
Success rate is 0 percent (0/5)
kissane#

06:41:12: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
06:41:13: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551234
06:41:13: BR0/0:1 PPP: Treating connection as a callout
06:41:13: BR0/0:1 PPP: Phase is ESTABLISHING, Active Open
06:41:13: BR0/0:1 LCP: O CONFREQ [Closed] id 30 len 10
06:41:13: BR0/0:1 LCP: MagicNumber 0x03063CE1 (0x050603063CE1)
06:41:13: BR0/0:1 LCP: I CONFREQ [REQsent] id 68 len 15
06:41:13: BR0/0:1 LCP: AuthProto CHAP (0x0305C22305)
06:41:13: BR0/0:1 LCP: MagicNumber 0x0552EC62 (0x05060552EC62)
06:41:13: BR0/0:1 LCP: O CONPACK [REQsent] id 68 len 15
06:41:13: BR0/0:1 LCP: AuthProto CHAP (0x0305C22305)
06:41:13: BR0/0:1 LCP: MagicNumber 0x0552EC62 (0x05060552EC62)
06:41:13: BR0/0:1 LCP: I CONPACK [ACKsent] id 30 len 10
06:41:13: BR0/0:1 LCP: MagicNumber 0x03063CE1 (0x050603063CE1)
06:41:13: BR0/0:1 LCP: State is Open
A problem with our username (CHAP: Username smarts not found). We just go back and fix it.

**Missing username/password (layer 2/3):**

```
sh isdn status
Global ISDN Switchtype = basic-ni
ISDN BRI/0 interface
dsl 0, interface ISDN Switchtype = basic-ni
Layer 1 Status:
  ACTIVE
Layer 2 Status:
  Layer 2 NOT Activated
Spid Status:
  TEI Not Assigned, ces = 1, state = 1(terminal down)
  spid1 configured, spid1 NOT sent, spid1 NOT valid
  TEI Not Assigned, ces = 2, state = 1(terminal down)
  spid2 configured, spid2 NOT sent, spid2 NOT valid
Layer 3 Status:
  0 Active Layer 3 Call(s)
Activated dsl 0 CCBs = 0
The Free Channel Mask: 0x80000003
Total Allocated ISDN CCBs = 0
```

```
debug isdn q921
ISDN Q921 packets debugging is on
```

```
ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:.....
Success rate is 0 percent (0/5)
```

```
%ISDN-6-LAYER2UP: Layer 2 for Interface BR0/0, TEI 64 changed to up
%ISDN-6-LAYER2UP: Layer 2 for Interface BR0/0, TEI 65 changed to up
%LINK-3-UPDOWN: Interface BR0/0:1, changed state to up
%ISDN-6-CONNECT: Interface BR0/0:1 is now connected to 5551234
```
Here we are looking for a username that does not exist. So we go in and make a username “smarts” (like it is asking for) with a password (usually the enable secret password).

So what have I learned here?
So is your mind fried yet? In this lab we learned how the theory of how ISDN operates and how to troubleshoot it. Like our other troubleshooting labs we just follow the OSI model. Please keep in mind that ISDN problems may not always be ISDN-related. In one of the next labs you will be setting up ISDN between four routers. At first when I could not get it to work I ran through my ISDN troubleshooting steps. I also used ICMP debug commands. Still it wouldn’t work. Then I remembered an obscure reference to a PPP command (MLP-PPP multilink) that made it work. So, what seems to be causing the problem may not be it at all. Keep your mind open. If something bothers you too much, just walk away and go shoot some pool for an hour. You would be amazed how easy the answer comes after that. Plus its fun to go shoot pool too.
ISDN Configuration with Multiple Routers (ADTRAN)

Objective:
To learn how to set up a ISDN network with four routers, using BRI interfaces. In this lab you will be using an ADTRAN box for ISDN emulation.

Tools and Materials:
(4) routers
(4) straight-through cables
(1) workstation
(1) ADTRAN Atlas 550
(2) PC/workstations
(2) console cables

Lab Design:

<table>
<thead>
<tr>
<th>Router</th>
<th>Geoffrey</th>
<th>Osowski</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loop 0</td>
<td>192.168.110.1/24</td>
<td>192.168.120.1/24</td>
</tr>
<tr>
<td>BRI0</td>
<td>192.168.1.1/24</td>
<td>192.168.1.2/24</td>
</tr>
<tr>
<td>Adtran Port</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Phone number</td>
<td>555-1234</td>
<td>555-4000</td>
</tr>
<tr>
<td>SPID1</td>
<td>51055512340001</td>
<td>51055540000001</td>
</tr>
<tr>
<td>SPID2</td>
<td>51055512350001</td>
<td>51055540010001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Router</th>
<th>Wilson</th>
<th>Tang</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loop 0</td>
<td>192.168.130.1/24</td>
<td>192.168.140.1/24</td>
</tr>
<tr>
<td>BRI0</td>
<td>192.168.1.3/24</td>
<td>192.168.1.4/24</td>
</tr>
<tr>
<td>Adtran Port</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Phone number</td>
<td>555-7000</td>
<td>555-8000</td>
</tr>
<tr>
<td>SPID1</td>
<td>51055570000001</td>
<td>51055580000001</td>
</tr>
<tr>
<td>SPID2</td>
<td>51055570010001</td>
<td>51055580010001</td>
</tr>
</tbody>
</table>
**Step-By-Step Instructions:**

