Part 1:
Networking Fundamentals

Look for the newest version of this manual on Lulu.com on August 1st. The new manual has Win2K labs and uses IOS 12.0-12.3 for the labs. There are also some security labs within that book. I have also written a computer security fundamentals book called

"The Script Kiddie Cookbook"

that also will be available from Lulu in mid-August. Thanks and I hope you enjoy the book. Please send me any edits too. Thanks!

Searching CISCO for CCNA Test information

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

Tools and Materials:
(1) PC with Internet access

Step-By-Step Instructions:

1. Open a browser window.
2. Navigate to www.cisco.com. You should see:

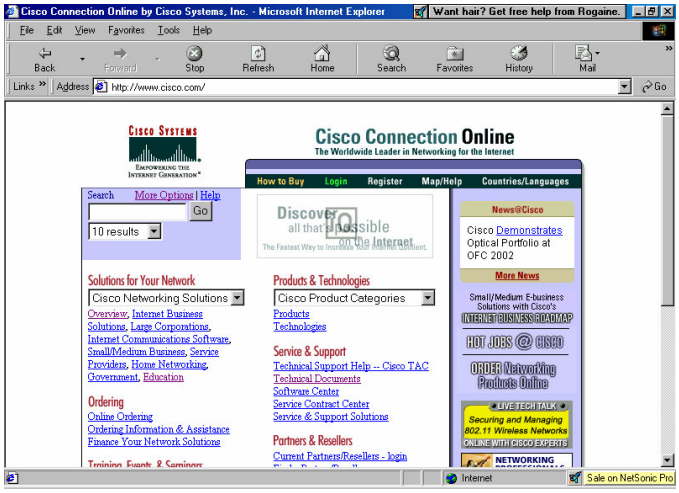


Figure 1—The main CISCO webpage.

3. Next, scroll down. On the left hand side you should see a link under the "Training, Events, & Seminars" heading called "Training/Certifications." You should see:

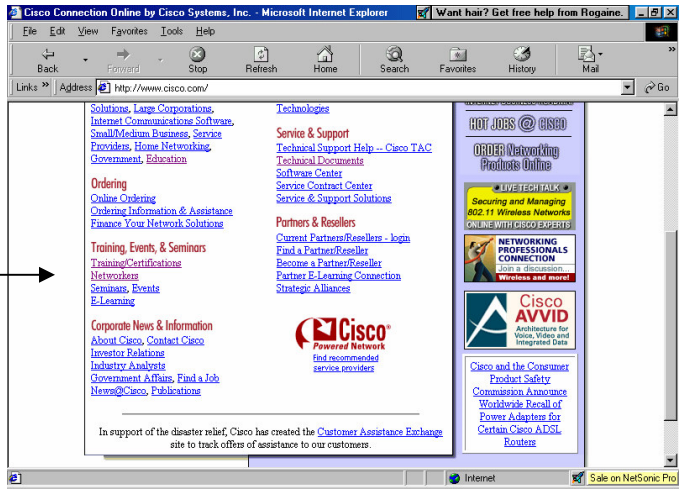


Figure 2—Scroll down to the Training heading and look for "certifications."

4. Click on the link for “certifications.” The page you should see next is:

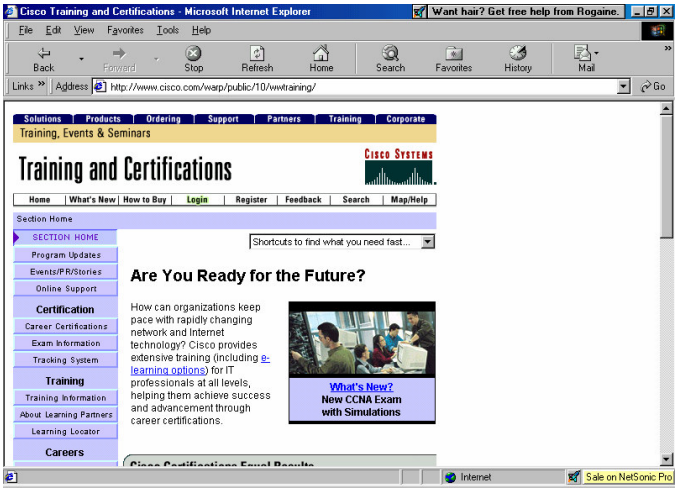


Figure 3—Training and Certifications main page.

5. Then scroll down again to the “current exams and outlines” link. It will take you to the page for current exams and outlines (isn’t that nice?). You should see:

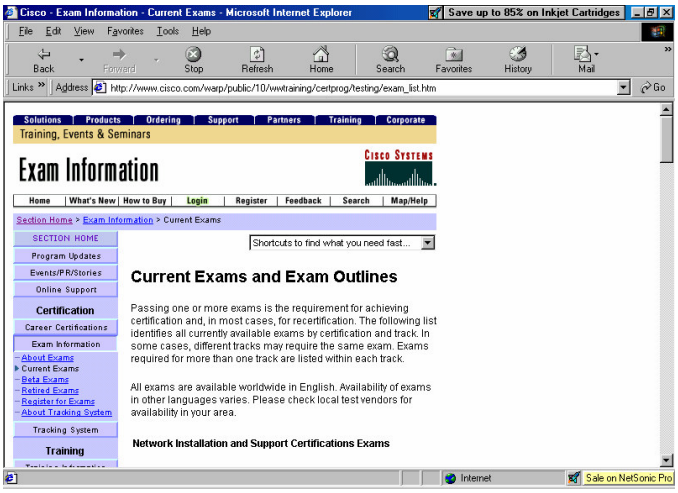


Figure 4—Current exams and outlines page.

6. Once again, scroll down until you find the CCNA 640-607 exam. You should see (figure 5 on next page).

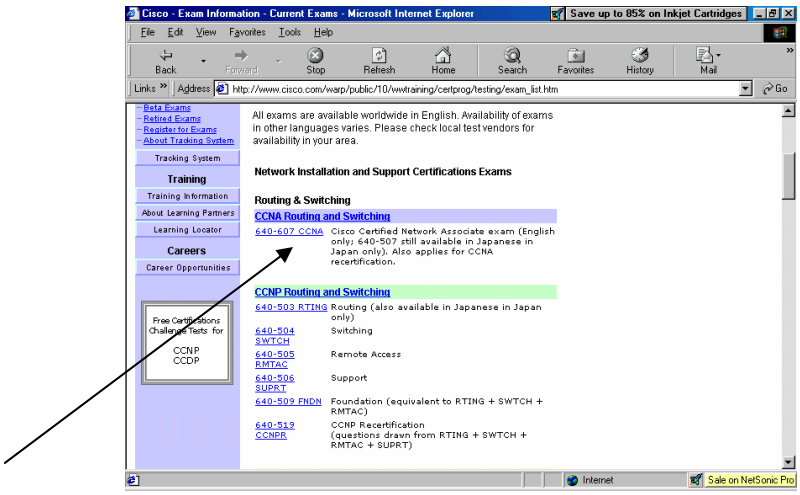


Figure 5—Scroll down to CCNA exam.

7. Click on the link “640-607” and another window should open. You should see:

Figure 6—CCNA test main page.

8. Again, scroll down a bit and you should see some available options (hyperlinks). You should see (figure 7 on next page):

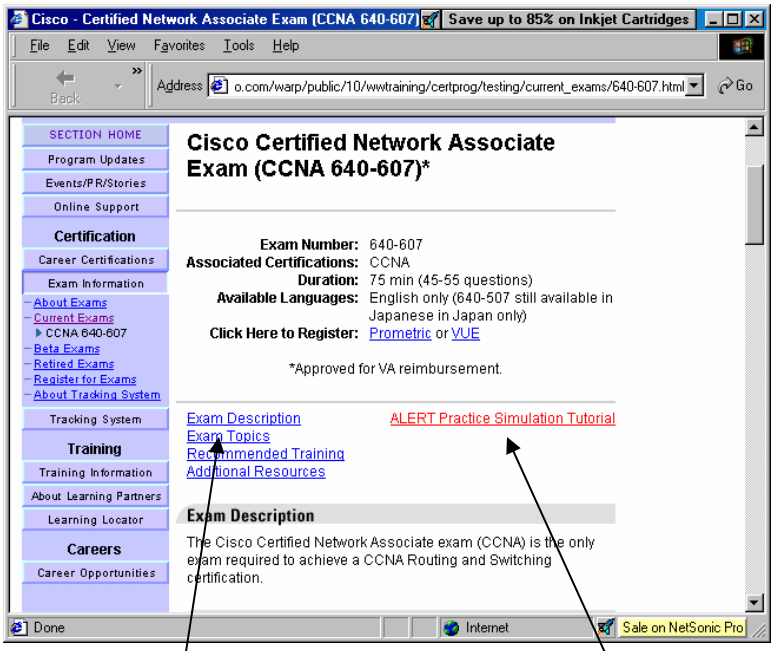


Figure 7—CCNA main page.

9. The simulation tool link will open another page. The instructions will read "Effective March 12, 2002, in addition to multiple choice and fill-in response questions, Cisco Career Certifications exams may include performance simulation questions. Performance simulations are test problems that approximate a real-life environment on a candidate's computer screen. Candidates will be presented with a real-life scenario and a networking topology to address specific tasks through appropriate responses. The responses that a candidate enters must be the same as those one would expect in a real-life networking situation. Prior to taking the CCNA 640-607 exam (the first exam to include simulations), candidates should become familiar with the exam simulation tool. Such practice will allow candidates to focus their exam-taking effort on the exam questions rather than how to correctly use the tool. To learn more about the simulation tool, use the following graphic tutorial." You may want to spend some time going through the instructions. Figure out if short-cut keystrokes are allowed or not.
10. Also look at the description of exam topics. Use this to guide your studies as you progress through your CCNA training.

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.

DOS Lab

Objective:
This lab is designed to become familiar with basic DOS commands and utilities on Windows Operating Systems.

Tools and Materials:
(1) Computer with Windows 95/98.
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.

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 (see figure 1). A black screen or a window with a black screen should appear.

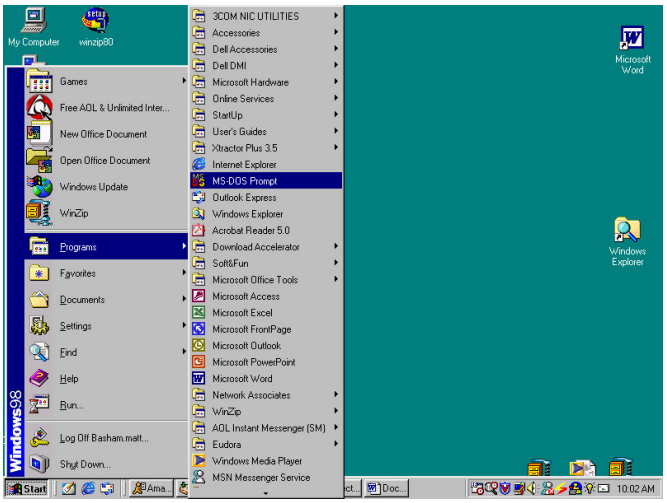


Figure 1—Starting MS-DOS from the task bar.

d. If you want to be a show-off then click on “Start” then “Run.” The pop-up window should see something like figure 2 (without the Windows menu on the side).

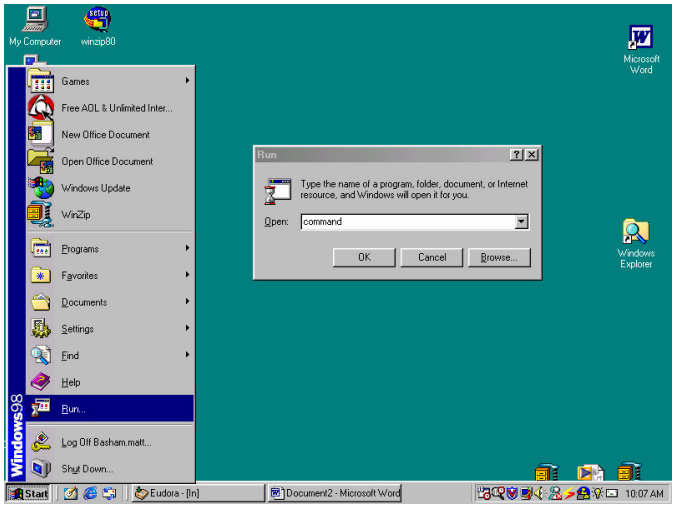


Figure 2—Starting the “run” utility.

e. Type in “command” (without quote marks) and the black screen DOS window should appear (see figure 3).

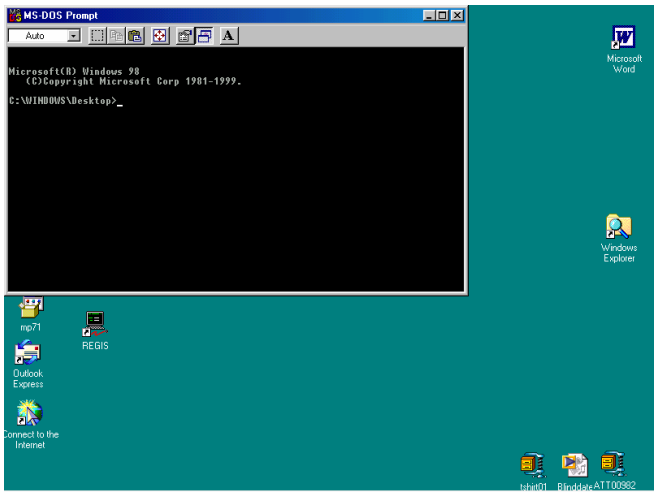


Figure 3—The MS-DOS prompt window.

f. To make the window fill your entire screen press the button with the arrows in all direction (like a compass pointer). If you want to get the window back then press Alt+Enter. If you want to leave the MS-DOS prompt session open in a full window, but you want to copy something from Windows you can use Alt+tab to “shuttle” between open programs. This is the hallmark of “switching between windows.”

g. 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. You should see a window like figure 4.

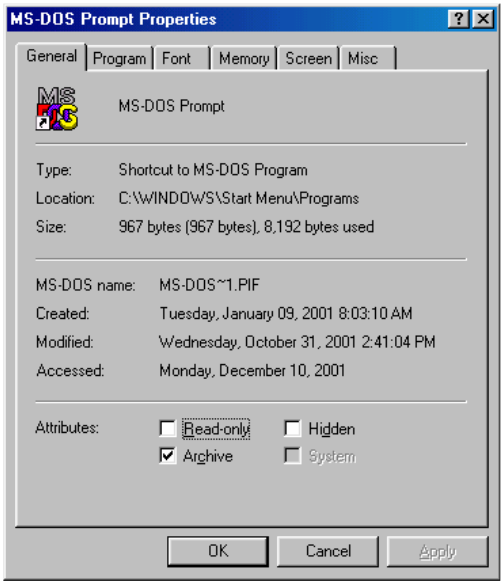


Figure 4—MS-DOS properties.

h. Ok…now you can really start showing off…click on the “misc” tab. You will see something like figure 5.

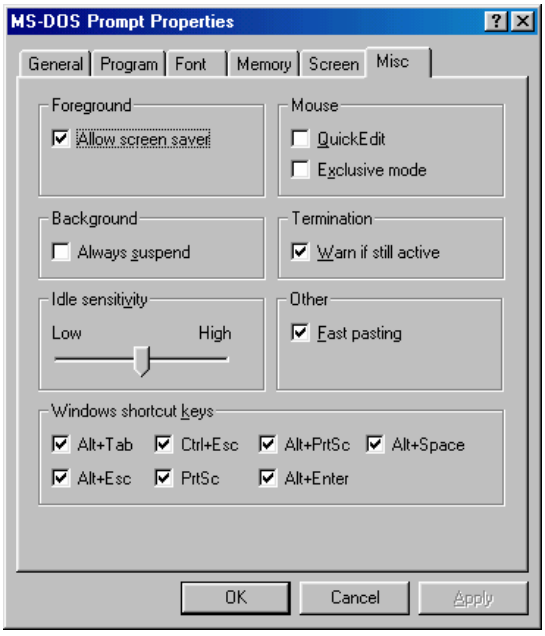


Figure 5—MS-DOS prompt miscellaneous settings.

i. 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.

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 DOS. Using the directory "tree" structure shown on the next page (figure 6) we could write down the paths for certain files. For example the complete path to the album.zip file would become:

C:\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):

autoexec.bat _____

letter.doc_____

winzip.exe _____

word.exe _____

command.com _____

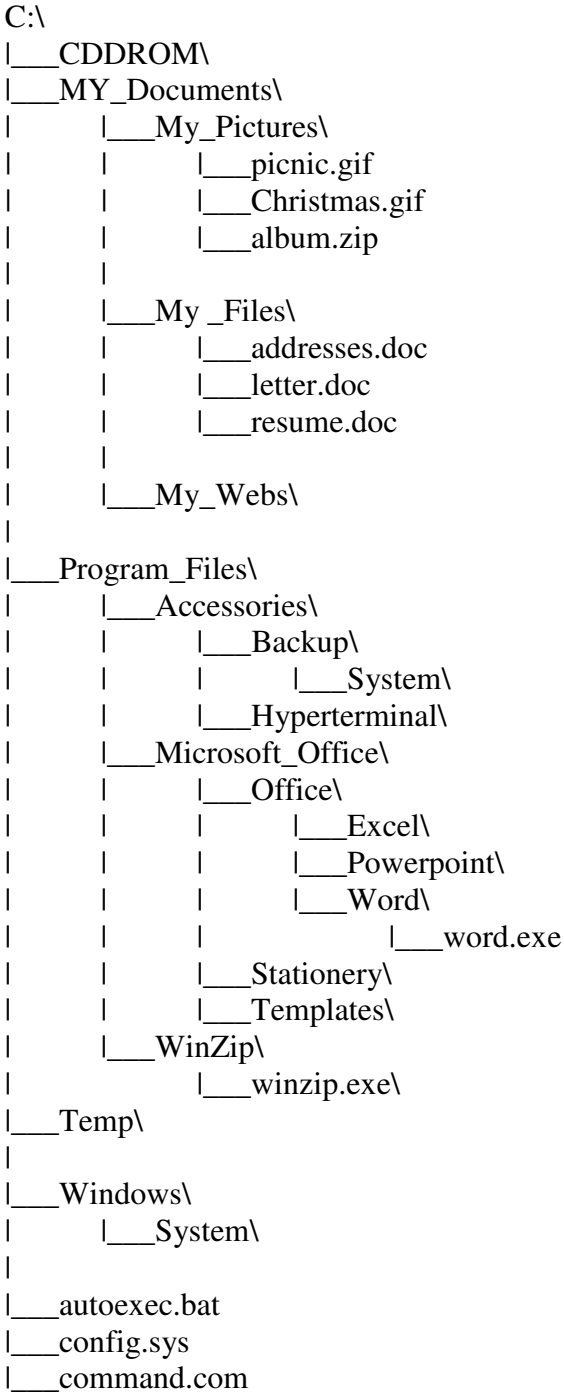


Figure 6—Hypothetical directory tree.

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. (This is probably gonna take a while…)

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." The first command you will learn allows you to change directories. You do this by typing "CD" or "CHDIR" 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:\Windows\desktop>" If we wanted to navigate to the my documents file directory (C:\windows\my documents) we could switch to it in one of several ways…(1) type "CD C:\windows\mydocu~1" or (2) type "CD.." this will change you from the directory "desktop" prompt to the "C:\windows" prompt. Then type "CD mydocu~1" to change to the my documents directory. Please note that you can use the dot-dot to go back one level with the CD command. If your prompt was C:\windows\system\oobe you could type "CD …." to return to the root. Two dots for one level and one dot for every level thereafter. This is called "going up the tree." Its opposite, "going down the tree," requires you typing in each directory or subdirectory. For example, to go from "C:" to "C:\windows\system\oobe" you could type "CD: C:\ windows\system\oobe" or from the root prompt type "CD windows" hit enter then type "CD system" hit enter, then type "CD oobe." There are literally many different ways to do the same thing.

So using figure 6 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:\windows to get to the root prompt _____

From letter.doc back up two levels _____

From winzip folder to system folder _____

From word.exe to temp folder _____

3. *Finding Files in DOS*. DOS incorporates a searching mechanism. To find a specific file you use a directory statement, then the file name. For example, if we were looking for the c:\autoexec.bat file we would (1) open the MS-DOS prompt window, (2) switch to the root directory, and (3) use a directory statement to find the file. (See script 1 for syntax). You must be in the correct folder to find the file otherwise you will be unsuccessful.

```
C:\windows>
C:\windows> CD..
C:\dir autoexec.bat

Autoexec.bat   338      12-02-2001     7:52a   autoexec.bat
```

Script 1—finding a specific file

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 butoexec.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"

What batch files (.bat) are found at the root, the windows, and windows\system folders on your computer?

_____
_____
_____
_____

What command files (.com) are found at the root, the windows, and windows\system folders on your computer?

_____
_____
_____
_____

What executable files (.exe) are found at the root, the windows, and windows\system folders on your computer?

_____
_____
_____
_____

What system files (*.sys) are found at the root, the windows, and windows\system folders on your computer?

_____
_____
_____
_____

What are some of the other files found on your root?

_____
_____
_____

4. *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 _____
cd _____
chcp _____
cls _____
copy _____
ctty _____
date _____
del _____
echo _____
exit _____
for _____
goto _____
if _____
mkdir _____
path _____
pause _____
prompt _____
rem _____
ren _____
rmdir _____
set _____
shift _____
time _____
type _____
ver _____
verify _____
vol _____

External commands: files with *.com or *.exe extensions. These are not built into the operating system and can vary between operating system versions.

attrib _____
chkdsk _____
command _____
deltree _____
diskcopy _____
fc _____
fdisk _____

find _____
format _____
keyb _____
label _____
mode _____
more _____
nlsfunc _____
setver _____
sort _____
subst _____
sys _____
xcopy _____

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

File name	Contents
Dave.txt	This is Dave's text file…so keep out!
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!

6. *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. See script 2.

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

Script 2—Copying and renaming a file to make a backup.

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.

7. *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. The ones in **bold** will be used a lot in up-coming labs.

ARP.EXE _____
CDPLAYER.EXE _____
CLIPBRD.EXE _____

CLSPACK.EXE _____
CLEANMGR.EXE _____
CONTROL.EXE _____
CVT1.EXE _____
DEFRAG.EXE _____
DIALER.EXE _____
DRVSPACE.EXE _____
EDIT.EXE _____
EXPLORER.EXE _____
FREECELL.EXE _____
FTP.EXE _____
IPCONFIG.EXE _____
JVIEW.EXE _____
MPLAYER.EXE _____
MSHEARTS.EXE _____
NBTSTAT.EXE _____
NET.EXE _____
NETSTAT.EXE _____
NETWATCH.EXE _____
NOTEPAD.EXE _____
PACKAGER.EXE _____
PBRUSH.EXE _____
PING.EXE _____
PROGMAN.EXE _____
QFECHECK.EXE _____
REGEDIT.EXE _____
ROUTE.EXE _____
RSRCMTR.EXE _____
SCANDSKW.EXE _____
SCANREGW.EXE _____
SETDEBUG.EXE _____
SETVER.EXE _____
SIGVERIF.EXE _____
SMARTDRV.EXE _____
SNDREC32.EXE _____
SNDVOL32.EXE _____
SOL.EXE _____
SYSMON.EXE _____
TASKMAN.EXE _____
TELNET.EXE _____
TOUR98.EXE _____
TRACERT.EXE _____
TUNEUP.EXE _____
UPWIZUN.EXE _____
VCMUI.EXE _____
WELCOME.EXE _____

WINREP.EXE _____
WINFILE.EXE _____
WINHELP.EXE _____
WINHLP32.EXE _____
WINIPCFG.EXE _____
WINMINE.EXE _____
WINPOPUP.EXE _____
WINVER.EXE _____
WJVIEW.EXE _____
WRITE.EXE _____
WUPDMGR.EXE _____

8. 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 _____

NET.EXE _____

PING.EXE _____

ROUTE.EXE _____

NETSTAT.EXE _____
_____
_____
_____
_____
_____
_____
_____

IPCONFIG.EXE _____
_____
_____
_____
_____
_____
_____
_____

NBTSTAT.EXE _____
_____
_____
_____
_____
_____

9. *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).
10. *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!

```
REM    ************************************************************
REM    *              Batch file to change names of those four text files          *
REM    ************************************************************
REM
REM    By Matthew J. Basham, 02/21/2002
REM    Copyright 2002
REM    May not be reproduced without explicit written permission of the
REM    author.
ECHO
ECHO    Let's start those little buggers up!
ECHO
Pause
copy c:\temp\dave.txt c:\temp\dave.bak
```

```
pause
copy c:\temp\matt.txt c:\temp\matt.bak
pause
copy c:\temp\scott.txt c:\temp\scott.bak
pause
copy c:\temp\tim.txt c:\temp\tim.bak
pause
ECHO  ALL DONE!
```

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 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.

Windows Utilities Lab

Objective:
To become better aware of utilities included with Windows 95/98 Operating systems.

Tools and Materials:
(1) computer with Win 95/98
paper and pencil
Win 95/98 CD may be needed

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, 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:
1. Which programs did you find that may be useful to you as a network administrator?
2. If you had two different computers, one with 95 and one with 98, what are the differences between the available programs?
3. Try a Windows 2000 or XP using the same techniques.
4. Make a chart comparing the "evolution" of programs in each operating system over time. 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 95/98 Utilities

KRNL386.exe	Never, never, never ever delete. This is the "glue" for the windows operating system. Get rid of this and you have got trouble.
IPCONFIG.exe	Shows IP, MAC, and gateway addresses of your workstation
WINREP.exe	A "mini-help desk" type program. Good for gathering information about your workstation.
NETWATCH.exe	Monitors access to your workstations and servers
WUPDMGR.exe	Takes you (conveniently?) to the Microsoft website for software updates. No fumbling around that old website trying to find the right spot.
QFECHECK.exe	When you log into the Microsoft site this program runs and reports to Microsoft to make sure all Microsoft software is registered with Microsoft including license numbers.
WINPOPUP.exe	A private messaging utility.
ARP.exe	Shows address resolution protocol table of your workstation.
FTP.exe	File transfer program.
PING.exe	Troubleshooting program. Lots of options. This can be used to generate network traffic for testing too. A must see!
ROUTE.exe	Adds a gateway to your computer from the DOS prompt.
TRACERT.exe	Shows routes between your computer and a destination. A good troubleshooting tool.
TELNET.exe	Opens terminal emulation sessions between networking devices. A must see!
NBTSTAT.exe	Displays protocol statistics and current TCP/IP connections using NETBIOS over TCP/IP.
NETSTAT.exe	Shows active connections to your workstation. Lets you do remote administration to other workstations.
NET.exe	Shows who can share what resources on your network.
EMM386.exe	Shows expanded memory services available. Never, never, ever delete.
*.pwl	Password list files. If these disappear then you will be prompted to input a new password.
SYSEDIT.exe	System file editor and configuration utility. Good for looking at the most important system files quickly in windows.
REGEDIT.exe	Utility for editing the registry. If you don't know what you are doing, then I would advise you to stay out of this. Always backup the registry before making any registry changes.

Installing a NIC: Hardware

Objectives:
To be able to install a network interface card (NIC) into a personal computer (PC). In the next lab you will complete the installation of the NIC by performing the software installation.

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" covers.

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.
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.

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.

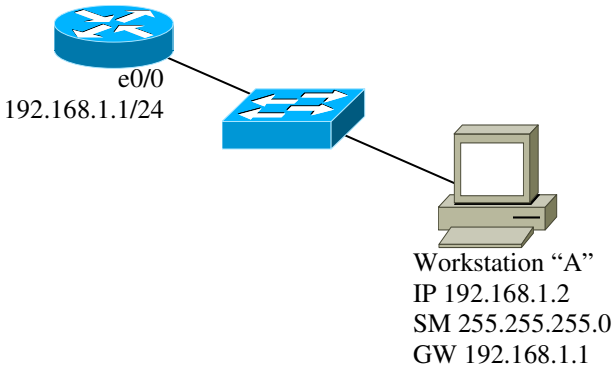
So What Have I Learned Here?
You have learned how to physically install a NIC. In the next lab you will be installing the software portion of a NIC installation.

Changing TCP/IP Settings on Your Computer

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

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 Network Neighborhood icon on the desktop using a right-click. Then click on "properties." You should see the network window:

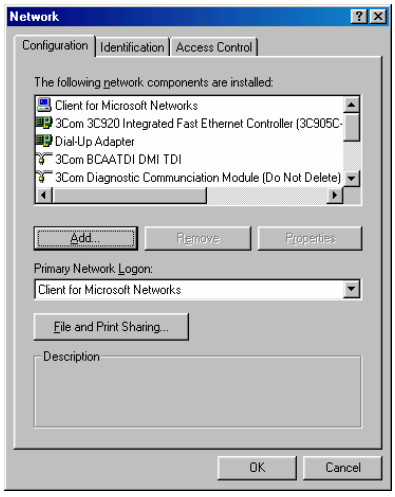


Figure 1—Network window.

2. Then scroll down to the TCP/IP configuration for your NIC. On my computer I picked this one (highlighted):

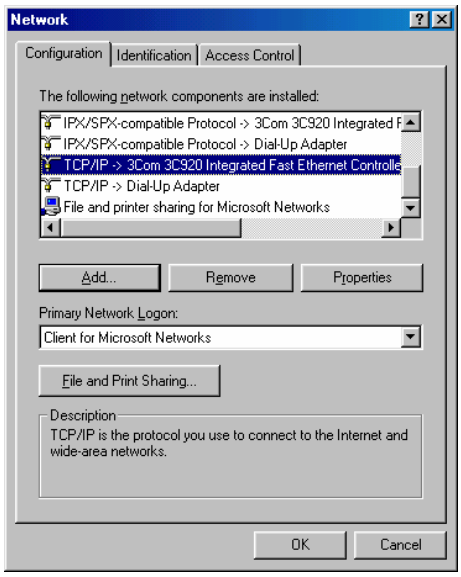


Figure 2—Finding the TCP/IP configuration for the NIC.

3. Double-click it or highlight it and select properties. You should see another pop up window like this:

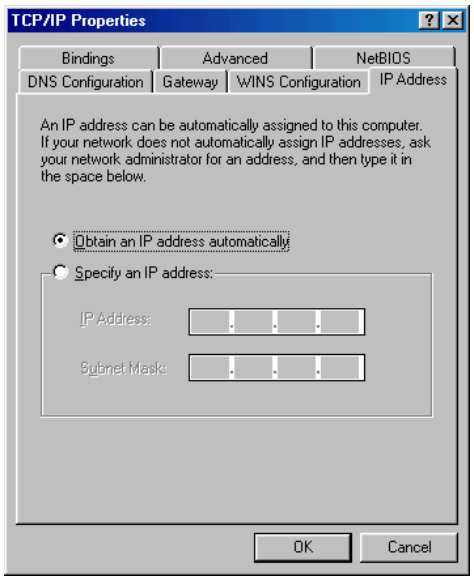


Figure 3—TCP/IP Properties pop up window.

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:

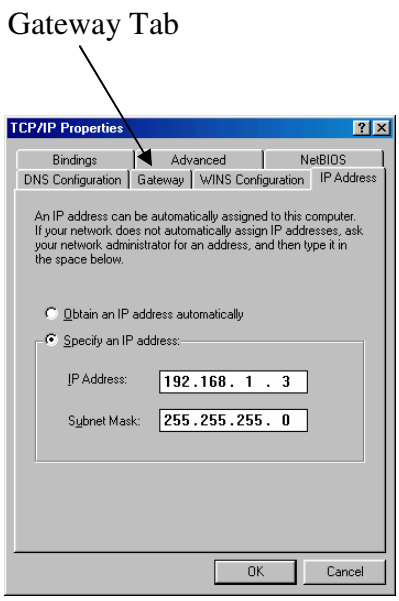


Figure 4—Putting in an IP address and mask.

5. Next we need to switch to the gateway tab (see figure 4) and put in the gateway address. We would type it in and click "add." Your pop up window will look like this:

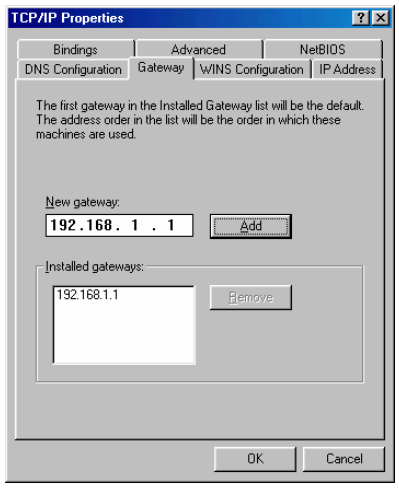


Figure 5—Adding a gateway.

6. Almost done. To finish it up we click on "ok" on the TCP/IP Properties window, and then "OK" in the network window. 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" or "WINIPCFG.EXE."

Supplemental Lab or Challenge Activity:

1. Try to find out about all of those other tabs and settings in the network and TCP/IP Properties windows.
2. 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. Better learn it good now.

Paper Lab: ICONS for Computer Diagrams

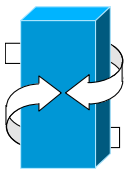
Objective:
To learn about ICONS used in CISCO drawings and for what each represents.

Tools and Materials:
None.

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

Router—Layer 3 device. Models include 2500 and 2600 series for access layer.

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.



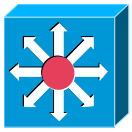
Gateway—Device that acts as a "gateway" to the network or Internet.

Bridge—Old school layer 2 device not used too much anymore.

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).

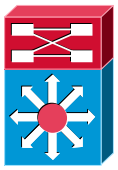
100BaseT hub—Not used too much anymore since switches cost about the same.

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 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.

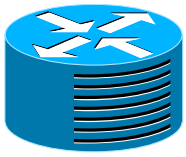
TAG router switch—uses TAG’s to forward packets. Does routing functions too.

Broadband router—Router for broadband connections.

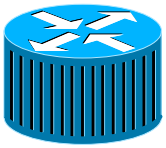
CISCO Net Ranger—CISCO security device.

ATM Router—Router for ATM. 8500 series routers.

CISCO 7505 Router—distribution/core layer router.

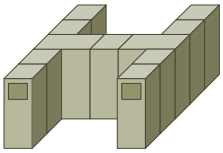


CISCO 7507 Router—distribution/core layer router.



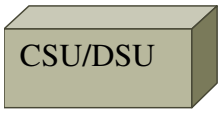
CISCO 7500 (7513) Router—distribution/core layer router.

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



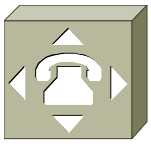
MAIN Frame—oh…that's the old school stuff.

IBM A/S 400—ditto, although these are still found in accounting departments.



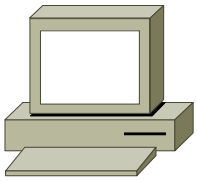
CSU/DSU—Channel Service Unit/Data Service Unit…from the "WAN cloud" into this and then into your router.

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

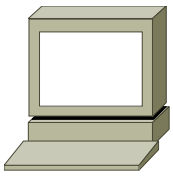


Small PBX—mini telephone company service that goes in your company. If you dial a "9" to get an outside line, then you have a PBX.

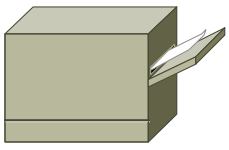
The "Cloud"—This is where all WAN starts and ends. We use this in many instances…to represent the Internet, a frame relay cloud, an ISDN cloud, a POTS cloud, etc.



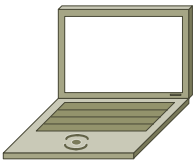
PC/Workstation—I really should not have to explain this one.



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.

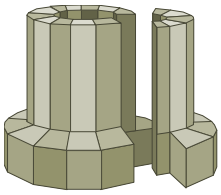


Printer—I really should not have to explain this one either. So there.

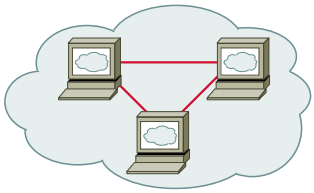


Laptop—ditto.

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.

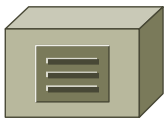


Supercomputer—See Nasa, Berkely, MIT, etc. Kind of like the W.O.P.R. in Wargames.



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.

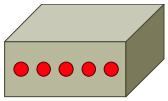
Token Ring—ICON to represent a layer 2 token ring topology.

FDDI—Icon to represent a layer 2 FDDI topology.

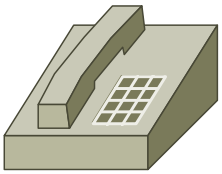
Ethernet—Icon to represent a layer 1/2 Ethernet cable.

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

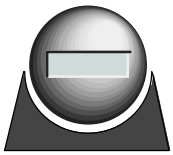
Circuit Switched Serial—ditto.



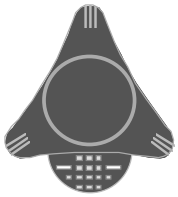
Modem—Modulator/Demodulator. Translates analog into digital signals.



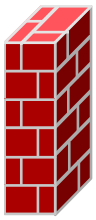
Phone—I should not have to explain this.



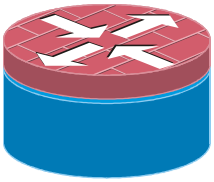
PC Camera—Itty bitty camera for your computer.



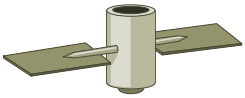
PolyComm phone—Speaker phone commonly used for conference calls.



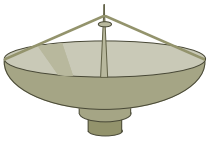
Firewall—Network Address Translation device. Great when they work properly. There is a big future in computer security…especially if you can get these things to work right.



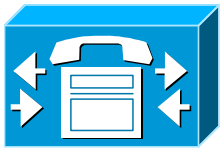
Router with firewall—Just what it sounds like…a router with the addition of firewall commands.



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.



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.

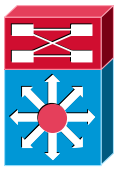
You will see some of these used in the drawings in this book. I put the other ones in here because I see them in articles and books.

More ICONs on the web!
http://www.cisco.com/warp/public/784/packet/icons/
http://www.cisco.com/warp/public/503/2.html

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?

_____

_____

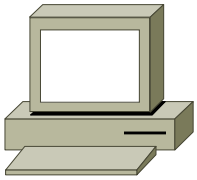


_____

_____

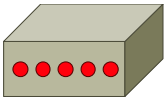
_____

_____



_____

_____



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." In fact, 66 and 110 punch down blocks are wired in this fashion:

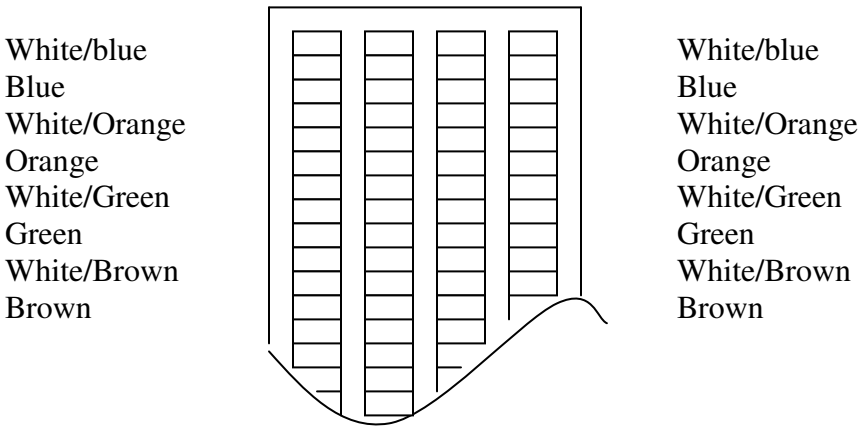


Figure 1—punch down block.

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.

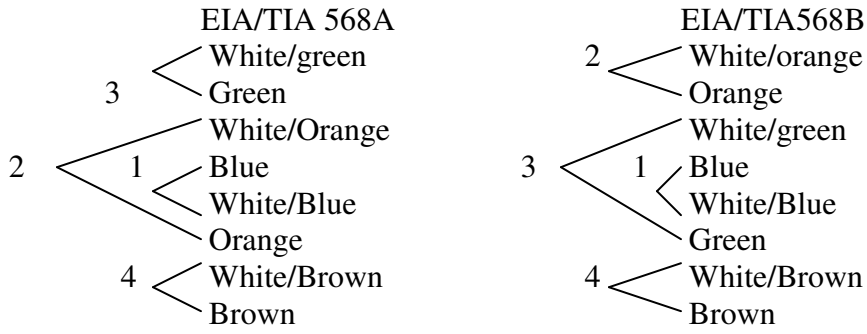
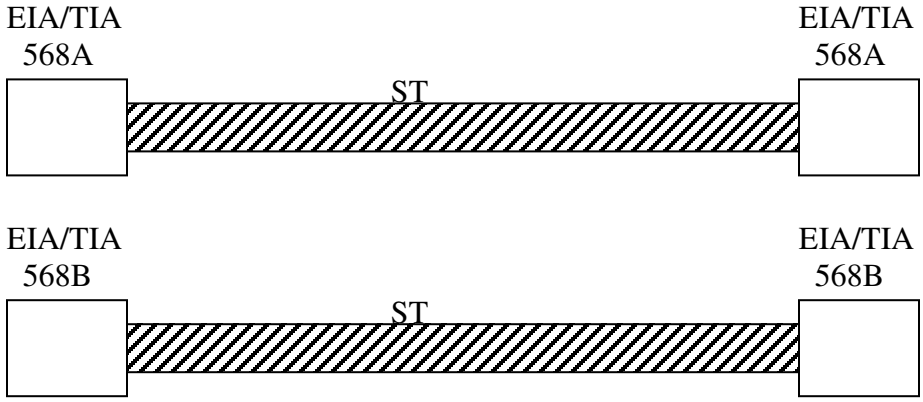
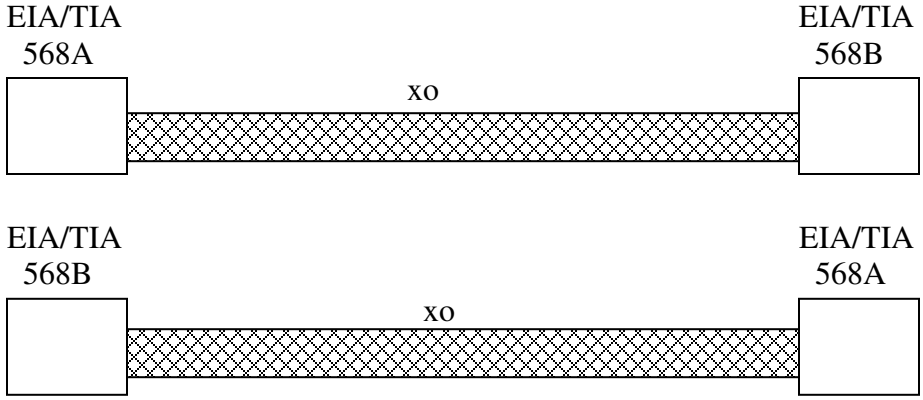


Figure 3—EIA/TIA 568A and B wiring patterns.

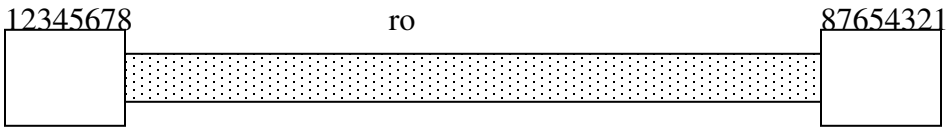
Straight Through (ST): Used for connecting **dis-similar devices** (workstations to hubs, switches to routers, hubs to switches, etc.). The cables are wired with the same wiring pattern on each end.



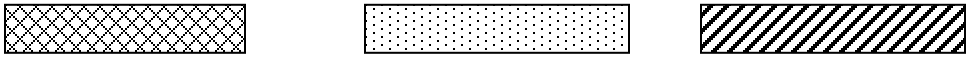
Crossover (xo): Used for connecting **similar devices** (workstations to workstations, switches to switches, hubs to hubs, etc). The cables are wired with pairs 2 and 3 "crossing over" from one end to the other (see also figure 3).



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.

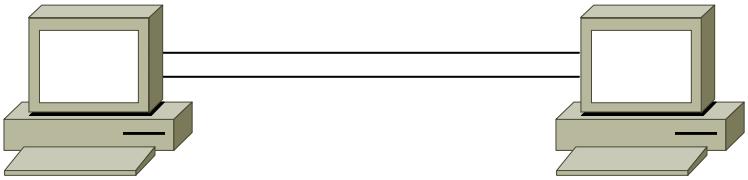


In the following diagrams indicate which type of cable is used, label each cable, apply the appropriate pattern in the drawing, and indicate which port or connection would be used at the each end of the cable.

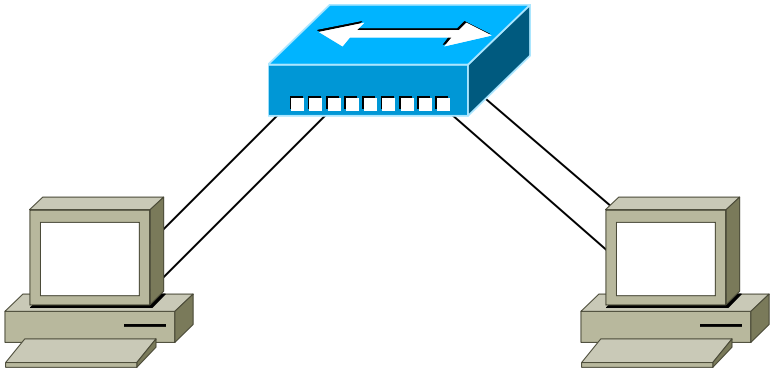


Crossover (xo) | Rollover (ro) | Straight-through (ST)