1. Configure the basics on each router. Don’t forget the loop back adapters. Set up and cable the lab as shown. Pick a routing protocol to use and advertise the networks.

2. Set the ISDN switch type on geoffrey and osowski routers:

   ```
   geoffrey(config)#isdn switch-type basic-ni
   geoffrey(config)#dialer-list 1 protocol ip permit
   osowski(config)#isdn switch-type basic-ni
   osowski(config)#dialer-list 1 protocol ip permit
   ```

3. Configure the ISDN interface on Geoffrey and Osowski We will start by getting two of them up first. You will be configuring the ip address, the isdn spid (service profile identifiers), and a dialer map (how to get from here to there).

   ```
   geoffrey(config)#int bri0/0
   geoffrey(config-if)#ip address 192.168.1.1 255.255.255.0
   geoffrey(config-if)#no shut
   geoffrey(config-if)#dialer-group 1
   geoffrey(config-if)#dialer map ip 192.168.1.2 name osowski 5554000
   geoffrey(config-if)#isdn spid1 51055512340001 5551234
   geoffrey(config-if)#isdn spid2 51055512350001 5551235
   geoffrey(config-if)#enc ppp
   geoffrey(config-if)#ppp authentication chap
   geoffrey(config)#username osowski password cisco
   geoffrey(config)#ip host osowski 192.168.1.2
   
   osowski(config)#int bri0/0
   osowski(config-if)#ip address 192.168.1.2 255.255.255.0
   osowski(config-if)#no shut
   osowski(config-if)#dialer-group 1
   osowski(config-if)#dialer map ip 192.168.1.1 name geoffrey 5551234
   osowski(config-if)#isdn spid1 51055540000001 5554000
   osowski(config-if)#isdn spid2 51055540010001 5554001
   osowski(config-if)#enc ppp
   osowski(config-if)#ppp authentication chap
   osowski(config)#username geoffrey password cisco
   osowski (config)#ip host geoffrey 192.168.1.1
   ```

4. Try to ping the loopback. You should not be able to see it. It should not have shown up in the ip routing table either. The ISDN line comes up, stays active, and then shuts off pretty quickly. It’s actually faster than the routing protocol (I used EIGRP). In order to make this work we need to set up some static routes between the two.

   ```
   geoffrey(config)#ip route 192.168.120.0 255.255.255.0 192.168.1.2
   ```

   This route basically is saying, “in order to get to the 192.168.120.0/24 network use the 192.168.1.2 interface.”
This route basically is saying, “in order to get to the 192.168.110.0/24 network use the 192.168.1.1 interface.” You should be able to ping and see all networks between geoffrey and osowski.

5. Set the ISDN switch type on each router of the other two routers. We will get the connection working between the other two and then add them all together:

```
wilson(config)#isdn switch-type basic-ni
dialer-list 1 protocol ip permit
```

```
tang(config)#isdn switch-type basic-ni
dialer-list 1 protocol ip permit
```

6. Configure the ISDN interface on “smarts.” We will start by getting two of them up first. You will be configuring the ip address, the isdn spid (service profile identifiers), and a dialer map (how to get from here to there).

```
wilson(config)#int bri0/0
ip address 192.168.1.3 255.255.255.0
no shut
wilson(config-if)#dialer-group 1
dialer map ip 192.168.1.4 name tang 5558000
wilson(config-if)#isdn spid1 51055570000001 5557000
wilson(config-if)#isdn spid2 51055570010001 5557001
wilson(config-if)#enc ppp
ppp authentication chap
username tang password cisco
wilson(config)#ip host tang 192.168.1.4
```

```
tang(config)#int bri0/0
ip address 192.168.1.4 255.255.255.0
no shut
dialer-group 1
dialer map ip 192.168.1.3 name wilson 5557000
isdn spid1 51055580000001 5558000
isdn spid2 51055580010001 5558001
enc ppp
ppp authentication chap
username wilson password cisco
ip host wilson 192.168.1.3
```

7. Try to ping the loopback. You should not be able to see it. It should not have shown up in the ip routing table either. The ISDN line comes up, stays active, and then shuts off pretty quickly. It’s actually faster than the routing protocol (I used EIGRP). In order to make this work we need to set up some static routes between the two.

```
wilson(config)#ip route 192.168.140.0 255.255.255.0 192.168.1.4
```
This route basically is saying, “in order to get to the 192.168.140.0/24 network use the 192.168.1.4 interface.”

tang(config)#ip route 192.168.130.0 255.255.255.0 192.168.1.3

This route basically is saying, “in order to get to the 192.168.130.0/24 network use the 192.168.1.3 interface.” You should be able to ping and see all networks between wilson and tang.