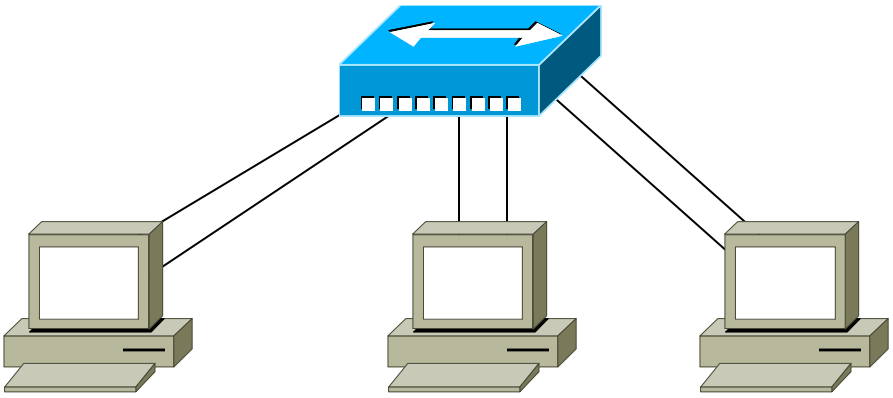
Peer-to-Peer Cabling



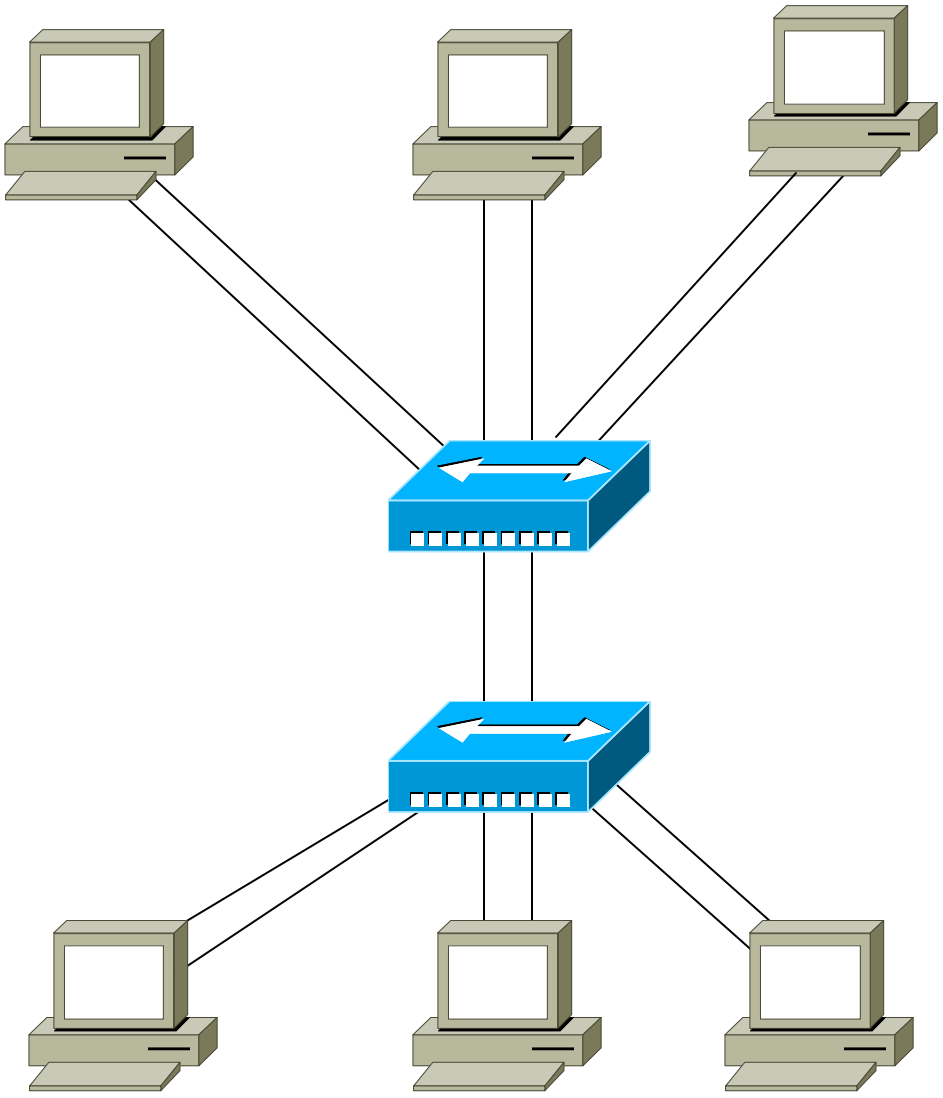
Two workstations and a hub



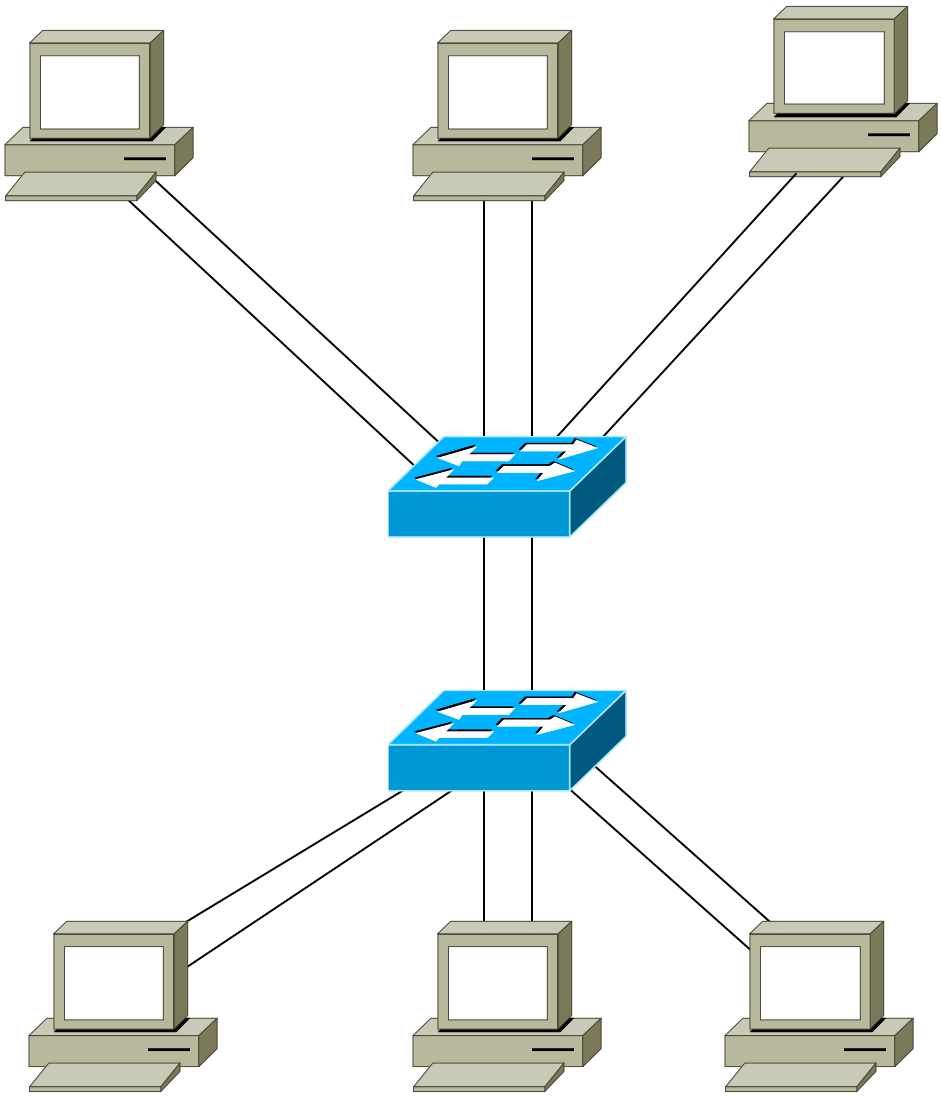
Three workstations and a hub



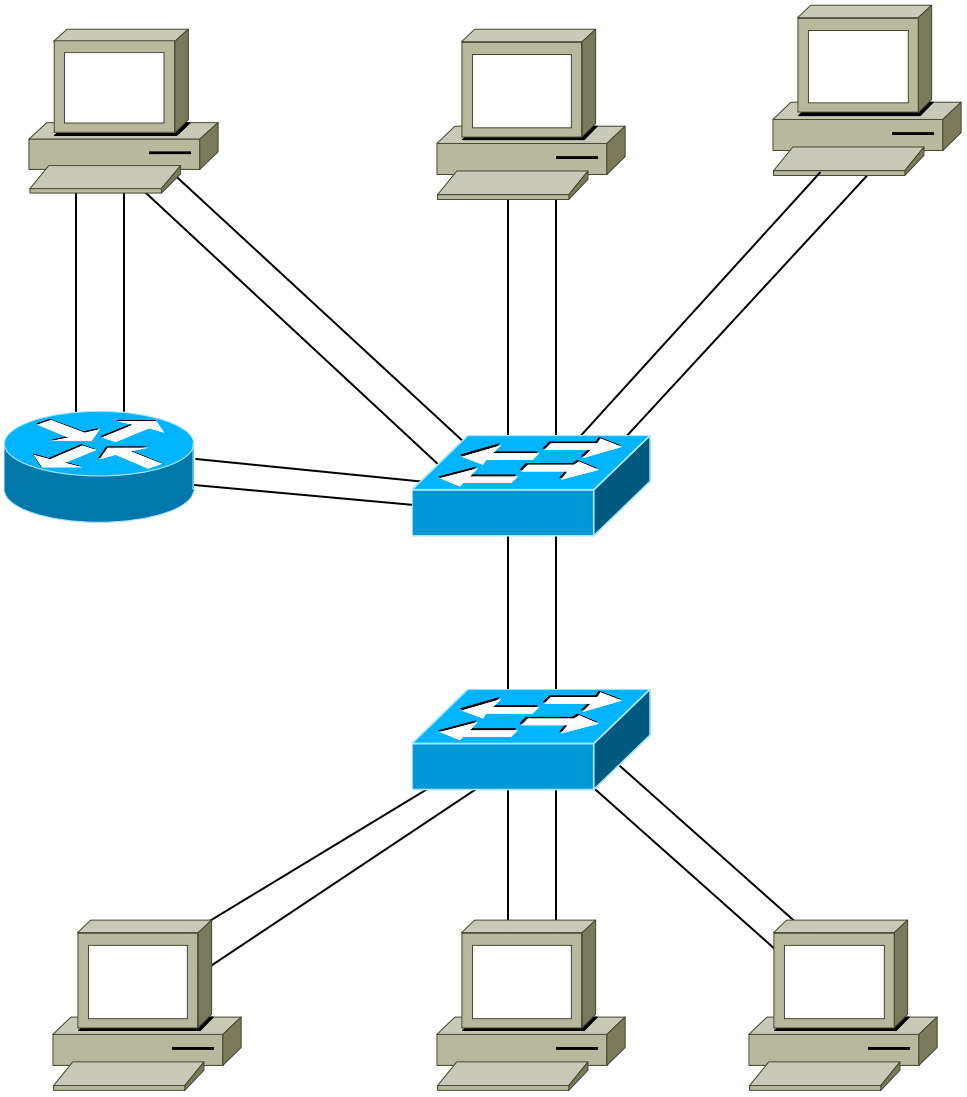
Six workstations (3 to a hub) and two hubs



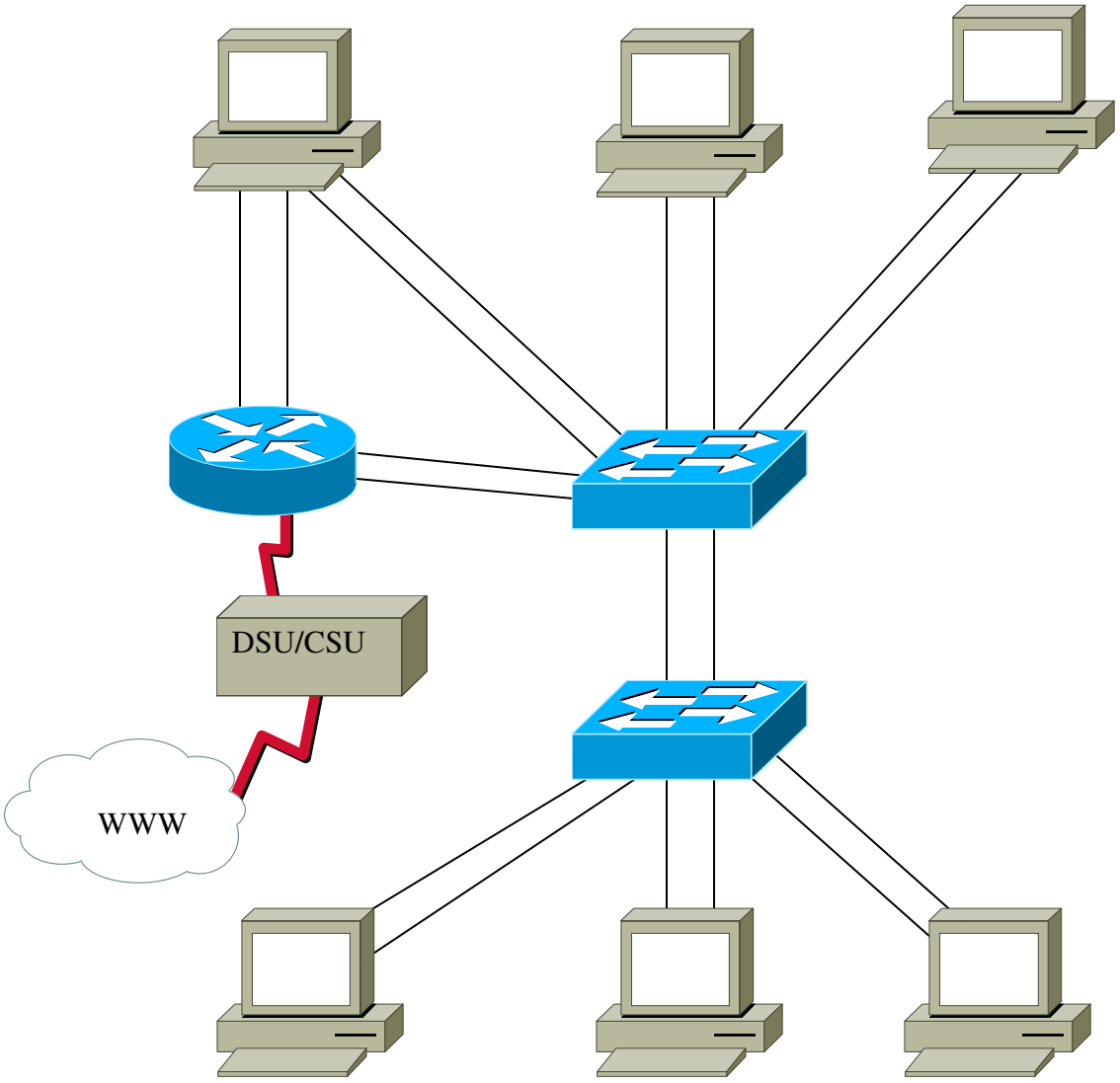
Change hubs to switches:



Add in a router:



Add in a web access:

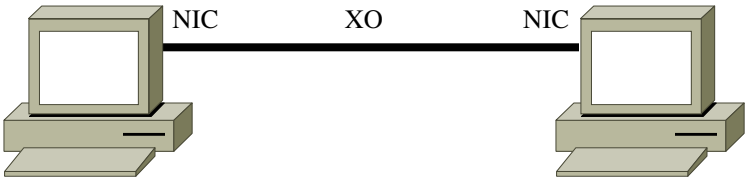


Peer-to-Peer Networking/File and Print Sharing

Objective:
To learn how to set up two computers to communicate and share files.

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

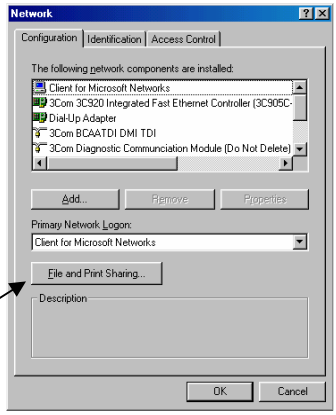
Lab Diagram:



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:

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. Do not reboot just yet…we have to enable file and print sharing first, then we can reboot the computer. Use the lab on "Installing a NIC: software" if you get stuck.
3. To enable file and print sharing right click on "network neighborhood" (just like you did for changing the TCP/IP settings. You should see:



file and print sharing

Figure 1—Network settings control panel.

4. Click on the file and print sharing box. You will see:

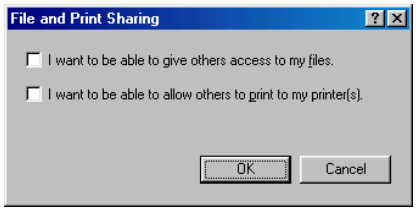


Figure 2—file and print sharing control panel

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:

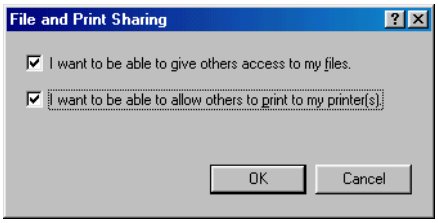


Figure 3—file and print sharing control panel

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.

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).

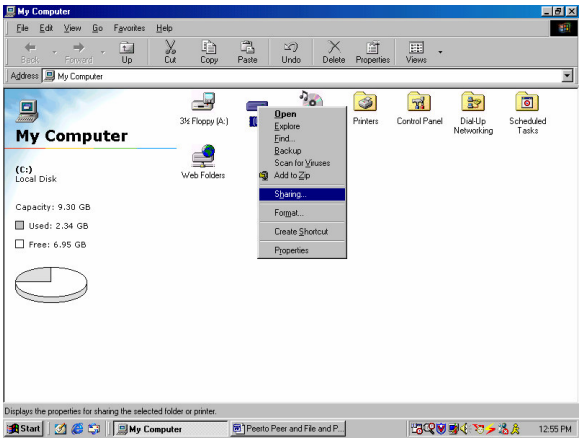


Figure 5—Now file sharing can be accomplished.

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 ☺)

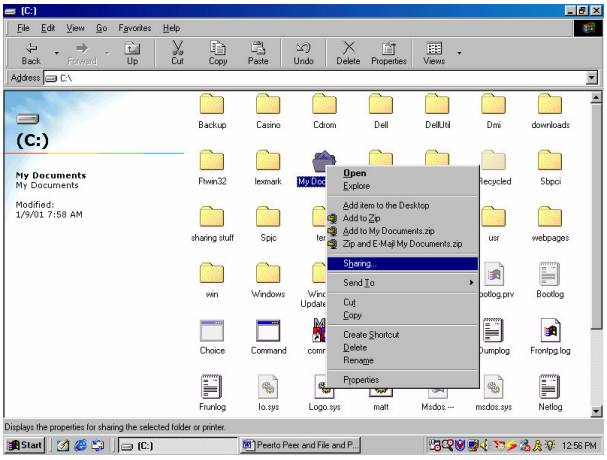


Figure 6—Selecting a specific folder to be shared.

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.

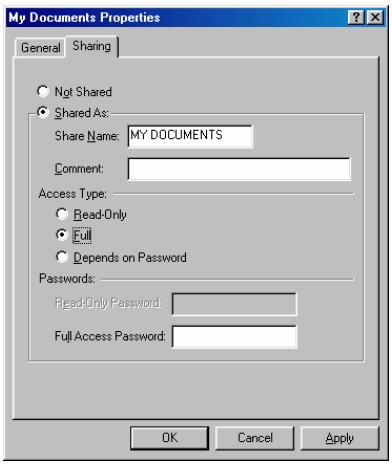


Figure 7—Selecting the options for a share.

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.

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?

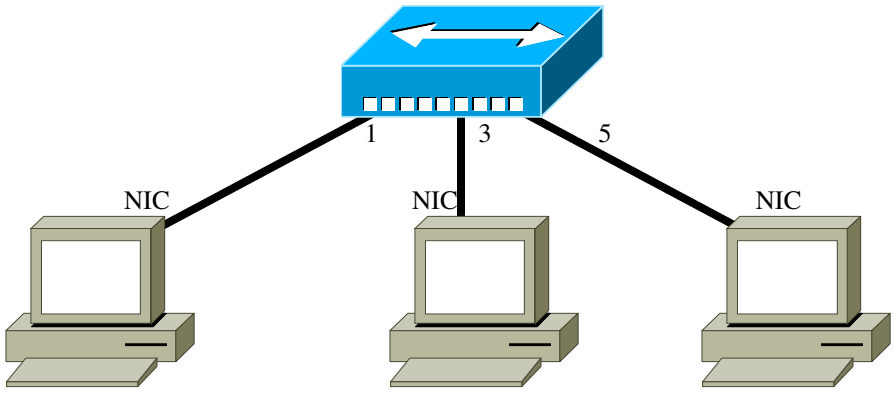
In this lab you have learned how to hook up two computers using peer-to-peer networking and file and print sharing. For this you needed to use your knowledge of TCP/IP software settings you learned in an earlier lab. In later labs you will be expanding upon this knowledge to build more complicated networks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

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
(1) Hub
(3) Straight-through cables

Lab Design:



Name:	A	B	C
IP address:	192.168.1.3	192.168.1.4	192.168.1.5
Mask:	255.255.255.0	255.255.255.0	255.255.255.0
Gateway:	none	none	none

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:

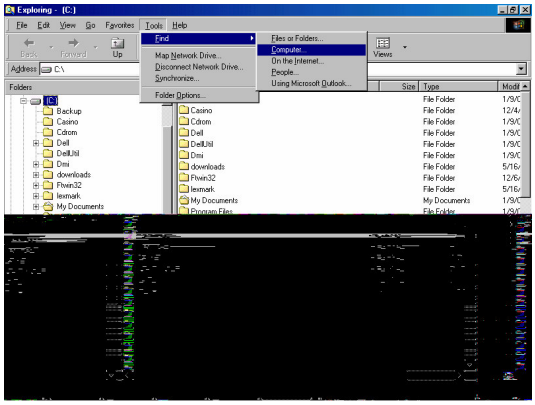


Figure 1—Using windows explorer to "find" computers on the network.

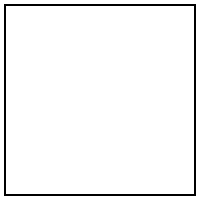


Figure 2—The "find computer" option pop up window.

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?

So What Have I Learned Here?

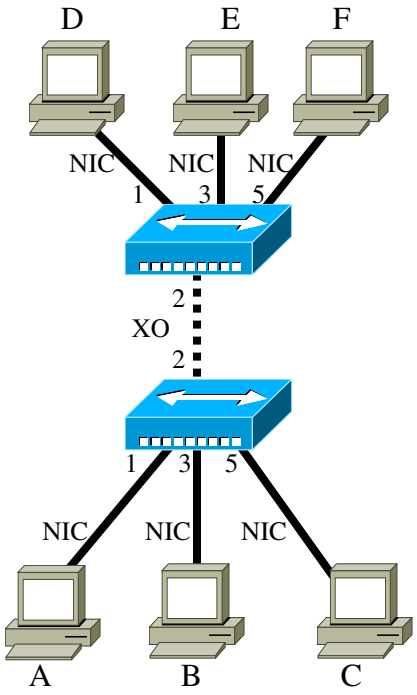
You have learned how to hook up several workstations to share files using a hub. You learned that the IP addresses had to be within the same subnet in order to communicate with each other. Also you were acquainted with the quirks of Microsoft networking for small networks. Microsoft really likes having 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
(6) Hub
(6) Straight-through cables (ST)
(1) Cross-over cable (XO)

Lab Design:



Name:	A	B	C
IP address:	192.168.1.3	192.168.1.4	192.168.1.5
Mask:	255.255.255.0	255.255.255.0	255.255.255.0
Gateway:	none	none	none

Name:	D	E	F
IP address:	192.168.1.13	192.168.1.14	192.168.1.15
Mask:	255.255.255.0	255.255.255.0	255.255.255.0
Gateway:	none	none	none

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.

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?

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

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:

	10^2	10^1	10^0
	100	10	1
532 =	5	3	2

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.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

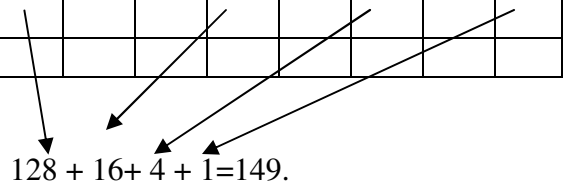
Each column heading represents a decimal number with a binary power. To convert between binary and decimal the rule is simple: Any place you have a "1" you just add the column heading to get the decimal total. For example, if we were given a binary number of 01101101 to convert into decimal we would 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.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	**64**	**32**	**16**	**8**	**4**	**2**	**1**
0	1	1	0	1	1	0	1

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.

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



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.

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							

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.

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						

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.

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					

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.

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				

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

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			

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

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		

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.

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	

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

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

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.

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							

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

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	1						

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.

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	1	0					

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

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	1	0	1				

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

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	1	0	1	1			

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

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	1	0	1	1	1		

Our next one: 3 - 2 = 1. So we put a "1" under the 2 heading and move to the next column heading.

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	1	0	1	1	1	1	

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

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	1	0	1	1	0	0	1

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

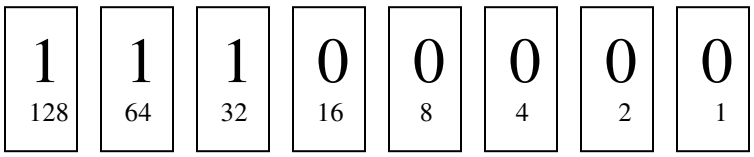
Then just subtract the column headings (in this case the little numbers on the bottom of the card)…234-128=106. Since it is a positive number flip the card and move on to the next one.

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

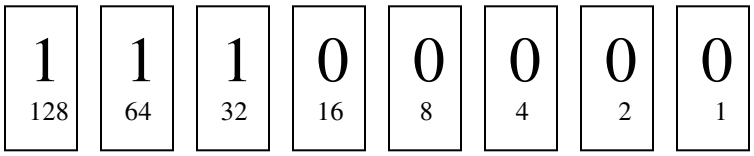
106 - 64 = 42. Since it is a positive number flip the card and move on to the next one.

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

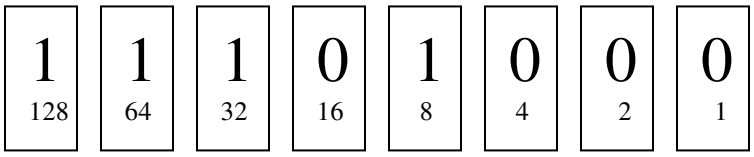
42 - 32 = 10. Since it is a positive number flip the card and move on to the next one.



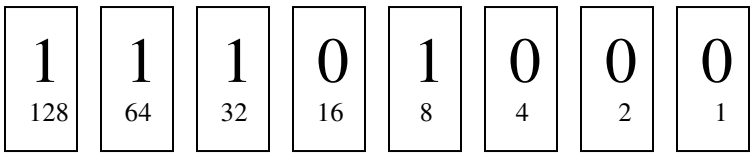
10 - 16 = -6. Since it is a negative number leave the card on zero and move on to the next one.



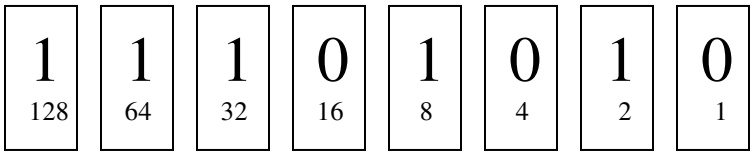
10 – 8 = 2. Since it is a positive number flip the card and move on to the next one.



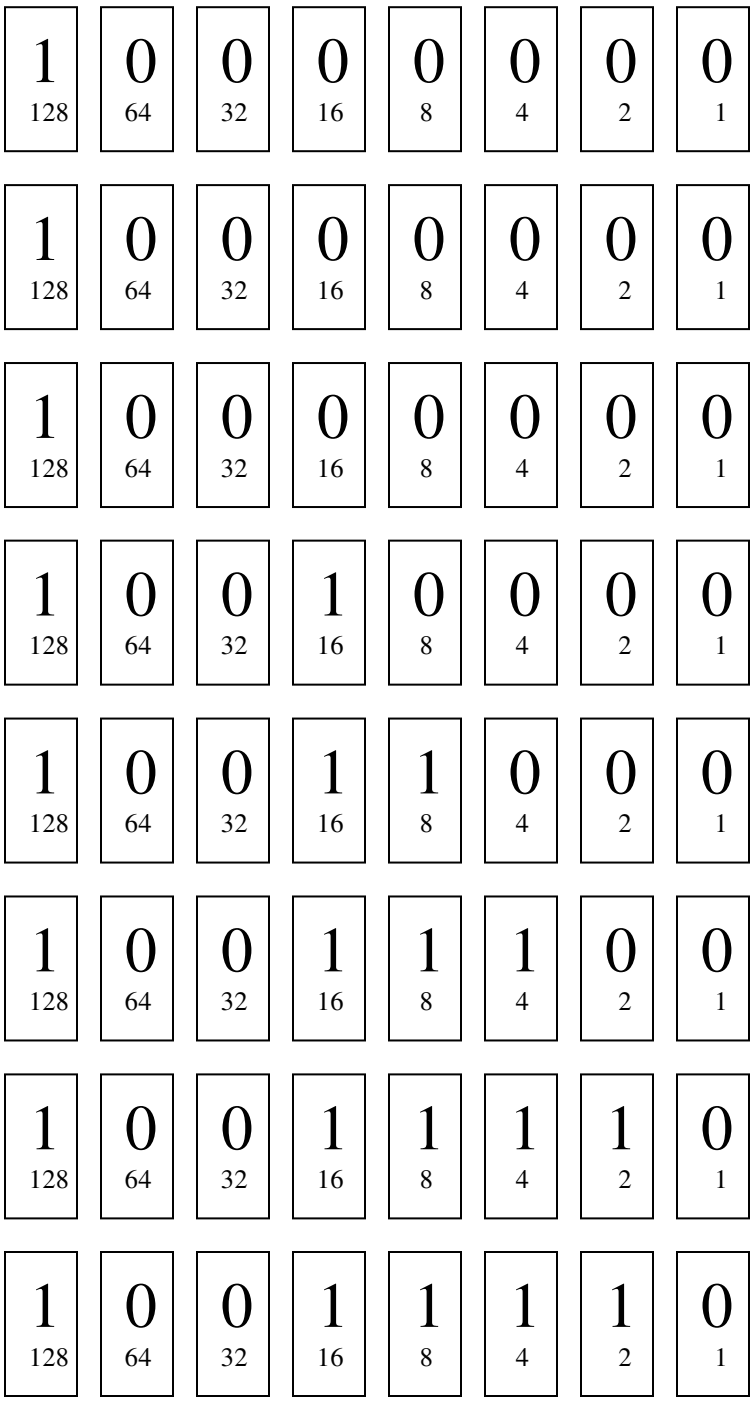
2 – 4 = -2. Since it is a negative number leave the card on zero and move on to the next one.



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).



Let me just walk through one more…you can do the math yourself. Let's convert 158 to binary.



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 two pages.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	**64**	**32**	**16**	**8**	**4**	**2**	**1**

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	**64**	**32**	**16**	**8**	**4**	**2**	**1**

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	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:

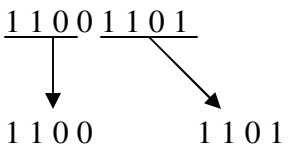
16^4	16^3	16^2	16^1	16^0
65536	**4096**	**256**	**16**	**1**

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, lets convert the binary octet 11001101 to hexadecimal.

1. We just divide the octet into two groups of 4 bits



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.

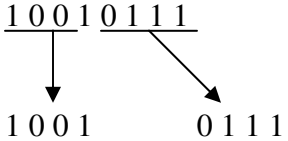
8 4 2 1	8 4 2 1
1 1 0 0	1 1 0 1
8+4=12	8+4+1=13

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



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.

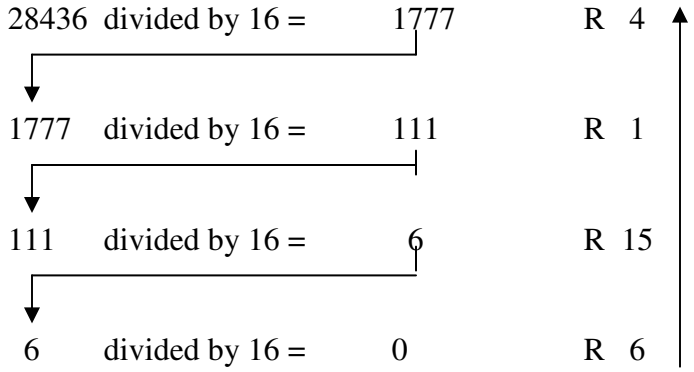
   9=9 7=7

   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:



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:

R 0	0/16	0.0000	R 8	8/16	0.5000
R 1	1/16	0.0625	R 9	9/16	0.5625
R 2	2/16	0.1250	R 10	10/16	0.6250
R 3	3/16	0.1875	R 11	11/16	0.6875
R 4	4/16	0.2500	R 12	12/16	0.7500
R 5	5/16	0.3125	R 13	13/16	0.8125
R 6	6/16	0.3750	R 14	14/16	0.8750
R 7	7/16	0.4375	R 15	15/16	0.9375

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:

B	11	multiplied by 16^0 =	11 x 1	=	11
3	3	multiplied by 16^1 =	3 x 16	=	48
C	12	multiplied by 16^2 =	12 x 256	=	3072
8	8	multiplied by 16^3 =	8 x 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.

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."

As a memory device we can remember from the top down that **A**ll **P**residents **S**eem **T**o **N**eed **D**ata **P**rocessors or "**A**ll **P**eople **S**eem **T**o **N**eed **D**omino's **P**izza." 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:

Application	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"
Presentation	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"
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.

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 is ″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 the 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.

Layer	CISCO definition	Your definition
Application	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"	
Presentation	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"	

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…its 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.

Paper Lab: LAN Topologies

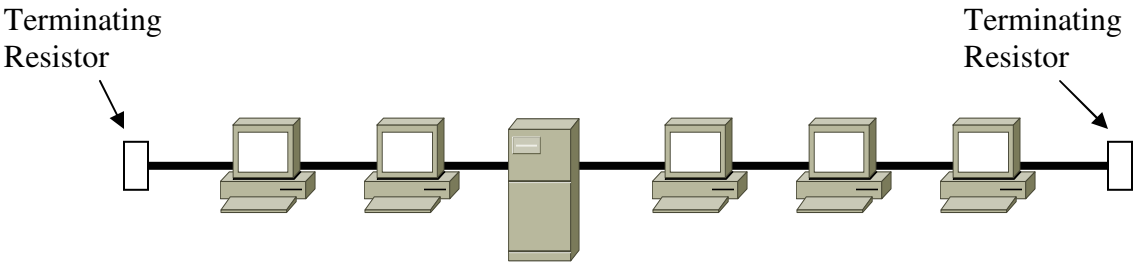
Objective:
To be able to learn more about the LAN topologies used in networking.

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.

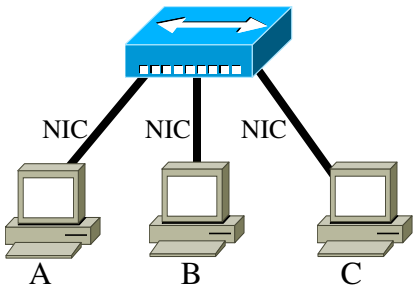
Bus Topology Diagram:



Bus 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.

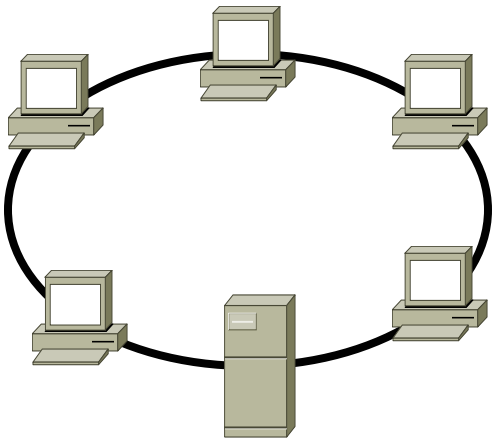
Star Topology Diagram:



Star topologies usually used 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.

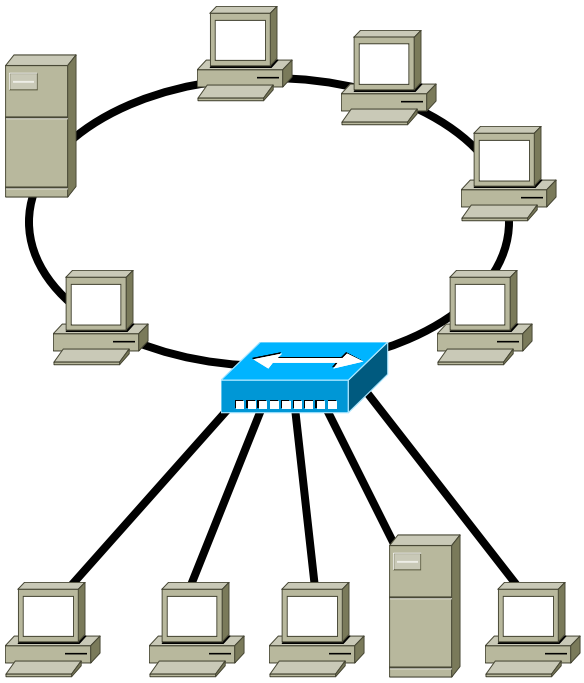
Ring topology diagram:



Ring 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:

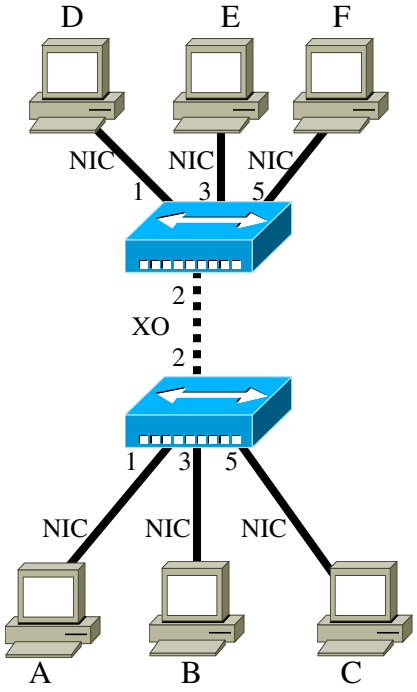


Star-Ring Hybrid Network

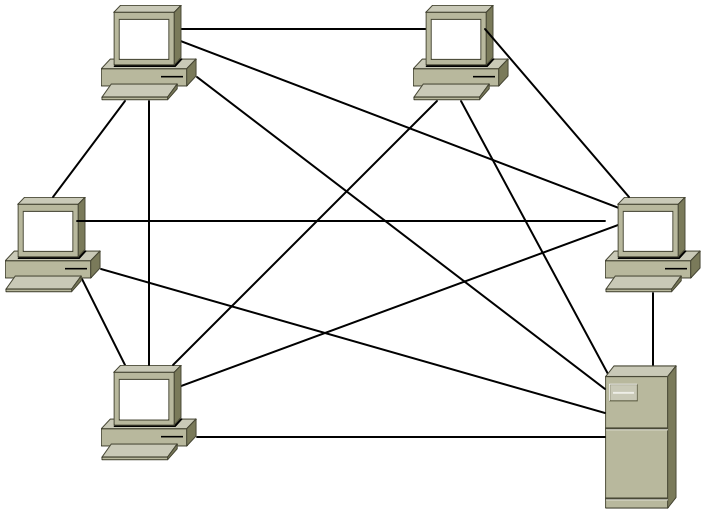
There are all kinds of other topologies that are just “more extreme” versions of the three basic topologies:

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

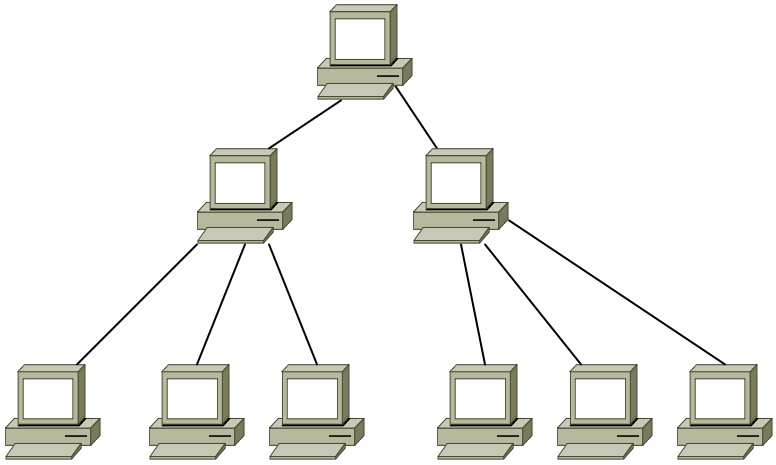
Extended Star: Two or more star networks connected together with a backbone cable.



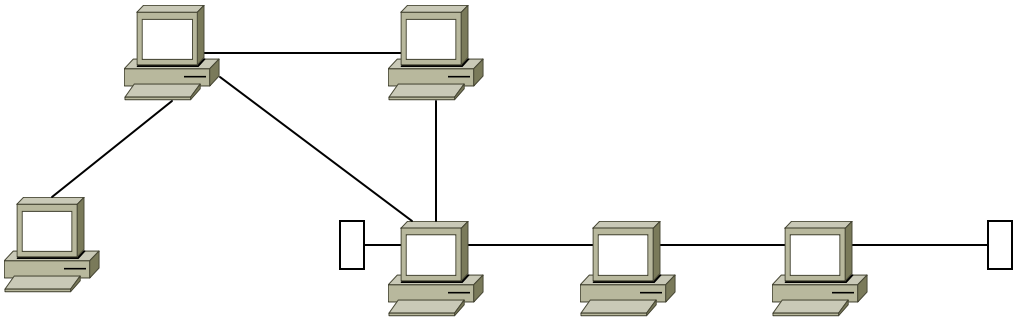
Mesh or Full-mesh: Every computer or networking device connected to every other computer or networking device (used primarily in frame relay networks).



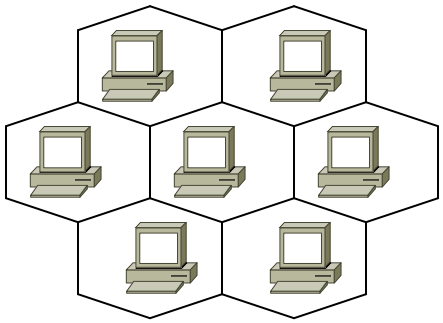
Tree: Like a hard drive structure with folders and documents. (I just used workstations to show the overall structure…other networking devices would be included and used to pass network traffic).



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).



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).



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.

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)

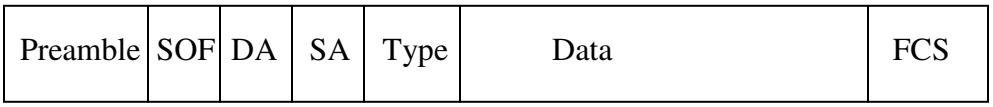


Figure 1—Generic Ethernet packet structure.

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: 0101010101010101010101010101010101010101010101010101010101010101 0101010101). 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 ***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#

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 Jell-O 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:

$$(X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X^{1} + 1)$$

802.2/802.3 Ethernet (RFC 1042)

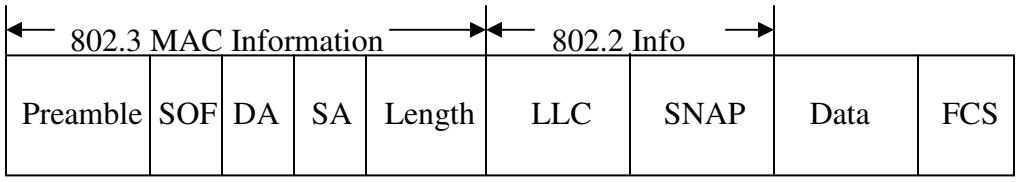


Figure 2—Ethernet SNAP packet structure.

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). This is more commonly used today.

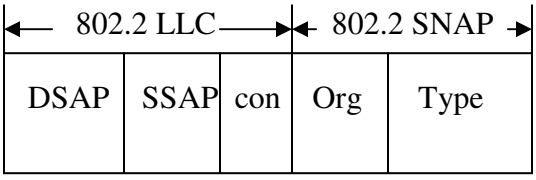
Stripped by the NIC:
The ***preamble*** can vary in length. The preamble basically is used to help set up the transmission and reception of the information through synchronization. The actual amounts of bits has varied over the years but the principle is still the same: a series of alternating zeroes 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: 1555555555555 In binary: 01010101010101010101010101010101010101010101010101010101010101010101010101 0101010101). 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).



The 802.2 LLC packet is composed of three fields:

> 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#

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). 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.

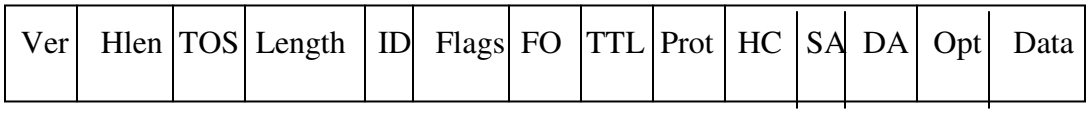


Figure 3—IP Data Field Composition

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.

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:

Type of Service	Binary	Hexadecimal
Normal service	0000	0x00
NNTP (Usenet news)	0001	0x02
IGP/SNMP	0010	0x04
FTP data//SMTP data/ DNS zone xfr/	0100	0x08
Telnet/Rlogin/DNS UDP query/SMTP command phase/TFTP	1000	0x10

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.

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 in networking. For your CCNA test you will probably not have anything this extensive and detailed. But when you get to the lab on using protocol inspectors it will be easier to understand.

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.

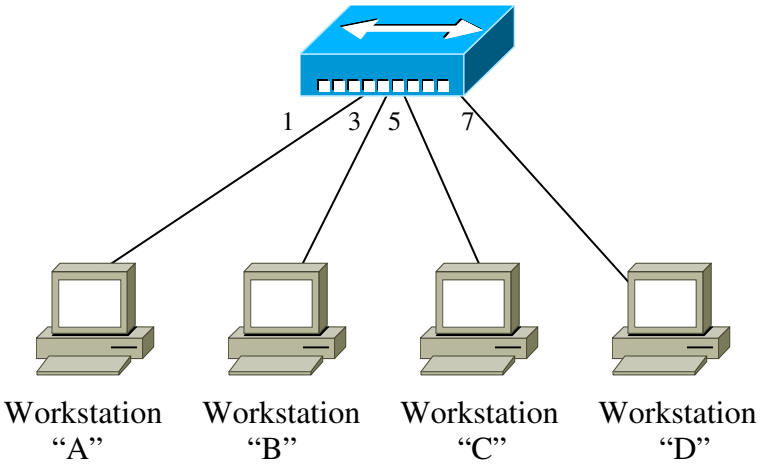


Figure 1—Small hubbed network.

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

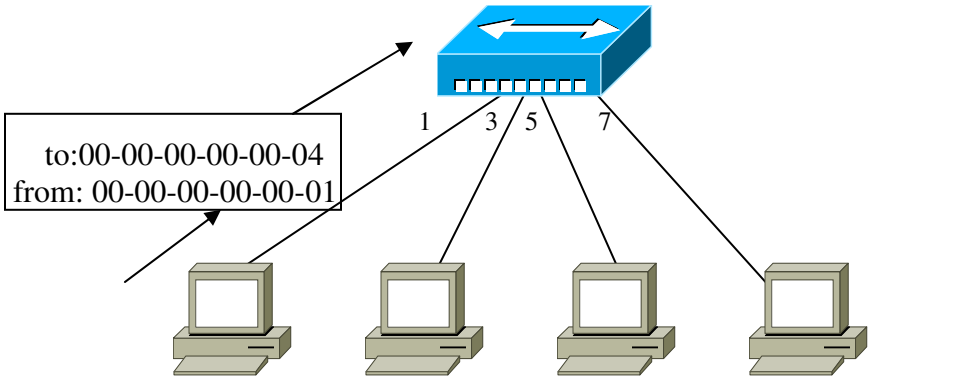


Figure 2—Workstation A sends a request to workstation D.

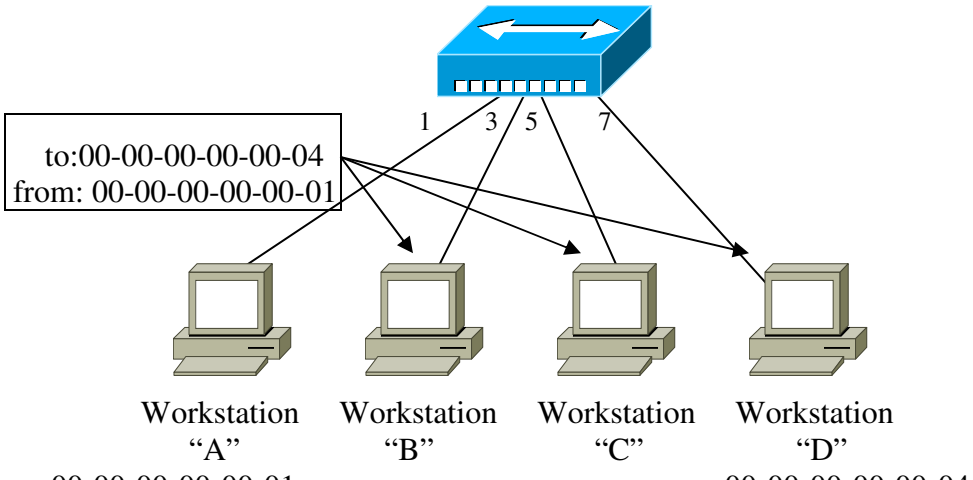


Figure 3—The information is duplicated and sent to every node attached to the hub.

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.

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 ping 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 accessible 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

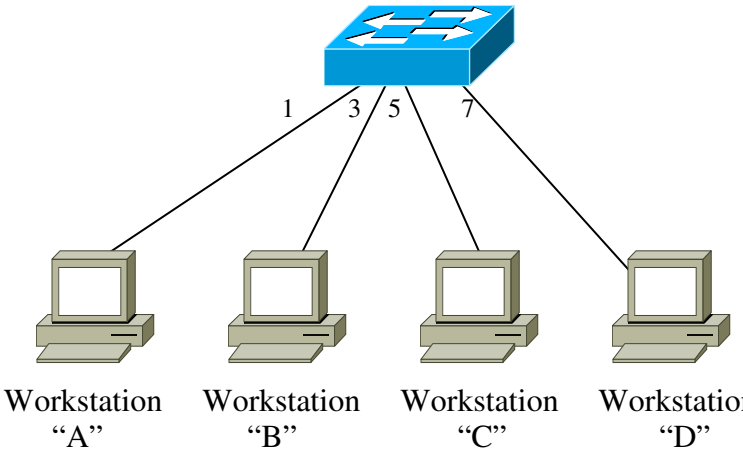


Figure 4—Small switched network.

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.

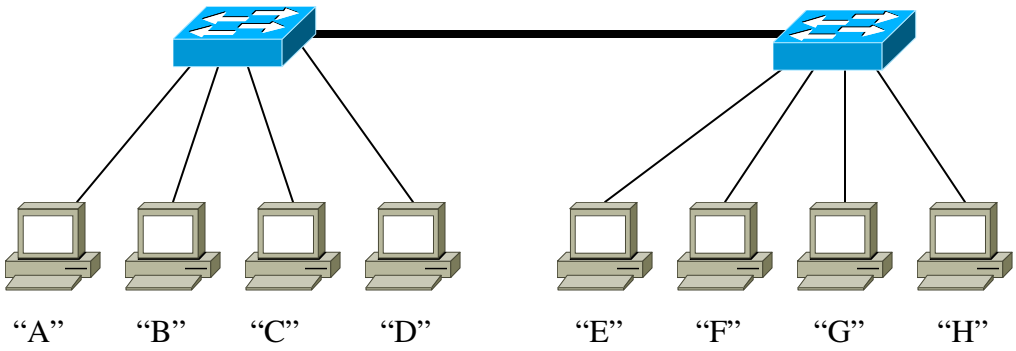


Figure 5—Small multiple-switched network.

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.

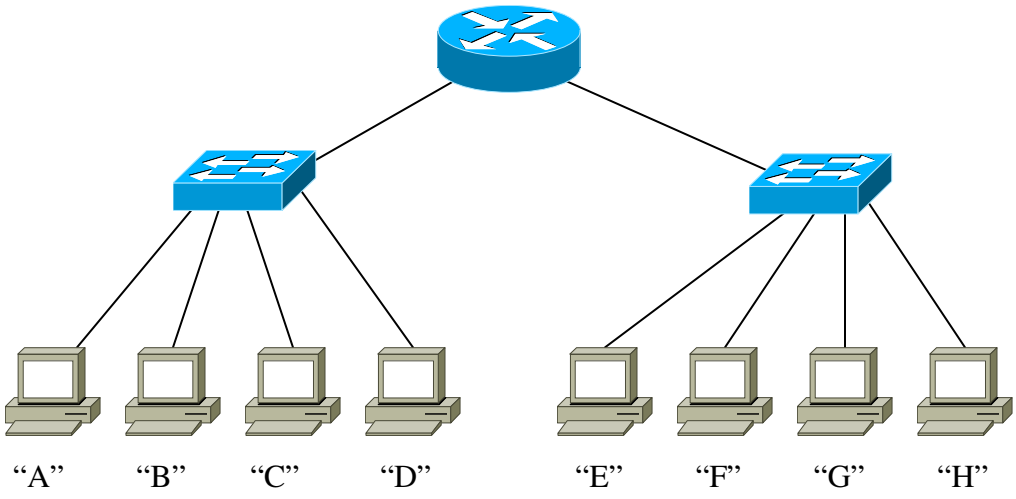
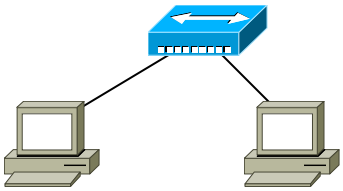


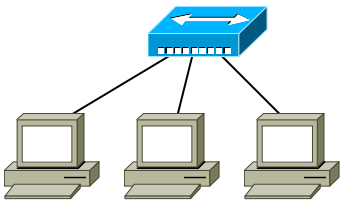
Figure 6—Small network.

Supplemental Labs or Challenge Activities:
Let's have you count up the number of collision domains and broadcast domains in several network types.

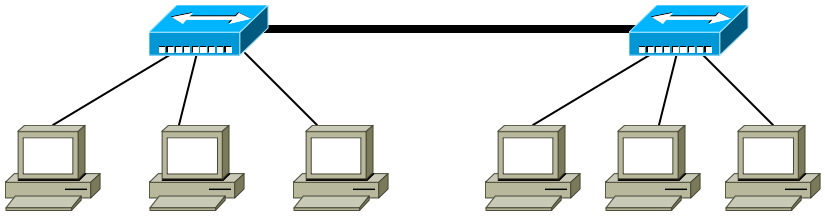
1. Collision Domains: _____ Broadcast Domains: _____



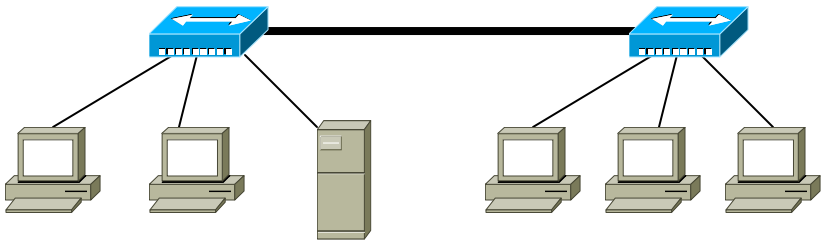
2. Collision Domains: _____ Broadcast Domains: _____



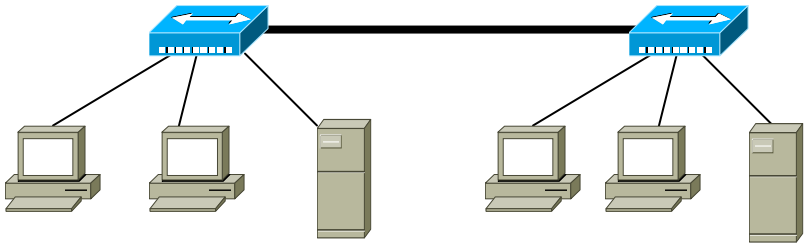
3. Collision Domains: _____ Broadcast Domains: _____



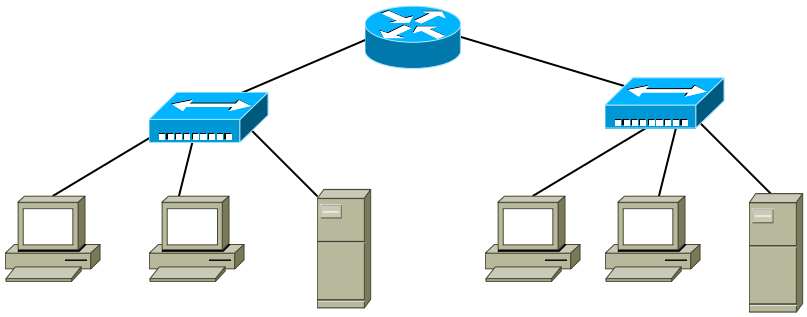
4. Collision Domains: _____ Broadcast Domains: _____



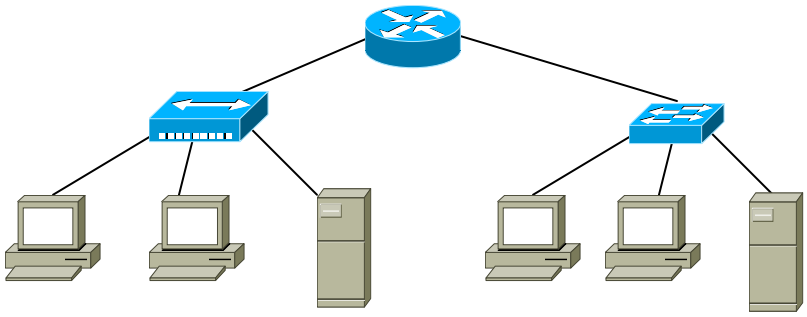
5. Collision Domains: _____ Broadcast Domains: _____



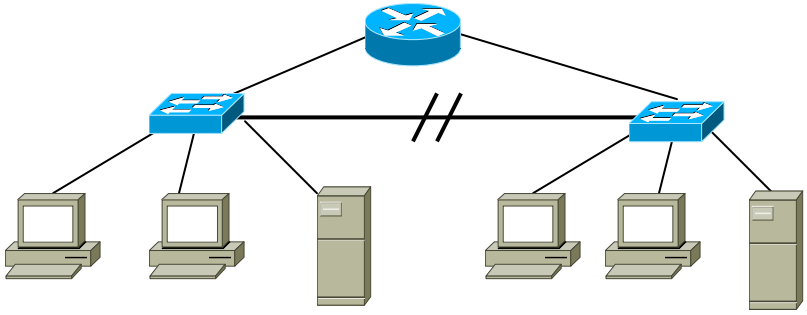
6. Collision Domains: _____ Broadcast Domains: _____



7. Collision Domains: _____ Broadcast Domains: _____



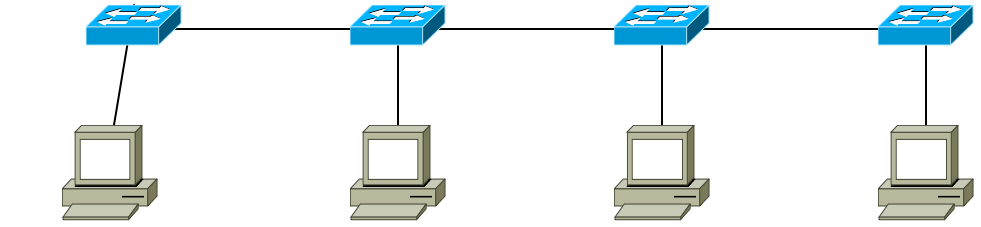
8. Collision Domains: _____ Broadcast Domains: _____



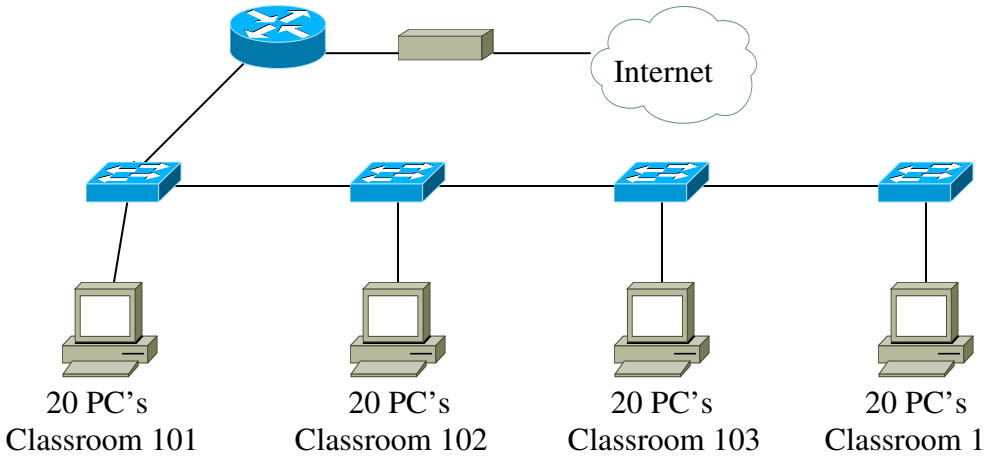
The redundant link will act as a backup in cast the main link goes down. You will learn how to set up redundant links between switches in Part 3.