8. Now we need to get those ISDN routers talking to each other. We must enable multilinking on the BRI interfaces with PPP. This will allow us to use the other B-channel for communicating.

goeffrey(config-if)#dialer map ip 192.168.1.3 name wilson 5557000
goeffrey(config-if)#dialer map ip 192.168.1.4 name tang 5558000
goeffrey(config-if)#ppp multilink
goeffrey(config)#username wilson password cisco
goeffrey(config)#username tang password cisco
goeffrey(config)#ip host wilson 192.168.1.3
goeffrey(config)#ip host tang 192.168.1.4
goeffrey(config)#ip route 192.168.130.0 255.255.255.0 192.168.1.3
goeffrey(config)#ip route 192.168.140.0 255.255.255.0 192.168.1.4

osowski(config-if)#dialer map ip 192.168.1.3 name wilson 5557000
osowski(config-if)#dialer map ip 192.168.1.4 name tang 5558000
osowski(config-if)#ppp multilink
osowski(config)#username wilson password cisco
osowski(config)#username tang password cisco
osowski(config)#ip host wilson 192.168.1.3
osowski(config)#ip host tang 192.168.1.4
osowski(config)#ip route 192.168.130.0 255.255.255.0 192.168.1.3
osowski(config)#ip route 192.168.140.0 255.255.255.0 192.168.1.4

wilson(config-if)#dialer map ip 192.168.1.1 name geoffrey 5551234
wilson(config-if)#dialer map ip 192.168.1.2 name osowski 5554000
wilson(config-if)#ppp multilink
wilson(config)#username geoffrey password cisco
wilson(config)#username osowski password cisco
wilson(config)#ip host geoffrey 192.168.1.1
wilson(config)#ip host osowski 192.168.1.2
wilson(config)#ip route 192.168.120.0 255.255.255.0 192.168.1.2
wilson(config)#ip route 192.168.110.0 255.255.255.0 192.168.1.1
tang(config-if)#dialer map ip 192.168.1.1 name geoffrey 5551234

tang(config-if)#dialer map ip 192.168.1.2 name osowski 5554000

tang(config-if)#ppp multilink

tang(config)#username geoffrey password cisco

tang(config)#username osowski password cisco

tang(config)#ip host geoffrey 192.168.1.1

tang(config)#ip host osowski 192.168.1.2

tang(config)#ip route 192.168.120.0 255.255.255.0 192.168.1.2

tang(config)#ip route 192.168.110.0 255.255.255.0 192.168.1.1

9. Test by pinging each network. Notice how you can only do two routes at a time. This is because you only have two b-channels. You would need to add more to do more at once.

**Challenge Lab or Supplemental Activities:**
1. Repeat the lab using Class “B” addresses.
2. Repeat the lab using Class “A” addresses.
3. Try using PAP as an encapsulation for PPP. Try HDLC.

**So what have I learned here?**
In this lab you have learned how to set up an ISDN BRI connection using the ADTRAN between four routers. The key here was the PPP multilink command. You also got some good training on how to break a large project down into smaller steps.

---

**Guest Router Name Derivation**

Geoffrey Osowski and Wilson Tang were accountants who worked for CISCO (yes...them!). They were charged with computer-related crimes when, being accountants, they illegally issued almost $8,000,000 in CISCO stock to themselves. They each received 34 months in prison, 36 months of probation, and restitution of almost $8,000,000. The moral of the story is don’t try this at home!

---

**Take college courses on a new schedule--yours.**

Choose from over 400 online courses in dozens of majors, including our new bachelor’s degree program in Technology Management. St. Petersburg College’s eCampus delivers Internet convenience and the quality instruction that has made SPC an international leader in online education. Visit our web site for a full listing of courses and programs, along with support services like online registration, academic advising and financial aid.

[http://e.spcollege.edu](http://e.spcollege.edu)

St. Petersburg College’s eCampus. Your time. Your place. Your future.
Frame Relay with ISDN Backup

*Objective:*  
To learn how to configure a network that uses a frame relay connection for its main network traffic and an ISDN line for redundant backup using floating static routes.