Ok…got the idea? Let's start getting bigger!

9. Collision Domains: _____ Broadcast Domains: _____

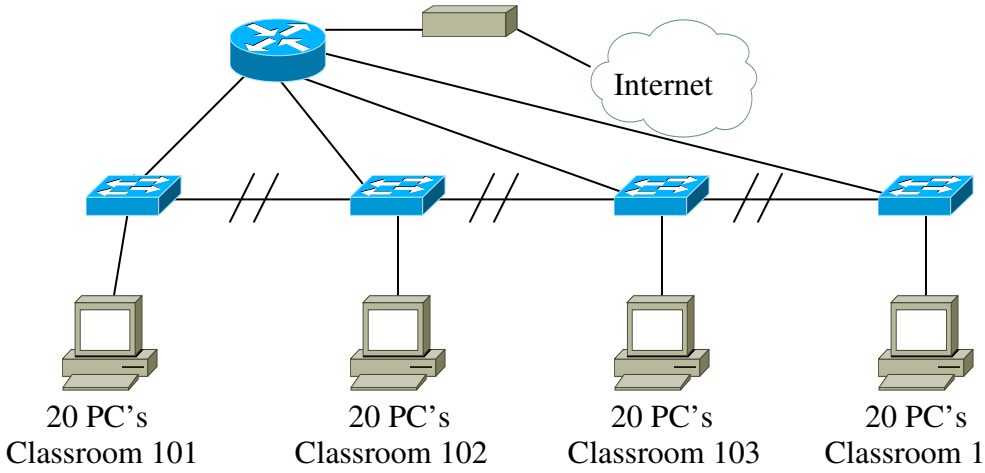


10. Collision Domains: _____ Broadcast Domains: _____



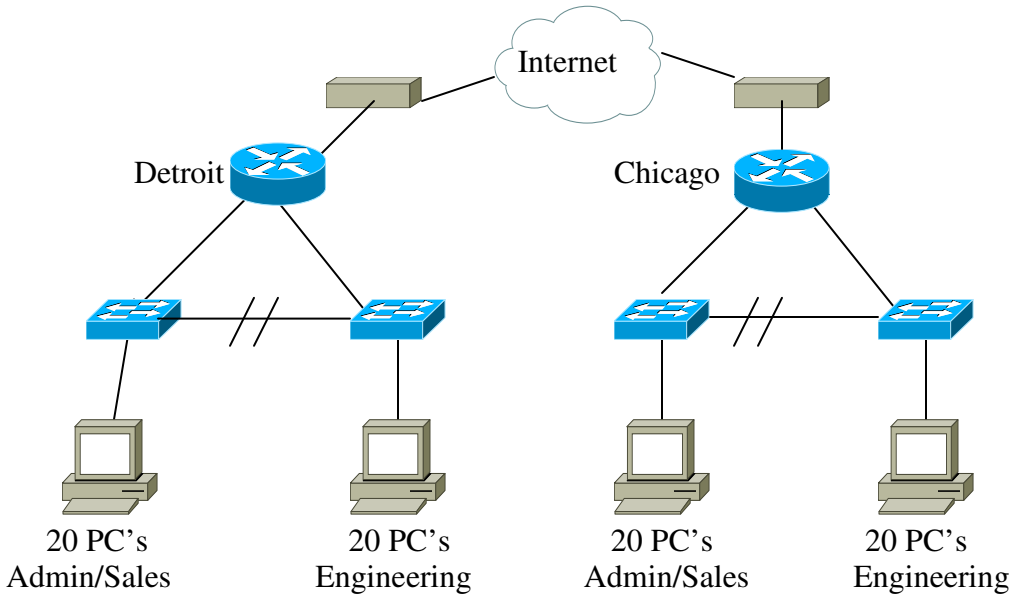
This is an "OK" design.

11. Collision Domains: _____ Broadcast Domains: _____



This is a better design.

12. Collision Domains: _____ Broadcast Domains: _____



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.

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.

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 using 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?
   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 147.30.74.01
   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?
   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?
   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?
    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?
    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?
    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?
    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?
    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?
    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?
    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 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 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. 00000-11111
    B. 00000000-111111111
    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. 1111111.1111111.1111111.00000000
   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?
   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?
   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

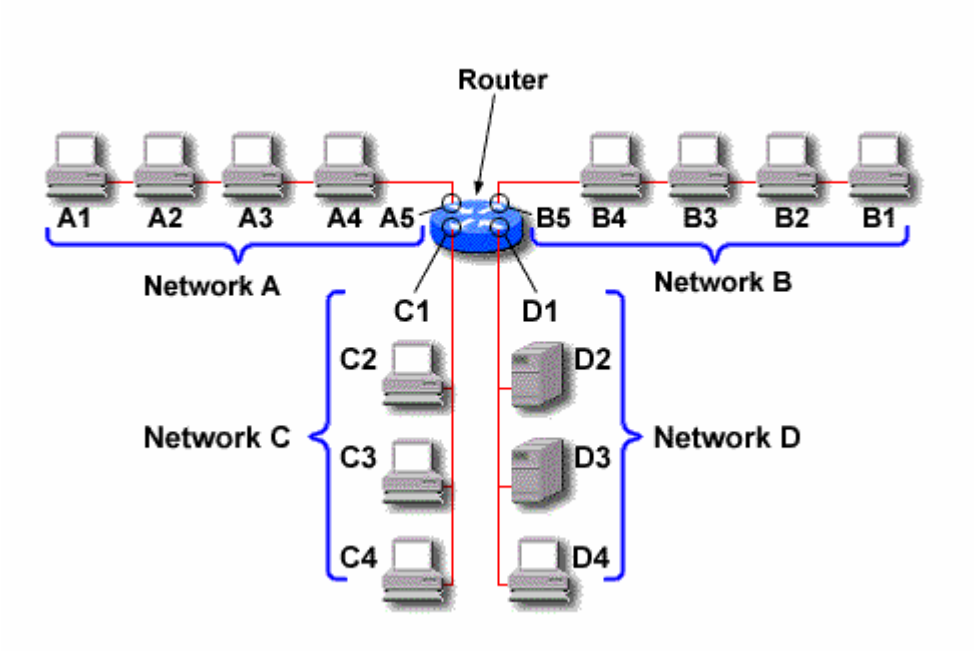
Router functions

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. 255.2555.0
   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.

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." Its 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.

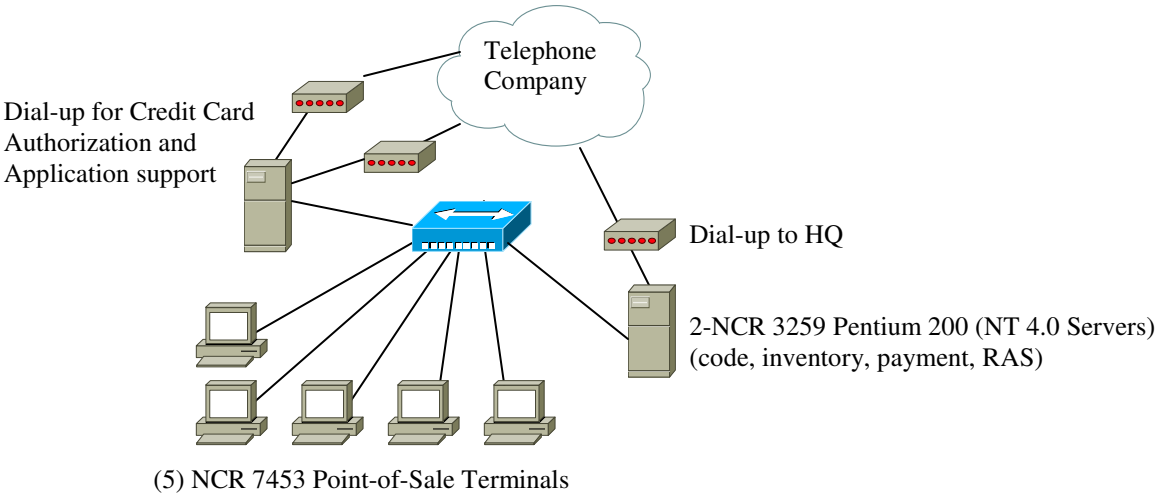
Subnetting Example: John's Brewhouse

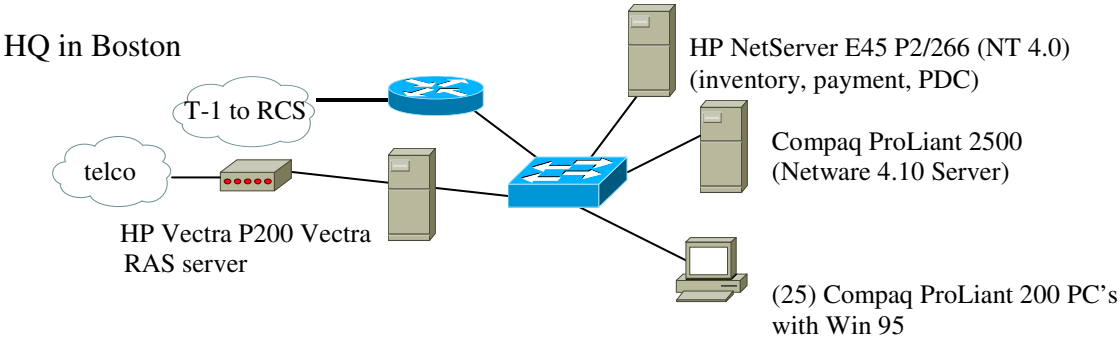
Objective:
To use your subnet knowledge to design an IP addressing scheme for the John's Brewhouse Restaurant Network.

Tools and Materials:
Paper and pencil

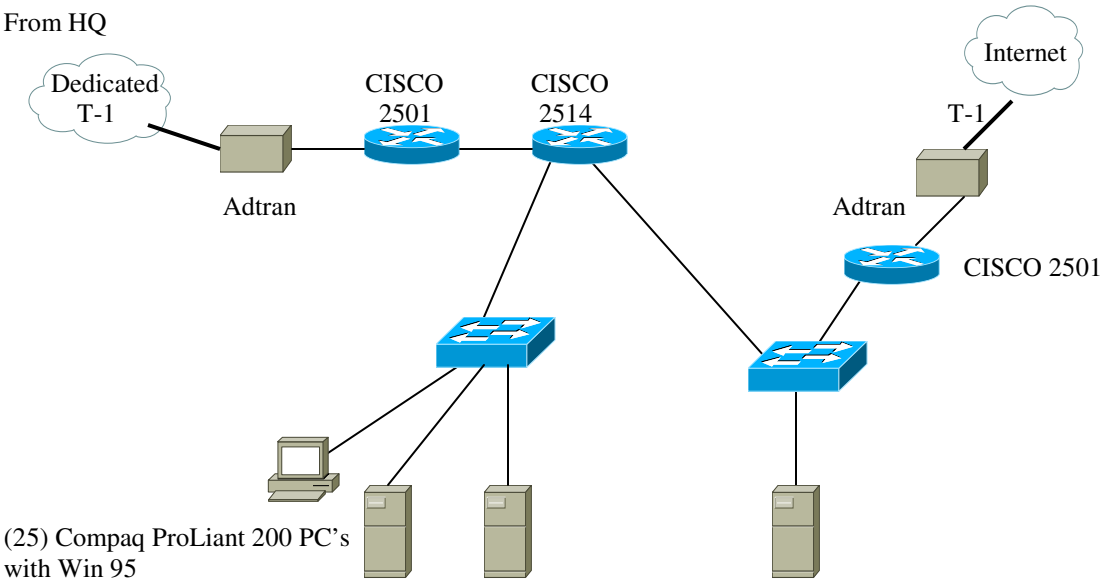
Background:
John Harvard's Brewhouse is a microbrewery/restaurant chain in New England. They have locations in Cambridge (MA), Framingham (MA), Wayne (PA), Springfield (PA), Pittsburgh (PA), Manchester (CT), Wilmington (DE), Providence (RI), Lake Grove (NY), and Washington DC. Three network topologies 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.



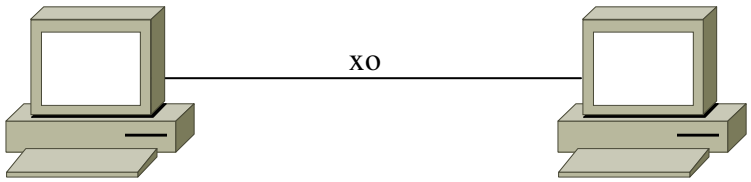
From Networking Computing Magazine Centerfold: John Harvard's Brewhouse.
http://www.networkcomputing.com/1005/1005centerfoldtext.html

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. Pick IP addresses and masks to make this peer-to-peer network function properly. Refer to the peer-to-peer lab if needed.
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. Trace route does not have any options.

```
C:\WINDOWS>ping /?
Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
        [-r count] [-s count] [[-j host-list] | [-k host-list]]
        [-w timeout] destination-list
Options:
   -t            Ping the specified host until stopped.
                       To see statistics and continue - type Control-Break;
                       To stop - type Control-C.
   -a            Resolve addresses to hostnames.
   -n count      Number of echo requests to send.
   -l size       Send buffer size.
   -f            Set Don't Fragment flag in packet.
   -i TTL        Time To Live.
   -v TOS        Type Of Service.
   -r count      Record route for count hops.
   -s count      Timestamp for count hops.
   -j host-list  Loose source route along host-list.
   -k host-list  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.
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:\WINDOWS\Desktop>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:\WINDOWS\Desktop>
```

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:\WINDOWS\Desktop>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
C:\WINDOWS\Desktop>
```

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. 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:

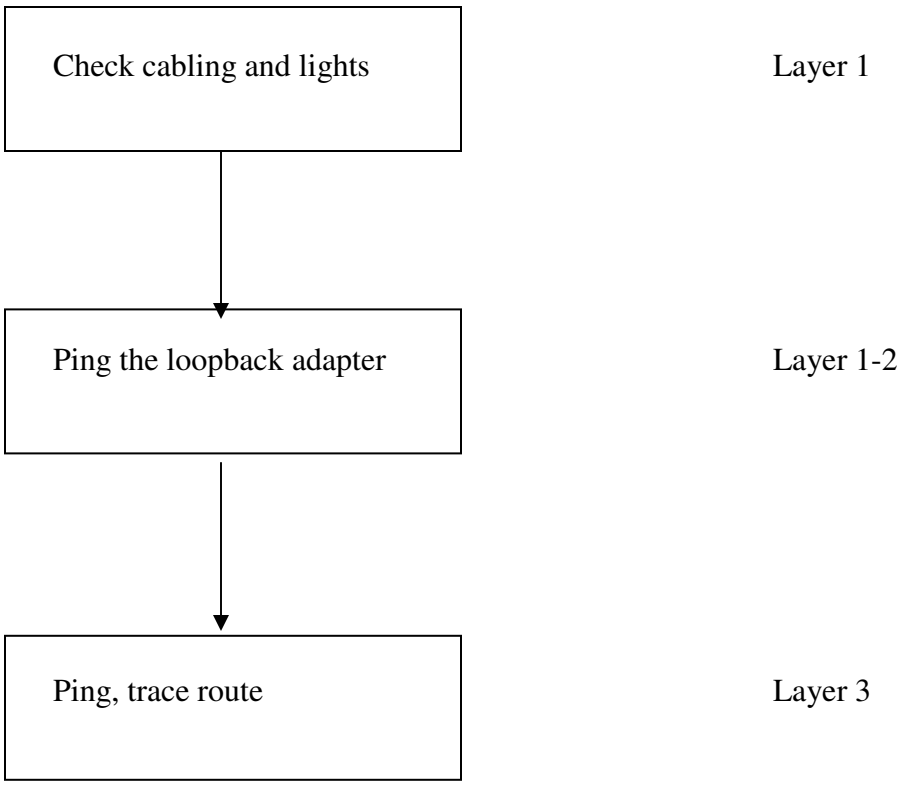
```
C:\WINDOWS\Desktop>tracert www.spjc.edu

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.

C:\WINDOWS\Desktop>
```

Now, when troubleshooting if we ran a trace route and got this:

```
C:\WINDOWS\Desktop>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.

C:\WINDOWS\Desktop>
```

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.

Basic Troubleshooting



Supplemental Lab or Challenge Activity:

1. Write a ping command that will continuously ping another workstation. When would you want to do this? Be careful! This is illegal…find out why.
2. Write a ping command to send 50000 bytes packets.
3. Write a ping command to send ten 50000 byte packets.
4. Write a ping command to send 100000 byte packets.
5. Open up multiple DOS windows and send pings to each workstation in your classroom only at the same time.
6. Go find out what a traffic generator is…how could you use your knowledge of ping to make a traffic generator?
7. 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.

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.

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.

Step-By-Step Instructions:

1. Open up a DOS window.
2. Then type "ipconfig" to see your IP settings using DOS. If you type "winipcfg" here it will open a windows utility to do the same. From DOS you should see something like this:

```
C:\WINDOWS\Desktop>ipconfig

Windows 98 IP Configuration

0 Ethernet adapter :
        IP Address. . . . . . . . . : 0.0.0.0
        Subnet Mask . . . . . . . . : 0.0.0.0
        Default Gateway . . . . . . :

1 Ethernet adapter :
        IP Address. . . . . . . . . : 192.168.151.122
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . : 192.168.151.1

C:\WINDOWS\Desktop>
```

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.

```
C:\WINDOWS\Desktop>ipconfig /?
Windows 98 IP Configuration
Command line options:
 /All - Display detailed information.
 /Batch [file] - Write to file or ./WINIPCFG.OUT
 /renew_all    - Renew   all adapters.
 /release_all  - Release all adapters.
 /renew   N    - Renew   adapter N.
 /release N    - Release adapter N.
C:\WINDOWS\Desktop>
```

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

```
C:\WINDOWS\Desktop>ipconfig /release_all
Windows 98 IP Configuration
0 Ethernet adapter :

        IP Address. . . . . . . . . : 0.0.0.0
        Subnet Mask . . . . . . . . : 0.0.0.0
        Default Gateway . . . . . . :

1 Ethernet adapter :
        IP Address. . . . . . . . . : 0.0.0.0
        Subnet Mask . . . . . . . . : 0.0.0.0
        Default Gateway . . . . . . :
C:\WINDOWS\Desktop>
```

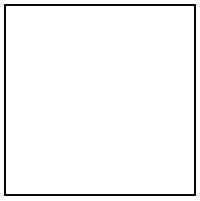
Then we can use ipconfig /renew_all to get a new one from the DHCP server. You should see:

```
C:\WINDOWS\Desktop>ipconfig

Windows 98 IP Configuration

0 Ethernet adapter :
        IP Address. . . . . . . . . : 0.0.0.0
        Subnet Mask . . . . . . . . : 0.0.0.0
        Default Gateway . . . . . . :

1 Ethernet adapter :
        IP Address. . . . . . . . . : 192.168.151.124
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . : 192.168.151.1

C:\WINDOWS\Desktop>
```

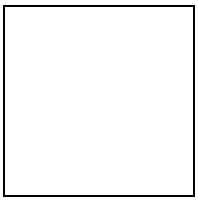
5. Notice how our address may differ slightly. When we give up our IP address it usually will go to one of the next devices requesting an IP…sometimes we get the same one back and sometimes we do not. Sometimes we encounter an error like this:

```
C:\WINDOWS\Desktop>ipconfig /renew_all
IP ConfigurationError

DHCP Server Unavailable: Renewing adapter ""

Windows 98 IP Configuration

0 Ethernet adapter :

        IP Address. . . . . . . . . : 0.0.0.0
        Subnet Mask . . . . . . . . : 0.0.0.0
        Default Gateway . . . . . . :

1 Ethernet adapter :

        IP Address. . . . . . . . . : 169.254.60.217
        Subnet Mask . . . . . . . . : 255.255.0.0
        Default Gateway . . . . . . :

C:\WINDOWS\Desktop>
```

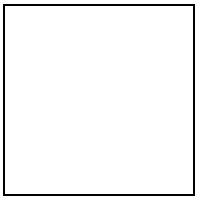
   Notice how our IP address is within the 169 network. Does this mean it worked? Not at all. Microsoft uses the 169 address as a "place holder" in case something goes wrong with DHCP.
6. Next, let's try the same thing with Windows. You can type winipcfg from the DOS prompt or from the RUN utility. You should see something like this when you first open it up:



   Notice how the IP configuration window comes up on the PPP adapter. This is not our NIC. We need to scroll down from the PPP adapter to our NIC. First, let's open up the scroll window:



You can see I am using a 3Com Etherlink PCI NIC in my computer. When I select that one then I can see my IP settings:



release all

renew all

The settings are similar to what we found in DOS. Instead of typing ipconfig /release_all now we can just hit the Release All button. When you release it will clear the ip addresses, masks and gateways. When you renew then you will get them back.

Supplemental Labs or Challenge Activities:

1. What kind of information can be found using the "more info" button? Can you get the same information from DOS?
2. Why did the IP information come up on "1 Ethernet Adapter" and not on the "0 Ethernet Adapter?"

So What Have I Learned Here?

You have learned how to release and renew the DHCP address from a workstation. In part 2 you will work more with DHCP and need to know how to do what we learned in this lab.

Free Protocol Inspector

Objective:

You will find here instructions on how and where to download a free protocol inspector. It's not real pretty but it works…and it's free. I use it through out 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).

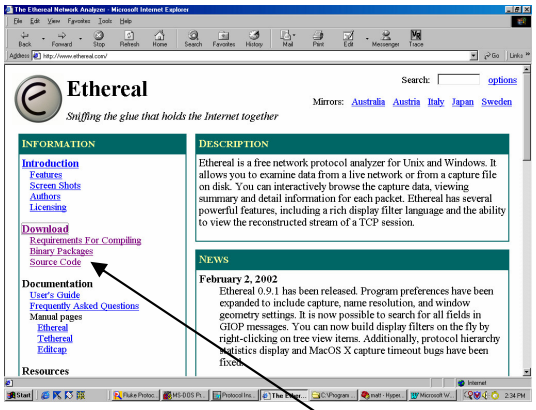


Figure 1—The "right" site.

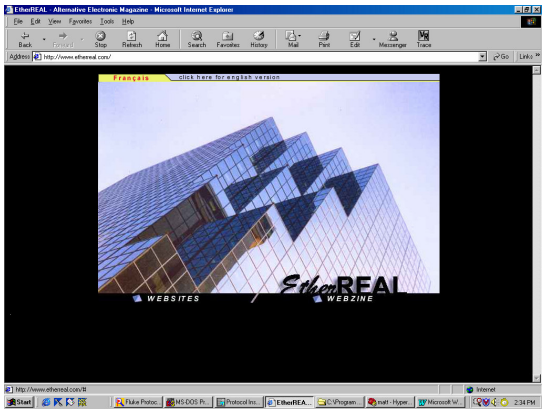


Figure 2—The "wrong site."

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).

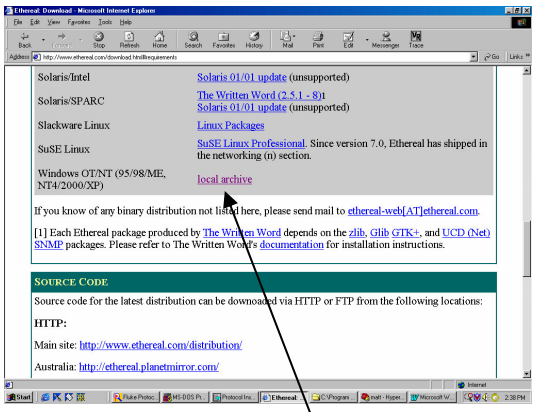


Figure 3—Click on "local archive."

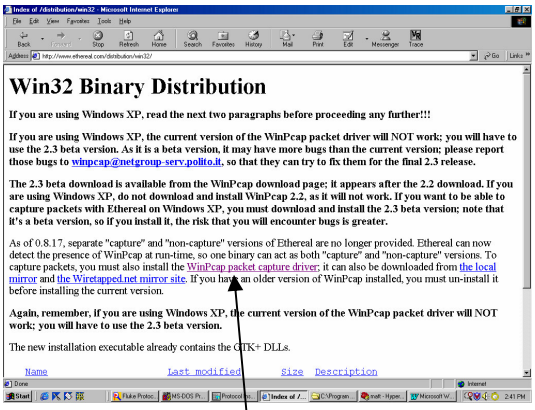
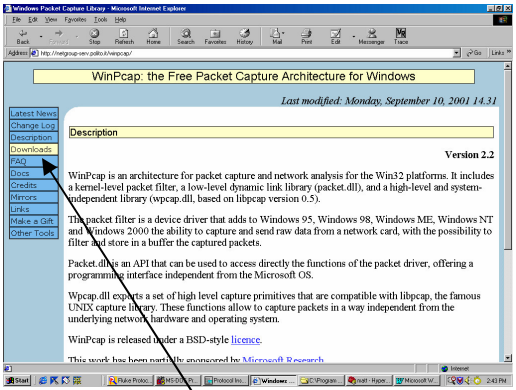


Figure 4—Click on Winpcap.

5. You need a driver library to make this work. Click on the Winpcap packet driver library link (see figure 4).
6. Click on "downloads" on the left side tool bar (see figure 5).
7. Click on Winpcap Auto-installer (driver + library) link (see figure 6). The file should start the download process. Don't forget where you put it. Execute this download file before running Ethereal or you will get an error message.



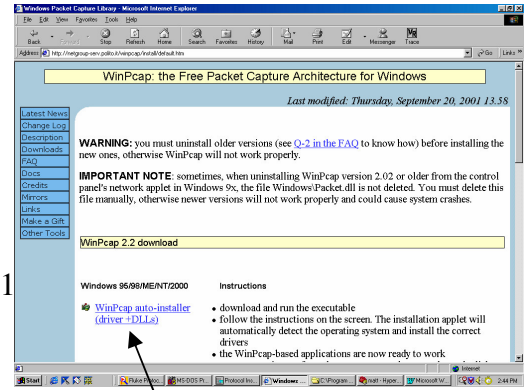
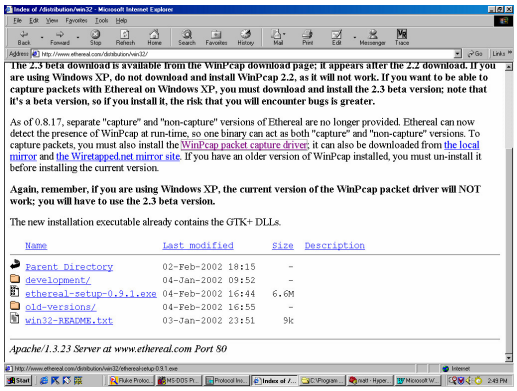


Figure 5—Click on downloads.

Figure 6—Click on winpcap auto-installer.

8. Click on the back browser to get back to the ethereal download window (see fig. 4).
9. Scroll down and click on the "ethereal-setup-0.9.1.exe." The file download process should start.
10. 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). Click "OK" at the bottom of that window to start the capture.



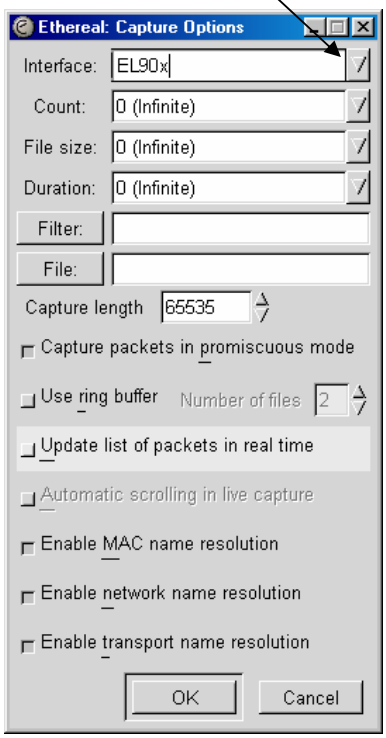


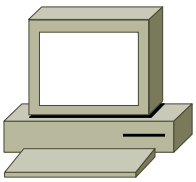
Figure 7—Click on ethereal-setup.

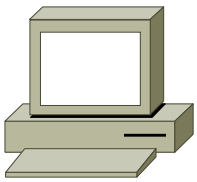
Using a Protocol Inspector

Objective:
To learn how to use a protocol inspector in a simple network setting.

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

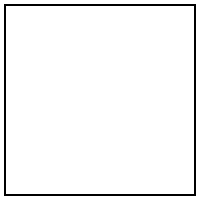
Lab Diagram:



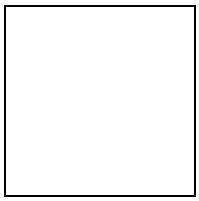


Step-By-Step Instructions:

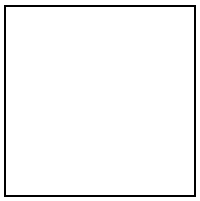
1. Set up and cable the lab as shown. Be sure to use IP addresses on the same subnet and using the same mask. Test your connectivity by pinging each other.
2. Open up Ethereal.
3. Start a "capture" of packets by using "control" plus "K" at the same time or using the capture pull-down menu and selecting start.
4. The capture preferences will open. Change from the MAC to the Ethernet Adapter. It should look like this (my NIC is ELO9x):



5. Then click on "ok." You should see the counters start for each protocol. It will look something like this:



6. 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 over-all 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:

1. Go to the Ethereal website and find the sample packets. Get the one on IPv6. How does it differ from IPv4?
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. We've looked at Ethernet packet structure. Go out and research icmp packet structures.

FTP/TFTP Lab

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 email 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 3.0.1 (859535212)
220-Welcome to ftp1.ipswitch.com
220-This server is located in Massachusetts, USA
220 ftp1.ipswitch.com X2 WS_FTP Server 3.0.1 (859535212)
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:

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

5. Type cd pub to change to the pub sub-directory.
6. Type dir to see what files and directories are available. Write them down here:

_____
_____
_____
_____
_____
_____
_____

7. Type cd msdos to change to the msdos sub-directory.
8. Type dir to see what files and directories are available. Write them down here:

_____
_____
_____
_____
_____
_____
_____

9. Type "get mmap.exe" to download the file to your computer.
10. Type "lcd" to find out where it put it on your computer. Where did it go?

_____
_____

11. Now you can go out an open the program. It will show you a map of your memory on your computer.
12. Type ? to see what commands are available. Write them down.

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

13. 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.
    Help ! _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____

14. 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?

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?

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. If this site still works: http://www.sniffer.com/naicommon/registration/survey.asp?code=gw73a This is a good protocol poster but they come and go so quickly. It is a registration site and then they will mail you a poster…they claim it will take 2-3 weeks…doesn't help you much here though. Instant access: http://ihide.virtualave.net/archive/protpost.pdf
2. Fill in the tables for the various protocol suites. Some protocols have overlap so be careful.

TCP/IP Suite

OSI Model	Protocol
7	
6	
5	
4	
3	

Novell Suite

OSI Model	Protocol
7	
6	
5	
4	
3	

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.

IBM Suite

OSI Model	Protocol
7	
6	
5	
4	
3	

ISO Suite

OSI Model	Protocol
7	
6	
5	
4	
3	

DECnet Suite

OSI Model	Protocol
7	
6	
5	
4	
3	

XNS Suite (Xerox)

OSI Model	Protocol
7	
6	
5	
4	
3	

Appletalk Suite

OSI Model	Protocol
7	
6	
5	
4	
3	

Banyan Vines Suite

OSI Model	Protocol
7	
6	
5	
4	
3	

Layer 2 Technologies: LAN's

OSI Model	Protocol

Layer 2 Technologies: WAN's

OSI Model	Protocol

Layer 1 Technologies

OSI Model	Protocol

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.

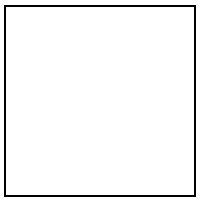
Telnet Lab

Objective:
To learn how to use terminal emulation (TELNET) software for Internet connectivity.

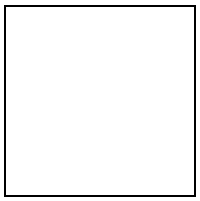
Background:
During your studies you will use many different software packages: FTP, TFTP, DOS, Protocol Inspector, and now you will learn TELNET. We saw it briefly back in the DOS lab but now we will use it to visit government sites, gopher sites, and other types of sites. We will also look briefly at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

Step-By-Step Instructions:

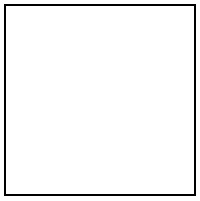
1. Open the telnet application. A quick way to do this is to click on Start>Run then type in telnet and press "ok." You should see the program come up like this:



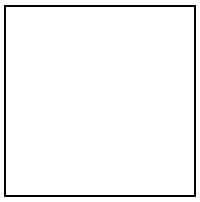
2. Start by reviewing everything in the help files. This will acquaint you more with what telnet can and cannot do.
3. Let's start with an easy one. Let's telnet to the Library of Congress. Start with this:



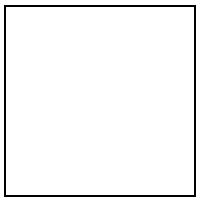
Once you click "connect" you should see this (after a couple of seconds):



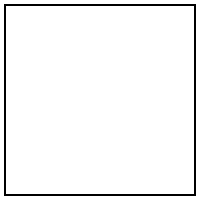
4. Let's try to telnet to a "MUD" site (multiple user dungeon)…it's a gaming site.



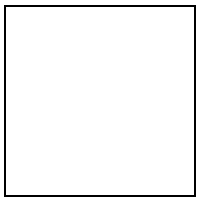
5. If there is an available line you will see:



6. Fun…isn't it? Here's one for the "Hard Drive Café"



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:



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."

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.thedirectory.org/ . It's fun for the whole family.

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 2nd Degree Felony, punishable by a minimum of 15 years for the 1st offense!*******

Port	Service	What it is…
7	Echo	Whatever you type in is repeated
9	Discard/null	
11	Systat	Lots of info on users in network
13	Daytime	Time and date at computer's location
15	Netstat	Lots of info on network—a must see!
19	Chargen	ASCII character stream
21	ftp	Transfer files
23	telnet	Terminal emulation program
25	Smpt	Mail program
37	Time	Time
39	Rlp	Resource location
43	Whois	info on hosts and networks
53	Domain	Name server
70	Gopher	Out-of-date information tool
79	Finger	UNIX information finder
80	http	Web server
110	Pop	Email post box server
119	nntp	News group servers
443	Shttp	Secure web servers
512	biff	Mail notification
513	rlogin	Remote login
514	Shell	Shell account for UNIX
520	Route	Routing information protocol

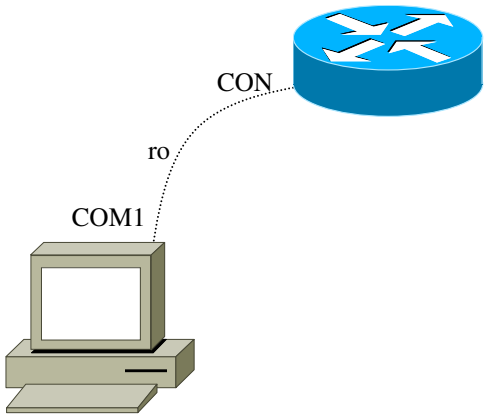
Hyperterminal Lab

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:
   9600 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 many 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!

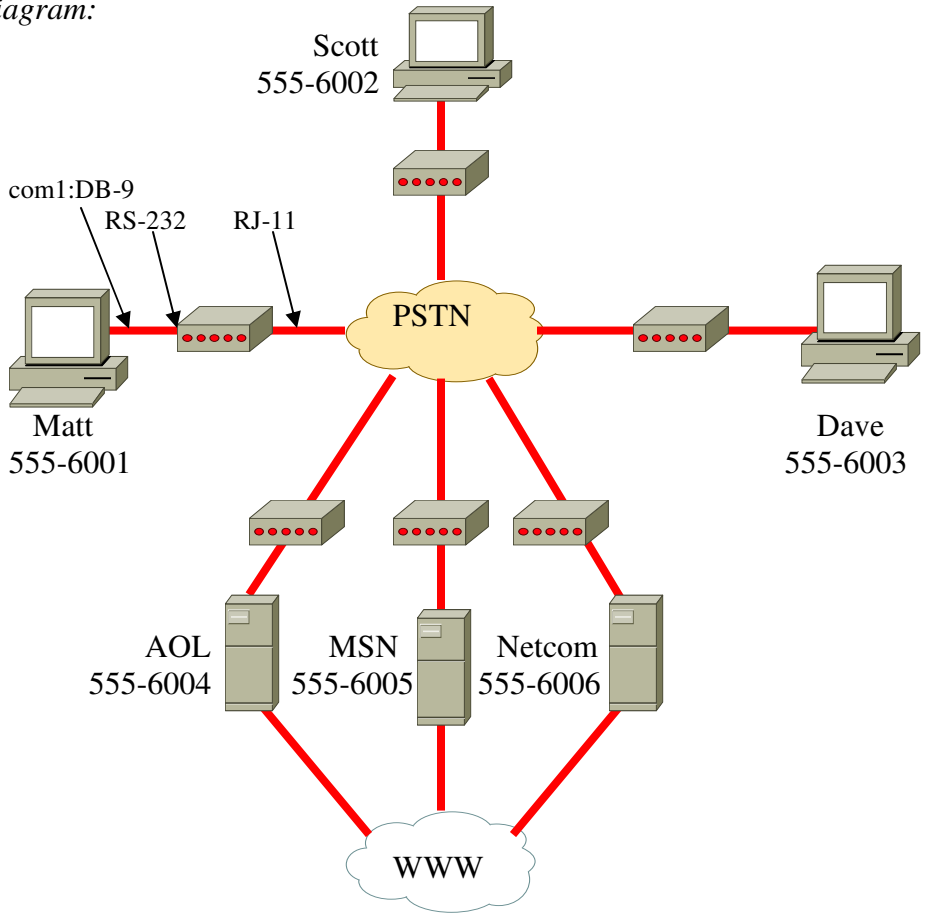
So What Have I Learned Here:

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.

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 you computer has dial-up networking capabilities first. If not, then you will have to install dial-up networking software from your Windows installation CD.
   a. Double-click on the "my computer" icon on your desktop.
   b. If you have a folder called "dial-up networking," then you have DUN installed and are ready to go!
   c. If not, then you will have to install DUN.
      i. Click on Start>Settings>Control Panel>Add/Remove Programs
      ii. Click on the tab for "Windows Setup"
      iii. The computer will search for settings. Then select "Communications."
      iv. Select "Dial-up Networking."
      v. Select "Dial-up Server." This will allow you to receive calls.

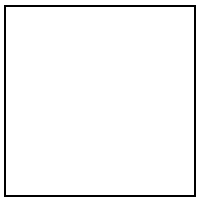


Figure 1—Select Dial-up Networking and Dial-up Server.

      vi. 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).

      vii. Click on "ok."
      viii. You may have to re-boot your computer.
2. Check to see if your computer has a modem and software installed.
   a. Click on Start>Settings>Control Panel>Modems. If you have one installed, then you should see one here. It may look like this:

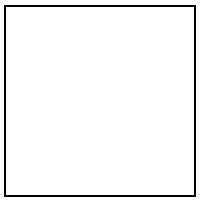


Figure 2—Laptop with PCMCIA modem card installed.

b. 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.
   i. Click on "start>settings>control panel>add new hardware
   ii. When the "add new hardware wizard" opens click on "next" twice.
   iii. Click on "no, I want to select the hardware from a list."
   iv. Then click on "next."
   v. Select "modem."
   vi. Click on "next."
   vii. 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.
   viii. Click on "next." The computer should find your modem on COM1.
   ix. If not, select "next."
   x. Select "have disk" and change to the CD-drive.
   xi. Select the modem. I selected Hayes V.90 PCI modem for my external one.
   xii. Select "next."
   xiii. Select "finish."

3. 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:
   a. Double-click on "my computer"
   b. Double-click on "dial-up networking"
   c. Click on "make a new connection"
   d. Give the connection a name (matt, scott, dave, etc)
   e. Select a modem to use
   f. Click on "next"
   g. Put in the phone number to call…In our example if I was configuring "matt" to call "dave" then I would use 555-6003.
   h. Select a country or region code (US)

(Step 2) *Configure the communication rules*

4. Configure the communication rules ("protocols"):
   a. On that connection we just made in the dial-up networking folder, right-click it
   b. Select "properties"

c. Make optional selections in the next steps.

5. 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):
   a. 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.
   b. Scripting—allows us the option to use a modem script or another type of script for the dial-up access.
   c. Multilink—allows us the option of using multi-link for connections.

Server Types Tab:

6. 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.
7. You can select any "advanced options."
   a. 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.
   b. Enable software compression—If your ISP requires use of compression technologies (most do not) then select this.
   c. 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.
   d. Require data encryption—Ditto..this just encrypts the data.
   e. 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.
8. 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:

9. 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:

10. 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

11. In the dial-up networking window select "connections" from the pull-down menu and then "Dial-up server."
12. Select "allow caller access."

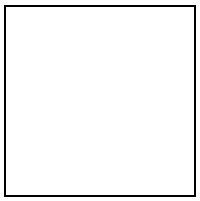


Figure 3—Setting up Dial-up Server.

13. Put a password in if you want.
14. Click on "server type."
15. Select "Type of Dial-up Server."
16. Select "PPP: Internet, Windows NT Server, Windows 98."
17. Disable "Require encrypted password" if none will be used.
18. 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.
   a. 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
   b. 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.

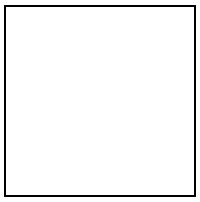


Figure 4—Selecting the TCP/IP for the Dial-up Adapter

c. 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:
    d. Open "my computer."
    e. Double-click on "dial up networking" folder
    f. Double-click on "make a new connection."
    g. Give a name to the connection
    h. Select modem to use
    i. Click on "next."
    j. Put in the phone number.
    k. Click on "next."
    l. Click on "finish."
    m. 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.

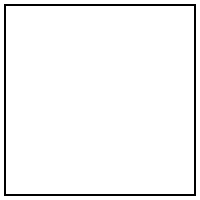


Figure 5a—Dialing to connect to Dave from Matt via dial-up networking.

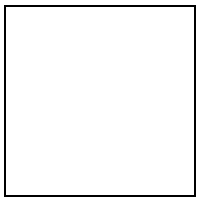


Figure 5b—Verifying user name and password (none) to connect to Dave from Matt.

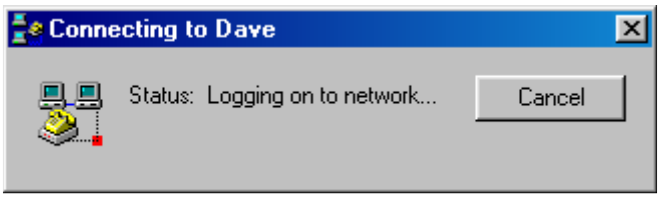


Figure 5c—Logging on to the network to connect to Dave from Matt.

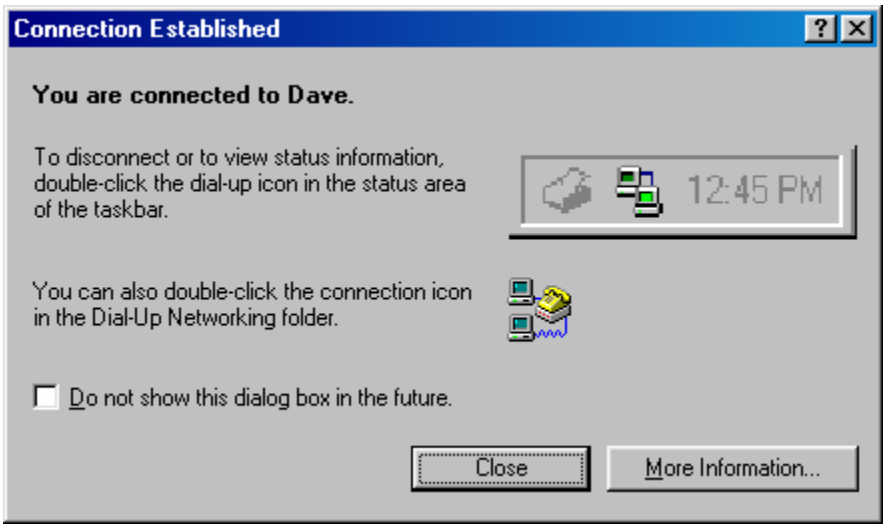


Figure 5d—Isn't that nice?

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."

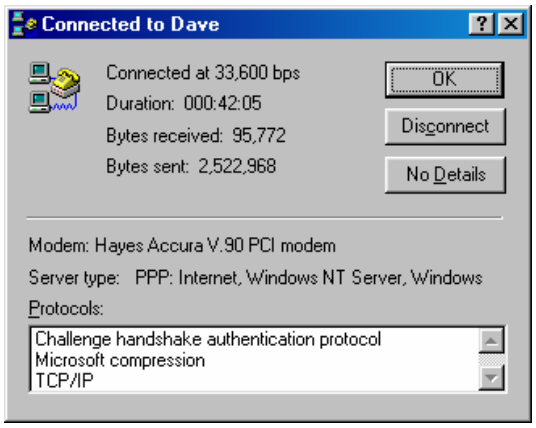


Figure 6—Active connection with DUN.

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.
19. Have Dave establish a DUN to Matt.
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.

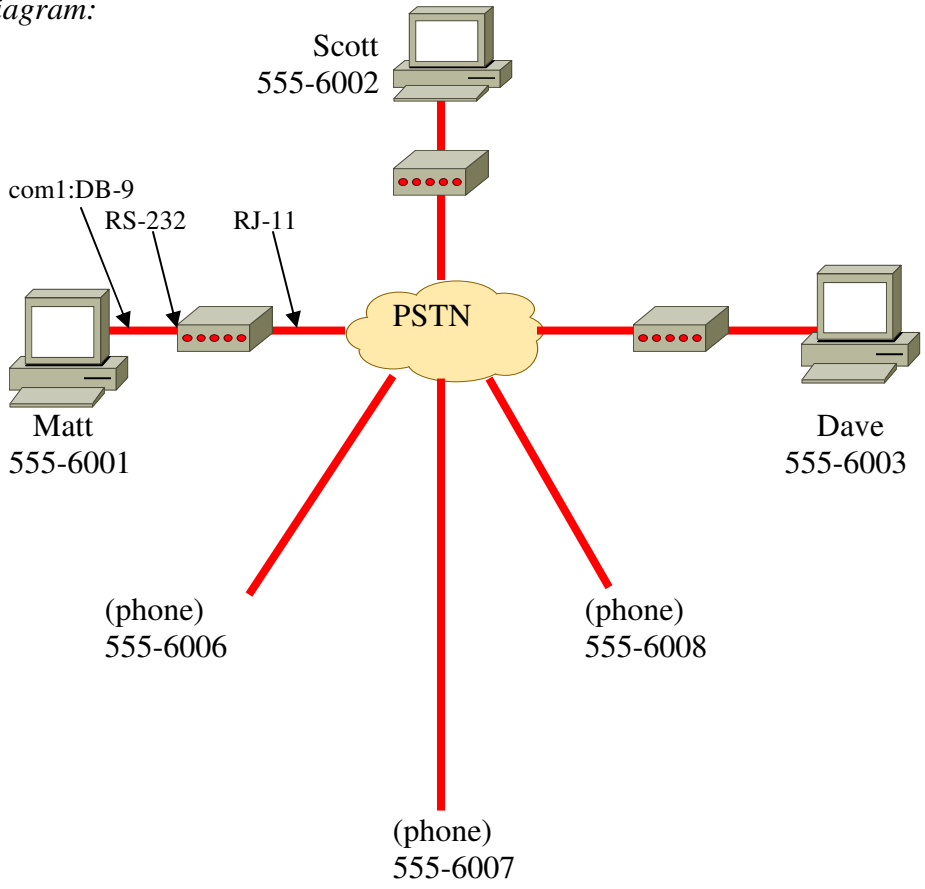
Supplemental Lab or Challenge Activities:

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!

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"	*	"asterisk"
^	"carat"	-	"hyphen"
$	"dollar sign"	:	"colon"
%	"percent sign"	@	"character command set"
&	"ampersand"	\	"backslash"
)	"parenthesis"	#	"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.

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

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:

TxD	transmit data	DSR	Data Set Ready
RxD	receive data	GRD	Signal ground
RTS	request to send	CD	Carrier detect
CTS	clear to send	DTR	Data terminal ready

You have also seen "blinking lights" on an external modem (if you used the external type). On my Hayes here is what those lights mean:

HS	High-speed	Lights when communicating at more than 4800kbps
RI	Ring Indicate	Blinks on and off when detecting incoming ring
CD	Carrier Detect	Lights when the DCD signal from the fax modem to the computer is on
OH	Off Hook	Lights when the fax modem is off hook
RD	Receive Data	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."
SD	Send Data	Flashes whenever data or commands are transmitted from the serial port of your computer or other device to the fax modem.
TR	Terminal Ready	Lights when the computer is ready to send or receive data. Indicates the status of the DTR signal from the terminal or computer.
MR	Modem Ready	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.

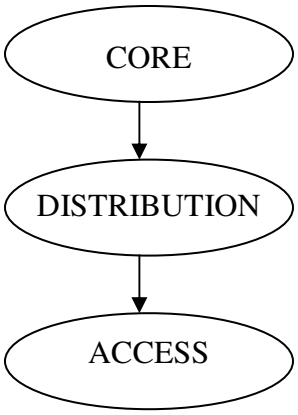
Part 2:
Basic Routing I

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.



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.

CORE
6500 switches
8500 switches
7000 routers
10000 routers
12000 routers
DISTRIBUTION
4000 switches
5000 switches
6000 switches
3600 routers
4000 routers
ACCESS
700 routers
800 routers
1700 routers
2500 routers
2600 routers
1900 switches
2820 switches
2900 switches

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.

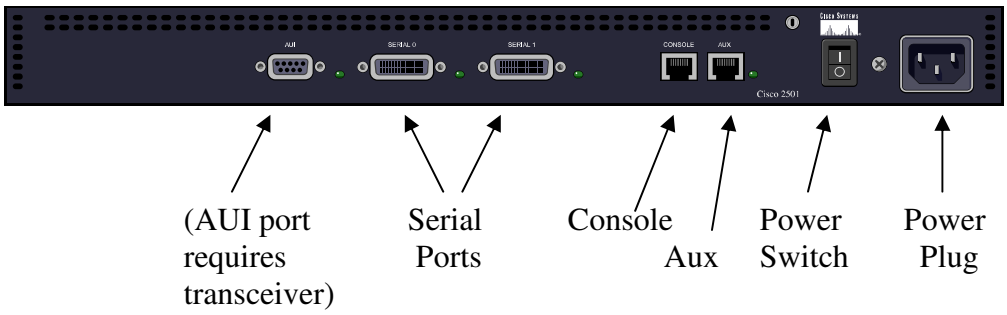


Figure 2—CISCO 2501 router "rear" view, dual serial, single AUX.

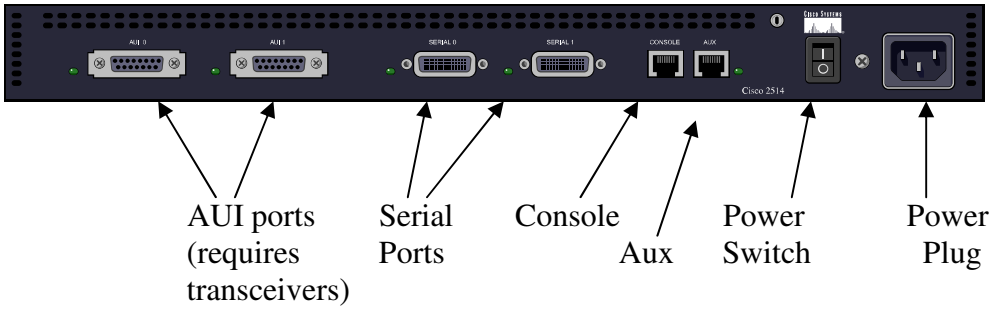


Figure 3—CISCO 2514 router "rear" view, dual serial, dual AUX.

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.

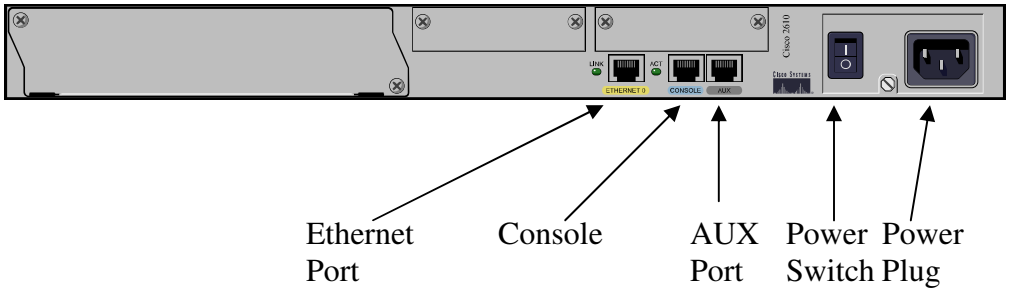


Figure 4—CISCO 2610 router “rear” view, single Ethernet, no serial.

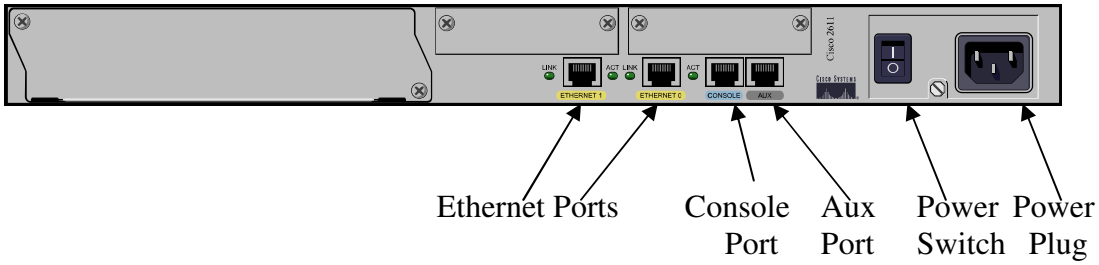


Figure 5—CISCO 2611 router “rear” view, dual Ethernet, no serial.

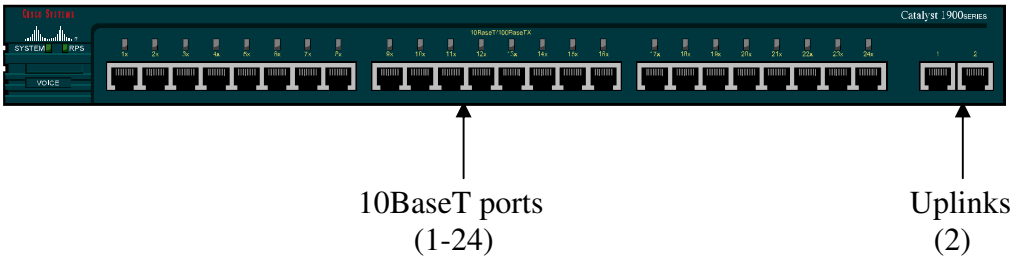


Figure 6—CISCO 1924 switch “front” view, 24-port switch (10Base T ports with 2 uplinks).

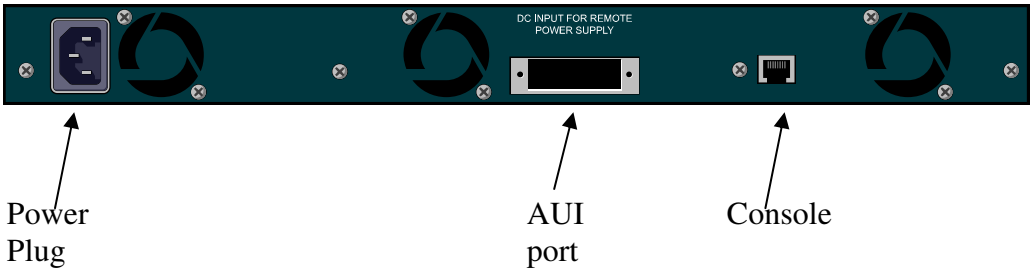


Figure 7—CISCO 1924 switch “rear” view, 24-port switch (10Base T ports with 2 uplinks)—same on 2924.

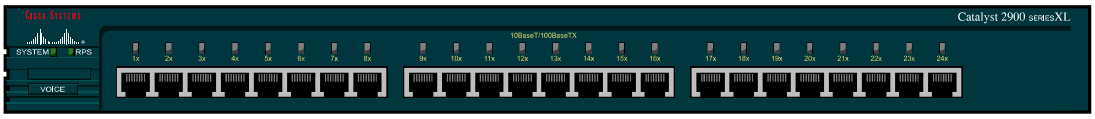


Figure 8—CISCO 2924 switch “front” view, 24-port switch (100 Base T ports—all ports capable of being uplinks).

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
3. Catalyst 1900/2820 Enterprise Edition Software Configuration Guide
4. Catalyst 2900 User Guide

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

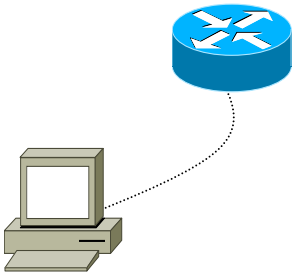
So What Have I Learned Here?
In this lab you have been introduced to the CISCO hierarchical model. We won’t be doing too much with this here in the CCNA course but if you want to learn about the design stuff (CCDA) plan on seeing it in your sleep. We also have a lab on it again in Part 3. This is a nifty overview of the routers and switches that you may encounter during your CCNA studies.

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 new 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:



Step-by-Step Instructions:

1. In the lab design above fill in the types of cables used (xo, ro, st) and into which port they will be inserted.
2. Cable the lab as shown.
3. Open the hyperterminal session on the workstation.
4. Turn the power on to the router and watch the text as the router boots. In a couple of labs you will learn the sequencing and purpose for all of that information. Finally the router will prompt you with the message:

   --- System Configuration Dialog ---

   Would you like to enter the initial configuration dialog? [yes/no]:

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]

   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. 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.
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."
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.       (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.

Shortcut	**Description**
Control+a	
Control+b	
Escape+b	
Control+e	
Control+f	
Escape+f	
Control+n	
Control+p	
Tab	Completes the entry

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 change 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

Command	**Description**

Privileged Model Options

Command	Description

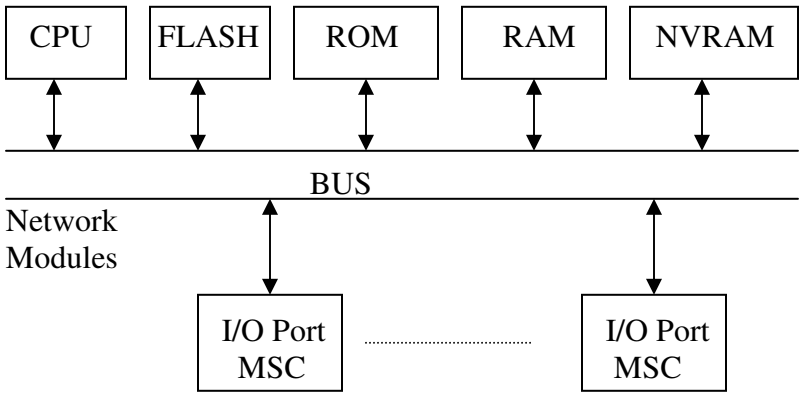
Router Boot Sequence

Objectives:
To more fully understand how routers hardware and software work together.

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

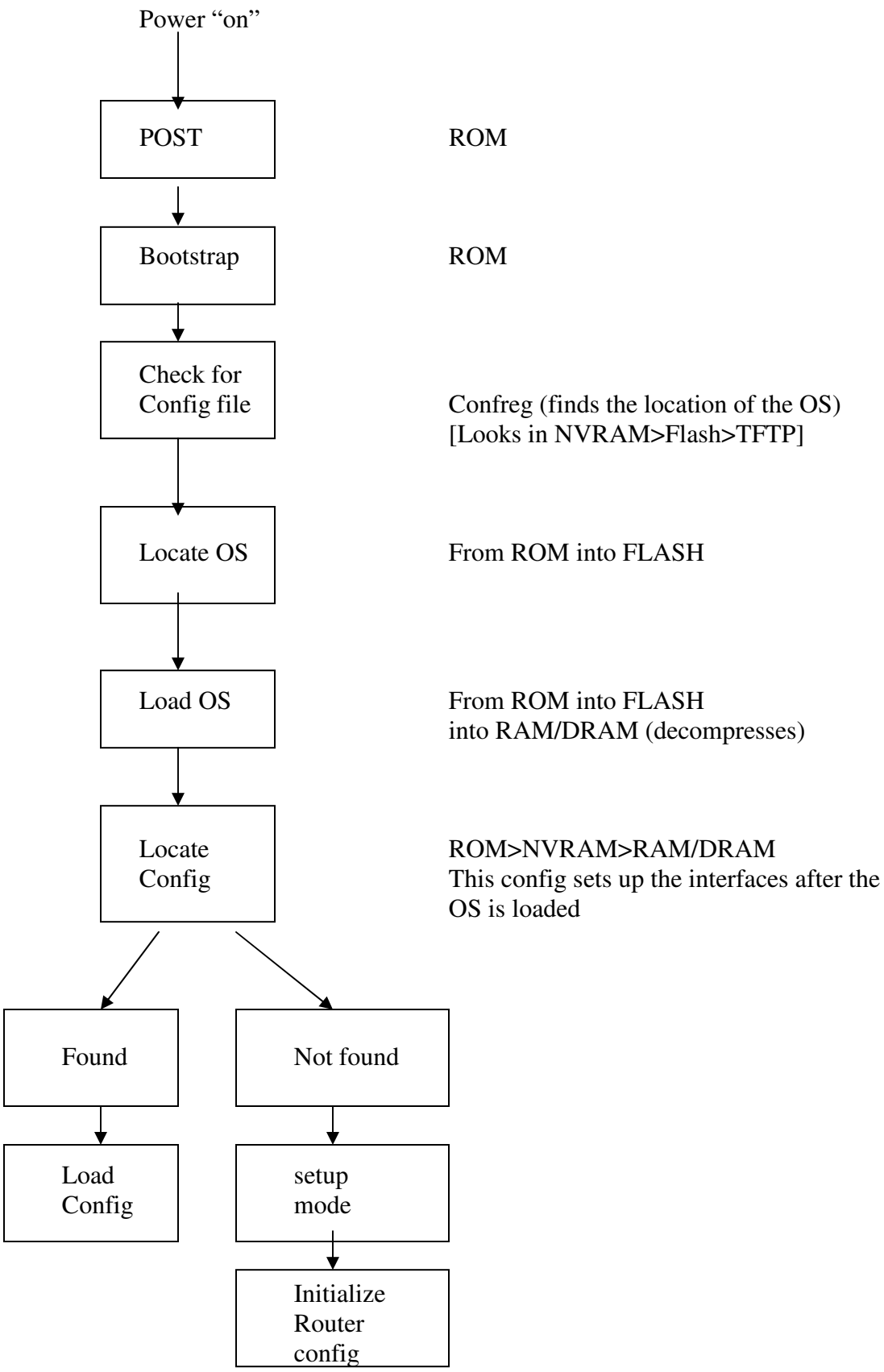
Background:
All routers essentially have 5 types of logic processors: CPU, FLASH, ROM, RAM, and NVRAM. Finding out information about these devices from CISCO or on the Internet is problematic, to say the least. Here we will discuss the block diagram of generic routers, how to identify the components 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-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
ROM
Bootstrap
ROM
Check for Config file
Confreg (finds the location of the OS)
[Looks in NVRAM>Flash>TFTP]
Locate OS
From ROM into FLASH
Load OS
From ROM into FLASH
into RAM/DRAM (decompresses)
Locate Config
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.)*

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.

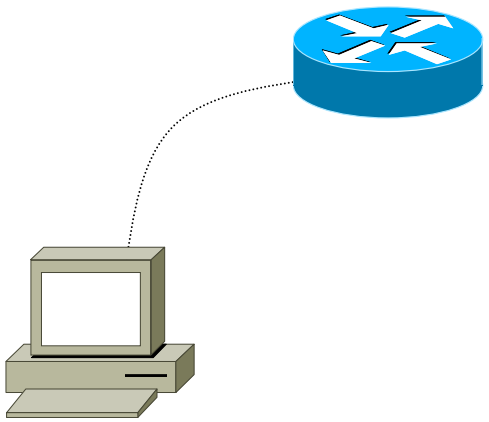
--- System Configuration Dialog ---

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?

Command	description	processor
sh buf (show buffers)		
sh fla (show flash)		
sh int (show interface)		
sh mem (show memory)		
sh pro (show processes)		
sh prot (show protocols)		
sh ru (show run)		
sh start (show start)		
sh stacks (show stacks)		
sh tech (show tech)		
sh ver (show version)		

3. Let's look at a basic router script (use show run). My comments are in h*andwriting:*