*Tools and Materials:*  
(3) routers  
(1) MERGE ISDN emulator  
(2) workstations  
(2) straight-through cables  
(2) DCE-to-DTE cables

*Lab Diagram:*

![Lab Diagram](image-url)  

<table>
<thead>
<tr>
<th>Router</th>
<th>Patrice</th>
<th>Williams</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0/0</td>
<td>192.168.1.1/24</td>
<td>192.168.1.2/24</td>
</tr>
<tr>
<td>BRI0/0</td>
<td>10.0.0.1/8</td>
<td>10.0.0.2/8</td>
</tr>
<tr>
<td>Loop0</td>
<td>1.1.1.1/8</td>
<td>2.2.2.2/8</td>
</tr>
</tbody>
</table>

*Step-By-Step Instructions:*  
1. Make a router into a frame relay switch.  
2. Configure the basics on each router.  
3. Configure the loop back networks on each router.  
4. Configure the serial interfaces on each router for frame relay.  
5. Pick a routing protocol and advertise the networks.  
6. Check the frame relay network connectivity. Use ping and trace route.  
7. Disconnect/shut-down the frame relay connection.  
8. Configure the ISDN BRI0/0 interfaces on each router. Be certain to advertise the networks.  
9. Check the ISDN network connectivity. Use ping and trace route.  
10. Set up the ISDN circuit as a back up network using floating static routes. Set your ISDN connection to time-out after 20 seconds.
Reconnect the frame relay network. Trace the route to the loopback on williams. You should see:

```
patrice#traceroute 2.2.2.2
Type escape sequence to abort.
Tracing the route to 2.2.2.2

    1 williams (192.168.1.2) 32 msec *  28 msec

patrice#
```

Notice how we are getting to the loopback through the frame relay network. Now let’s unplug the frame line on patrice (to simulate a network failure) and re-trace route to our loopback again. We expect to see it route through the ISDN network (10.0.0.2) after it comes up.

```
patrice#traceroute 2.2.2.2
Type escape sequence to abort.
Tracing the route to 2.2.2.2

    1 10.0.0.2 16 msec 16 msec *

patrice#
```

Challenge Lab or Supplemental Activities:
1. Add PPP as an encapsulation. Use PAP or CHAP for authentication.
2. Switch address schemes to a pure Class “B” network.
3. There are many ways to perform backups: using dynamic floating routes, static routes with “backup” commands, etc. Try repeating this lab using different techniques.
4. Which works best and why? Did I use it here or mix it up?
5. Try this lab a bit differently: main ISDN with frame relay backup.
6. Try this lab a bit differently: main ISDN with ISDN backup.
7. Try this lab a bit differently: main frame Relay with frame relay backup.
8. If you have the equipment, then go back and forth between ADTRAN’s and MERGE boxes for ISDN and back and forth between routers and ADTRAN’s for frame relay.

So what have I learned here?
In this lab you have learned how to set up the bare minimum requirements for a Frame relay main connection with a backup ISDN BRI connection using the ADTRAN ISDN
simulators. This is one of the more common WAN configurations you will see in the “real-world” in small-to-medium sized businesses.

Guest Router Name Derivation

Patrice Williams was sent to prison in 2002 after she, and a partner (Makeebrah Turner), hacked into the Chase Financial Corporation. Apparently this dastardly duo stole credit card numbers and used them to purchase about $600,000 worth of merchandise on 68 different accounts. They also “distributed” some of those numbers to someone else in Georgia who, in turn, purchased about $100,000. The brain trust plea-bargained down to a one-year and a day prison term in return for a guilty plea.

The “Script Kiddie Cookbook” Available Mid-September 2004 at http://www.lulu.com/learningbydoing

This book title is based upon the Anarchist Cookbook that came out in the 1980’s. In that tremendously popular book it showed you how to build bombs at home. Similarly the Script Kiddie Cookbook will show you step by step hacking and hacking methodology. You will go to jail if you use this information improperly!!!

Learn the basics about computer security with this book. This book assumes you know nothing about security and starts there. All tools and software used are freeware. Other books will build upon the materials in this book. Learn about the nuts and bolts of SPAM, port scanning, Knoppix tools, cookies, and bring it all together with exercises in hacking at legal gaming sites.

GET THE BOOK THE GOVERNMENT DOES NOT WANT YOU TO OWN!!!
**Part 4 Command Review**

**Objective:**
To list all commands utilized in Part 4 of this textbook.

**Step-by-Step Instructions:**
1. For each of the commands give a description of the command, the prompt for configuration, and any abbreviations for that command.

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Command</th>
<th>Shortcut</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prompt</td>
<td>Command</td>
<td>Shortcut</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>