```
Router#sh ru
Building configuration...

Current configuration:
!
version 12.0                          Our IOS version
service timestamps debug uptime       enables time stamping[1]
service timestamps log uptime         enables time stamping[1]
no service password-encryption        disables service passencryption[2]
!
hostname Router                       Our router name
!
memory-size iomem 15                  sets memory i/o to 15[2]
ip subnet-zero                        Let's us use subnet zero!

interface Ethernet0/0                 interface
 no ip address                        ip address for this interface
 no ip directed-broadcast             drops ip directed broadcasts[3] (good…prevents
                                      Denial of Service attacks)
 shutdown                             Note: shut down by default
!
interface Serial0/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache                   Disables fast switching of IP packets[4]
 shutdown
!
interface Serial0/1
 no ip address
 no ip directed-broadcast
 shutdown
!
```

```
ip classless                    Disables classless routing behavior[5]
!
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 numbered[1-5]. 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.

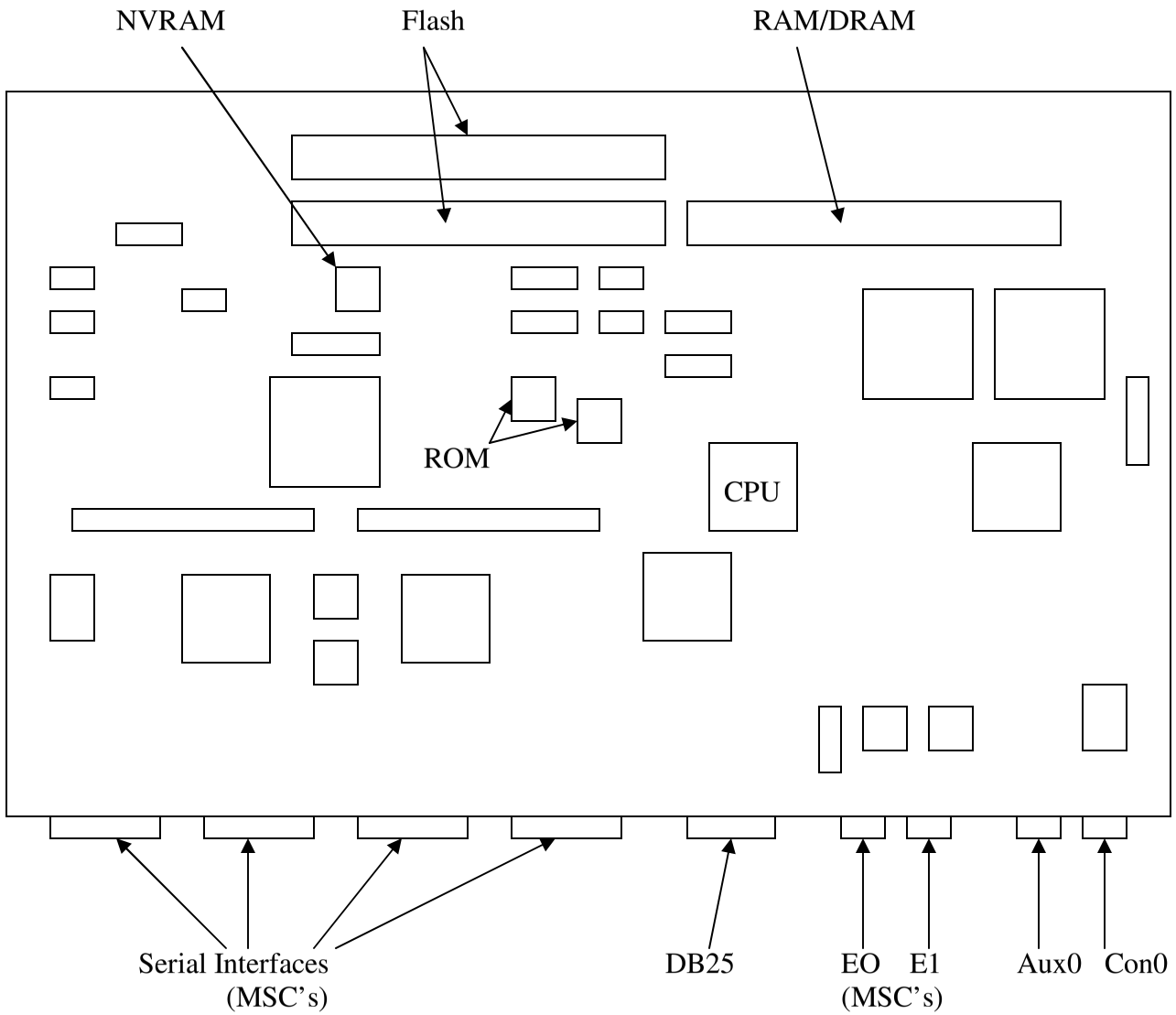
So What Have I Learned Here?

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.

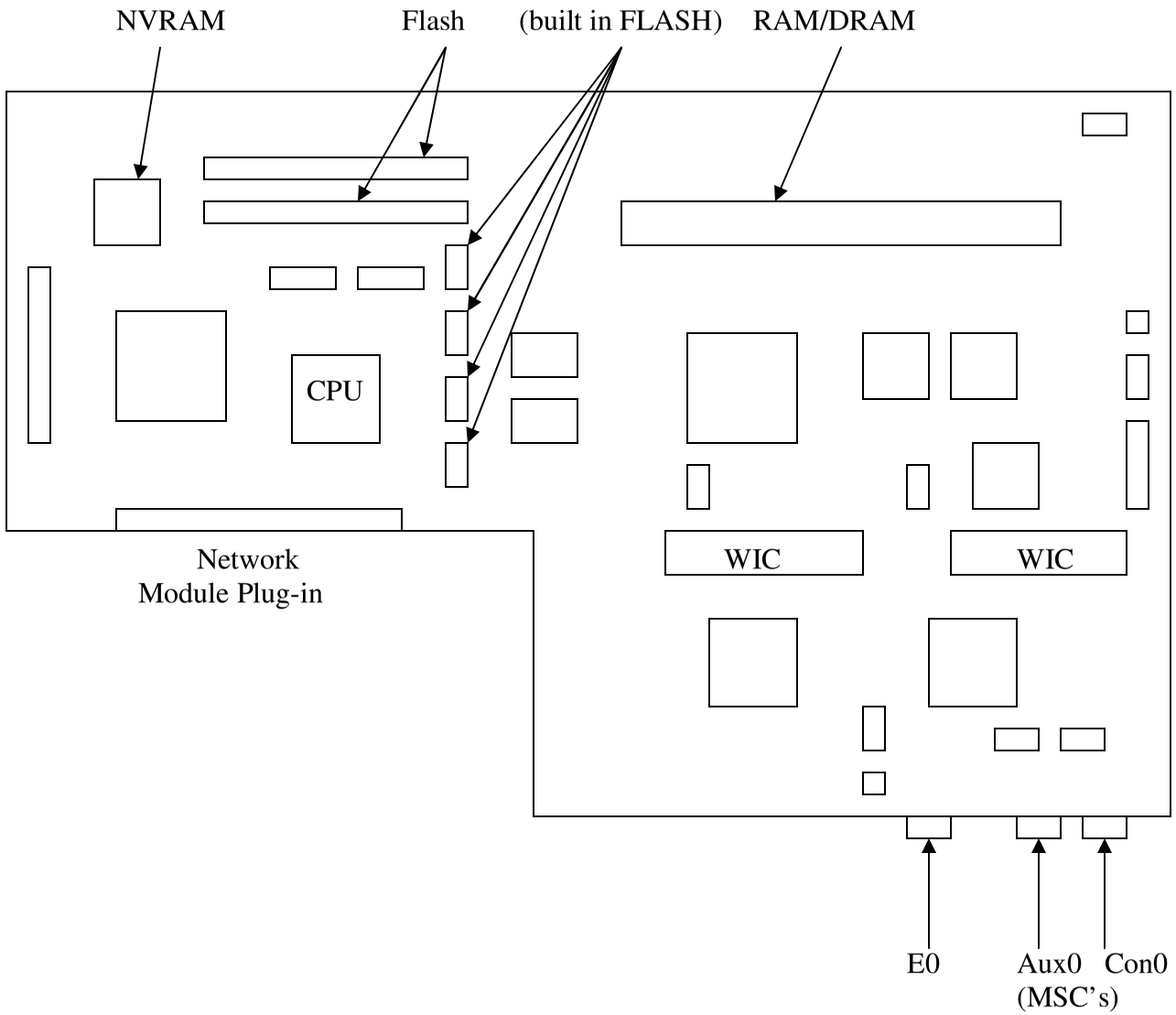
Bollapragada, Vijay, Murphy, C. and White, R. (2000). Inside CISCO IOS Software Architecture. Indianapolis, IND: CISCO Press. ISBN: 1-57870-181-3.

Coulibaly, M. (2000). "Chapter 8: The Hardware-Software Relationship" in CISCO IOS Releases: The Complete Reference. Indianapolis, IND: CISCO Press. ISBN 1-57870-179-1.

Held, Gil (2000). CISCO Router Performance: Field Guide. New York: McGraw-Hill. ISBN: 0-07-212513-6.



CISCO 2500 series Router Motherboard Configuration



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.

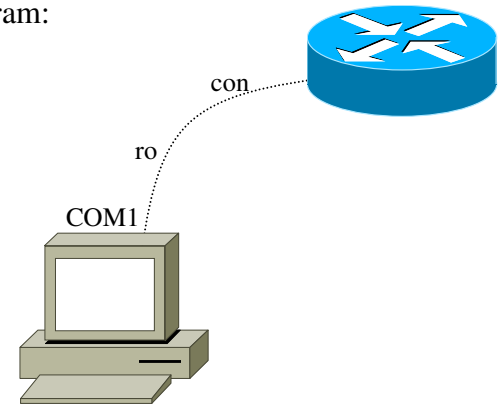
Background:
During the course of your CCNA studies you will be setting up many routers with many different router configurations. It is a good idea to learn to set up routers in "steps."

Step 1—start with setting up basic router configuration.
Step 2—configure interfaces
Step 3—configure routing protocol
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:

Router>**enable**	(or just "**en**")
Router#	since no enable password is set yet, the router does not ask for a password

2. Enter configuration mode:

Router#**configure**	(or just "**config t**")
Router#**terminal**	
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." These are required to have telnet access into your router. If you do not want anyone to be able to telnet into your router, then not setting a password is one way to do it.

   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 example, 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.

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.

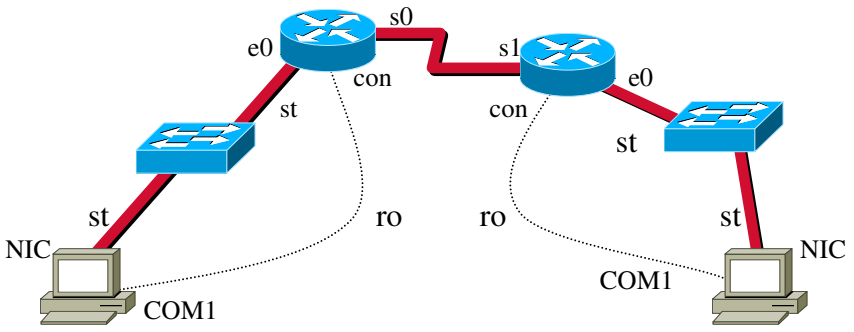
Basic Rip

Objectives:
To learn about the Routing Information Protocol (RIP version 1).

Background:
http://www.cisco.com/pcgi-bin/Support/PSP/psp_view.pl?p=Internetworking:RIP
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/rip.htm

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:



Workstation "A" Workstation "B"

Addressing:

Routers		
Hostnames	Randy	Ward
E0	192.168.3.1/24	192.168.4.1/24
S0	192.168.30.1/24 (DCE)	n/a
S1	n/a	192.168.30.2/24 (DTE)

Workstations	A	B
IP	192.168.3.2	192.168.4.2
SM	255.255.255.0	255.255.255.0
GW	192.168.3.1	192.168.4.1

Step-by-Step Instructions:

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. Complete the basic router setup 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. 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**

4. Configure the routing protocol and advertise/associate/publish 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>
   ```

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).

   Randy>**sh cdp neighbors**

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    Capability  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.

10. Let's see if we have any CDP traffic being generated. CDP updates every 60 seconds by default.

    Randy>**sh cdp traffic**

    You should see:

    ```
    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.

11. Let's use the protocols command to see what we have.

    Randy>**sh protocols**

    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.

12. Let's look at our path or "route" from one router to another:

    Randy>**sh ip route**

    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."

13. 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)

You should see: {This is what I sent}

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.

14. 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).

    Randy#**debug ip rip**

    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#

15. 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

16. 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.

History of BBS: http://www.zdnet.com/pcmag/issues/1414/pcm00161.htm

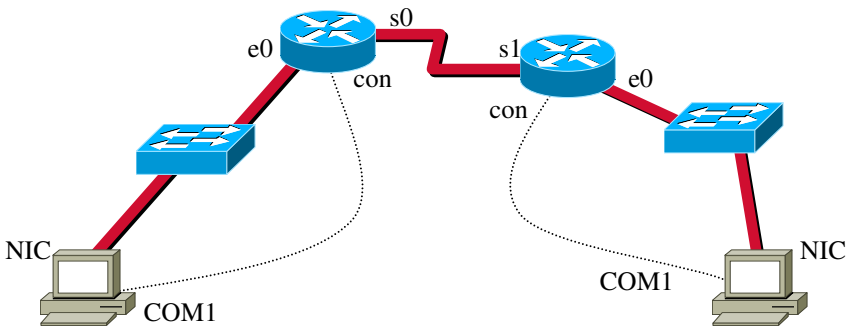
Basic Troubleshooting: Router-to-Router

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:



Workstation "A" Workstation "B"

Addressing:

Routers		
Hostnames	Randy	Ward
E0	192.168.3.1/24	192.168.4.1/24
S0	192.168.30.1/24 (DCE)	n/a
S1	n/a	192.168.30.2/24 (DTE)

Workstations	A	B
IP	192.168.3.2	192.168.4.2
SM	255.255.255.0	255.255.255.0
GW	192.168.3.1	192.168.4.1

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.

Line	Protocol	what it means
UP	UP	everything is fine.
UP	DOWN	connection problems (check your cabling)
DOWN	DOWN	interface problems
AD. DOWN	DOWN	disabled…everything is wrong.

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
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)
Conversations 0/0/256 (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
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 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
DCD=down DSR=down DTR=down RTS=down CTS=down

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):

Response	Description
!	Successful
.	Timed out
U	Destination unreachable
&	Packet Time to Live (TTL) exceeded
?	Packet type unknown
C	Congestion experienced during transit
I	Interruption of Ping packet

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:

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
Randy#

Once you find out if a destination is unreachable you can use the trace route command to "pin-point" where the problem may be:

Randy#**tracertoute 192.168.4.1.**

You should see:

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 likely to least likely response):

Response	Description
*	Timed out
U	Port was unreachable
N	Network was unreachable
P	Protocol is unreachable
!H	Received but not forwarded…ACL is set

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
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#
```

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]
Randy#
```

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?

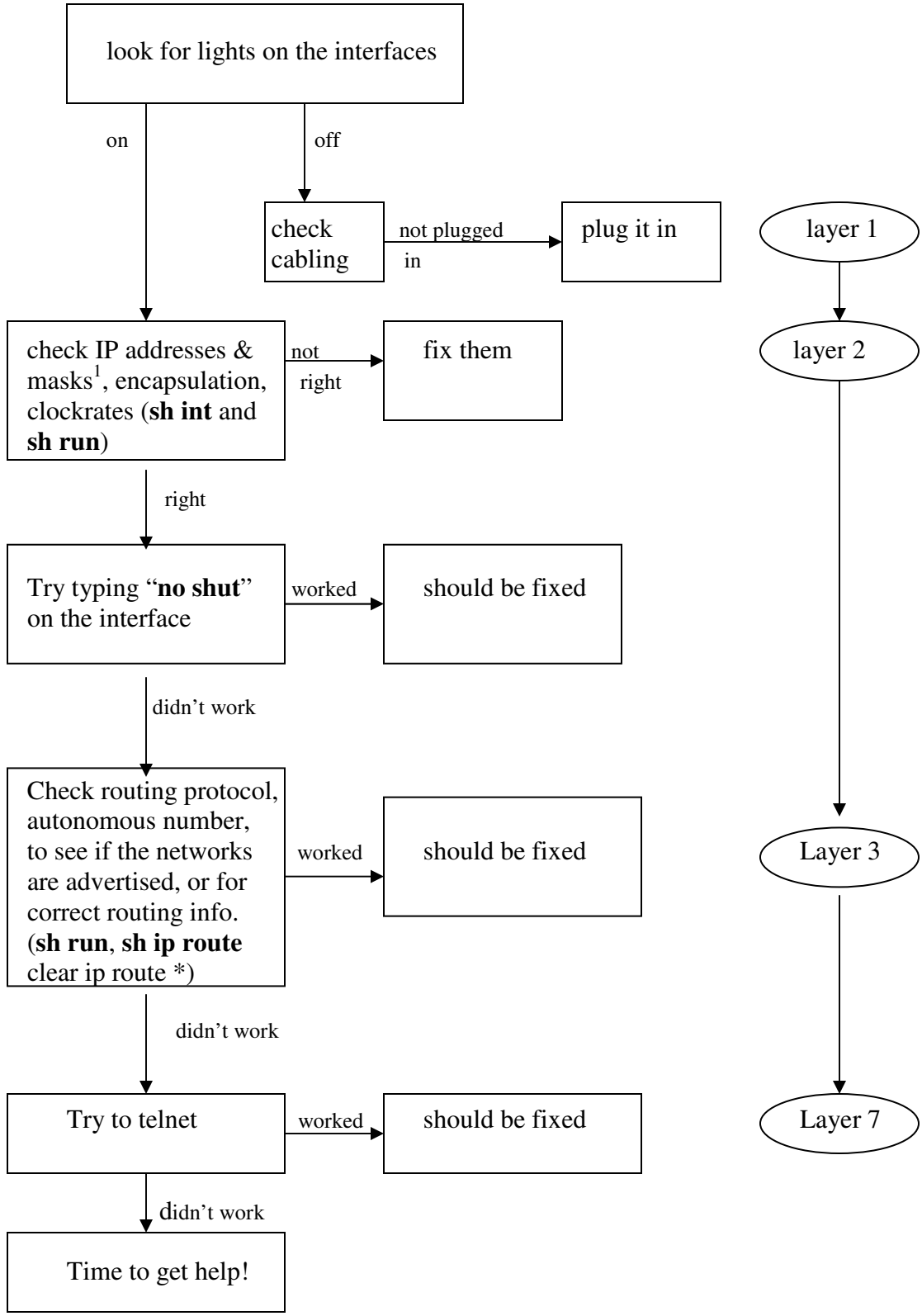
In this lab you learned the basics of troubleshooting from one router to the other. As you move up in your studies you will learn more precise troubleshooting methods. I cannot tell you how many times a student told me their network didn't work and all that was wrong was an unplugged cable. Keep it simple 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.

History of BBS: http://www.zdnet.com/pcmag/issues/1414/pcm00161.htm

BASIC TROUBLESHOOTING—RIP



[1] I know IP addressing is a layer 3 function, but it uses a layer 2 command **sh int** to view its status.

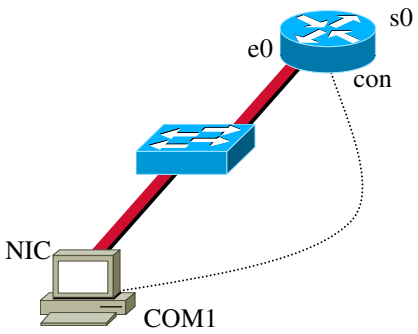
Loopback Interfaces

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
 LADRF=0x0020 0x0100 0x0000 0x0000
 CSR0  =0x00000072, CSR3  =0x00001044, CSR4  =0x0000491D,
CSR15 =0x00000000
 CSR80 =0x0000D900, CSR114=0x00000001, CRDA  =0x01D175C0,
CXDA  =0x01D17A20
 HW filtering information:
  Promiscuous Mode Disabled, PHY Addr Enabled, Broadcast Addr
Enabled
  PHY Addr=0002.FD45.AE60, Multicast Filter=0x0020 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  0100.5e00.0009  0000.0000.0000         0
  0xC0:  0  0100.0ccc.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 (ie., 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 its good practice to always "no shut" anytime you are configuring an interface. Exit from the configuration mode and ping from the router to the looback 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 looback interface from your workstation:

```
Loopback 1    11.11.11.11/24
Loopback 2    22.22.22.22/24
Loopback 3    33.33.33.33/24
Loopback 4    44.44.44.44/24
```

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>

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.

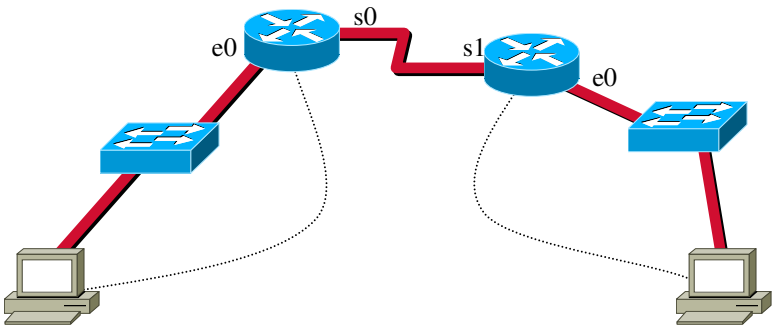
Basic RIP with Protocol Inspector

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
(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. See the lab in part 1 for downloading instructions if you have not done so already.

Lab Diagram:



Workstation "A" with protocol inspector

Workstation "B" with protocol inspector

Addressing:

Routers		
Hostnames	Emmanuel	Goldstein
E0	10.1.3.1/24	10.1.4.1/24
S0	10.1.192.1/24 (DCE)	n/a
S1	n/a	10.1.192.2/24 (DTE)
Workstations	A	B
IP	10.1.3.2	10.1.4.2
SM	255.255.255.0	255.255.255.0
GW	10.1.3.1	10.1.4.1

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.
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.

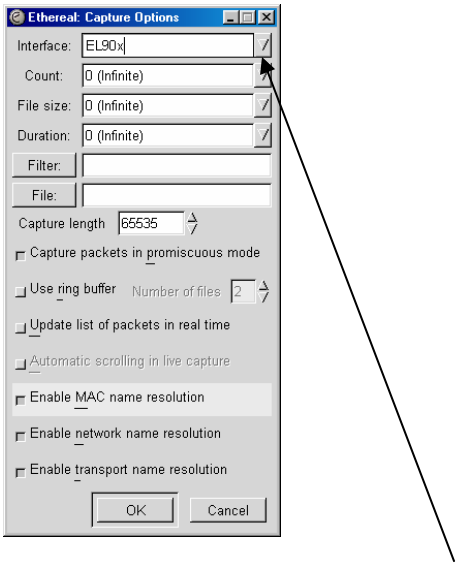


Figure 1—Be sure to select the NIC as your capture interface.

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.

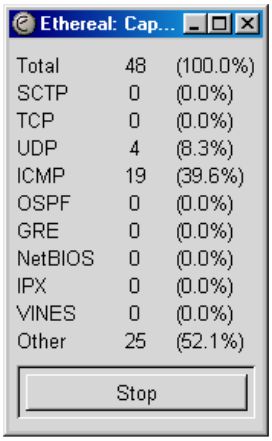


Figure 2—A capture in progress.

5. Wait two minutes. 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):

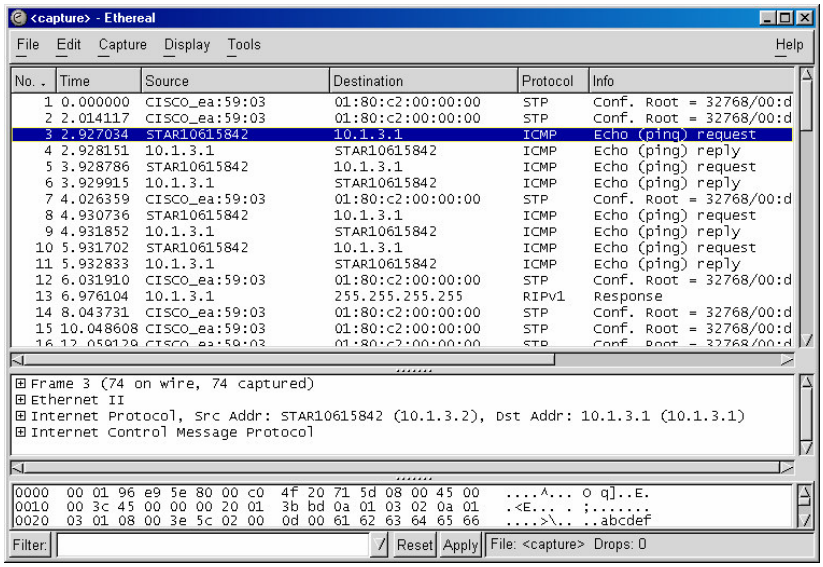


Figure 3—Here we can see what we were expecting…RIP and ICMP. And those STP packets? You will learn about them in part 3.

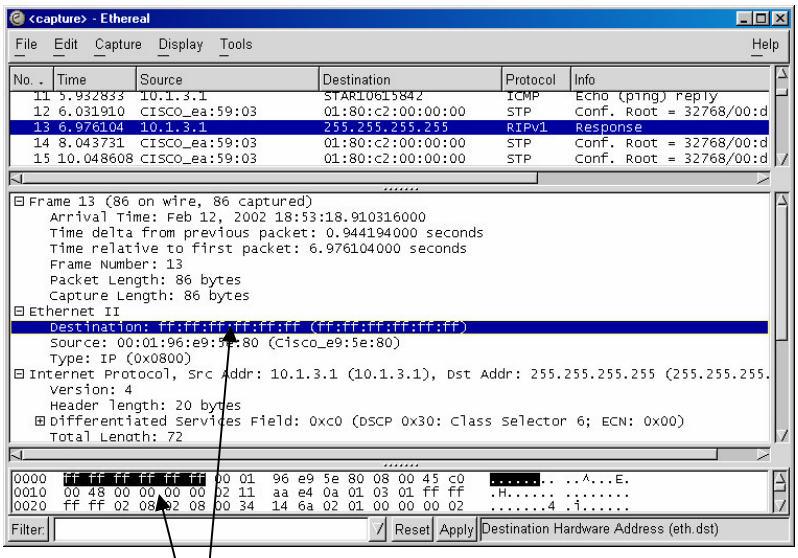


Figure 4—We can even look down to the HEX information (bit-by-bit) at our RIP update captures. Notice how our RIP Operation is a "broadcast" (FF FF FF FF FF FF) on the network.

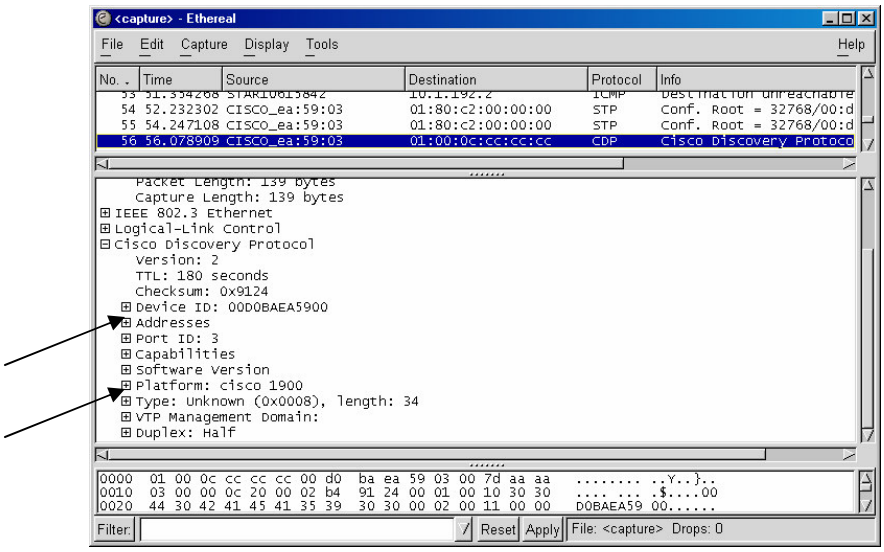


Figure 5—We can even see CDP packets with our protocol. Notice how we see those CDP characteristics built into the frame: identification (device ID), address, and platform.

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.

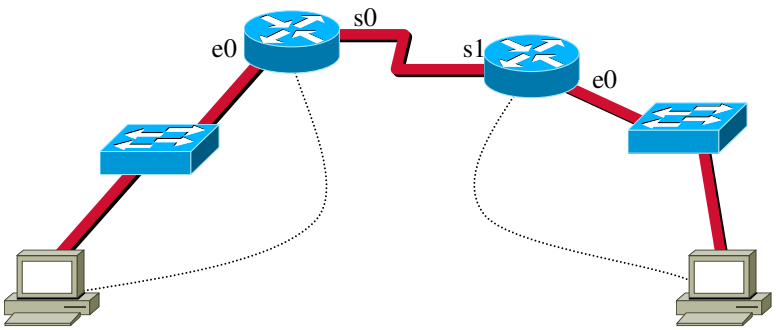
Router Telnet Lab

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:



Workstation "A" Workstation "B"

Addressing:

Routers		
Hostnames	William	Gibson
E0	180.11.3.1/24	180.11.4.1/24
S0	180.11.12.1/24 (DTE)	n/a
S1	n/a	180.11.12.2/24 (DCE)
Workstations	A	B
IP	180.11.3.2	180.11.4.2
SM	255.255.255.0	255.255.255.0
GW	180.11.3.1	180.11.4.1

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
00:36:59: TCP66: Telnet sent WILL SUPPRESS-GA (3)
00:36:59: Telnet66: 80000 80000 253 24
00:36:59: TCP66: Telnet sent DO TTY-TYPE (24)
00:36:59: Telnet66: 10000000 10000000 253 31
00:36:59: TCP66: Telnet sent DO WINDOW-SIZE (31)
00:36:59: TCP66: Telnet received DO SUPPRESS-GA (3)
00:36:59: TCP66: Telnet received WILL TTY-SPEED (32) (refused)
00:36:59: TCP66: Telnet sent DONT TTY-SPEED (32)
00:36:59: TCP66: Telnet received WILL WINDOW-SIZE (31)
00:36:59: TCP66: Telnet received WILL LOCAL-FLOW (33)
00:36:59: TCP66: Telnet sent DO LOCAL-FLOW (33)
00:36:59: Telnet66: Sent SB 33 0
00:36:59: TCP66: Telnet received DO ECHO (1)
00:36:59: TCP66: Telnet received WONT TTY-TYPE (24)
00:36:59: TCP66: Telnet sent DONT TTY-TYPE (24)
```

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.

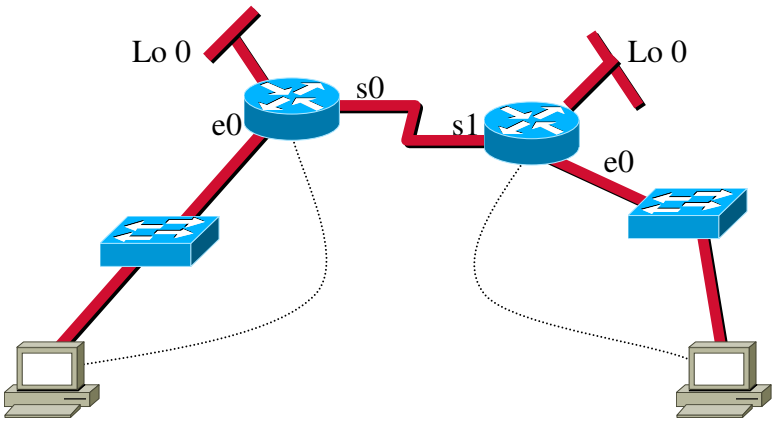
Route Summarization with RIP

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:



Workstation "A" Workstation "B"

Addressing:

Routers		
Hostnames	Phiber	Optik
S0	161.20.4.1/30 (DCE)	n/a
S1	n/a	161.20.4.2/30 (DTE)
L0	161.20.3.1/30	161.20.5.1/30
E0	161.20.2.1/24	161.20.1.1/24
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. 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 lets 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, Serial0/0
C       161.20.4.0/30 is directly connected, Serial0/0
C       161.20.3.0/30 is directly connected, Loopback0
C       161.20.2.0/24 is directly connected, Ethernet0/0
phiber#
```

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:

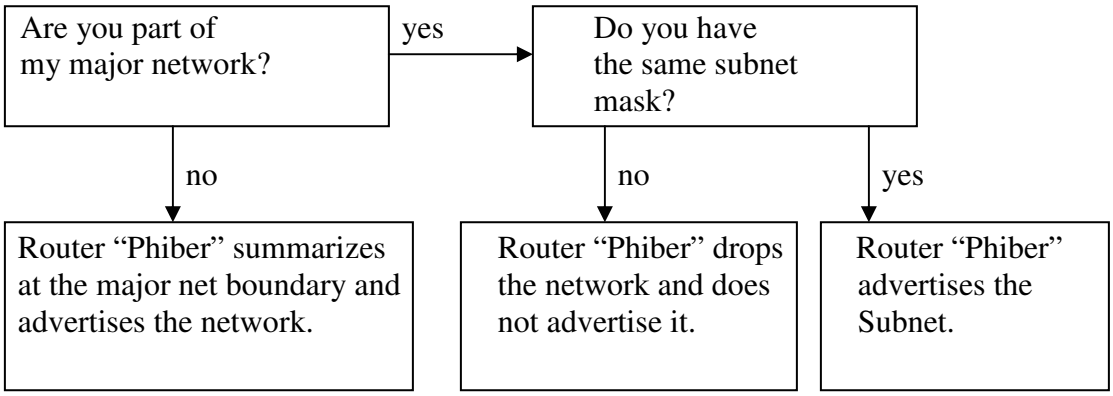
```
phiber#debug ip rip
RIP protocol debugging is on
phiber#
01:26:15: RIP: received v1 update from 161.20.4.2 on Serial0/0
01:26:15:      161.20.5.0 in 1 hops
01:26:19: RIP:sending v1 update to 255.255.255.255 via Ethernet0/0 (161.20.2.1)
 - suppressing null update
01:26:19: RIP: sending v1 update to 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:

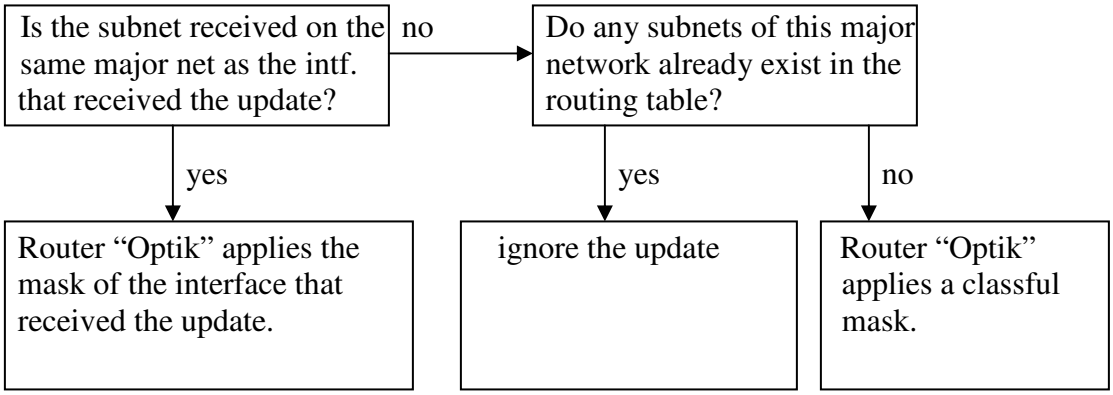
```
optik#debug ip rip
RIP protocol debugging is on
01:26:35: RIP: sending v1 update to 255.255.255.255 via Ethernet0/0
(161.20.1.1) - suppressing null update
01:26:35: RIP: sending v1 update to 255.255.255.255 via Serial0/1 (161.20.4.2)
01:26:35:      subnet  161.20.5.0, metric 1
01:26:35: RIP: sending 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):



How RIP *receives* updates (for example on Phiber to Optik):



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, and18 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 workstations where possible.
2. A good command to remember is to use is **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		
Hostnames	Phiber	Optik
S0	10.2.4.1/30 (DCE)	n/a
S1	n/a	10.2.4.2/30 (DTE)
E0	192.168.1.1/24	192.168.5.1/24
E1	192.168.2.1/24	192.168.4.1/24
Workstations	A-E0	B-E0
IP	192.168.1.2	192.168.5.2
SM	255.255.255.0	255.255.255.0
GW	192.168.1.1	192.168.5.1
Workstations	A-E1	B-E1
IP	192.168.2.2	192.168.4.2
SM	255.255.255.0	255.255.255.0
GW	192.168.2.1	192.168.4.1

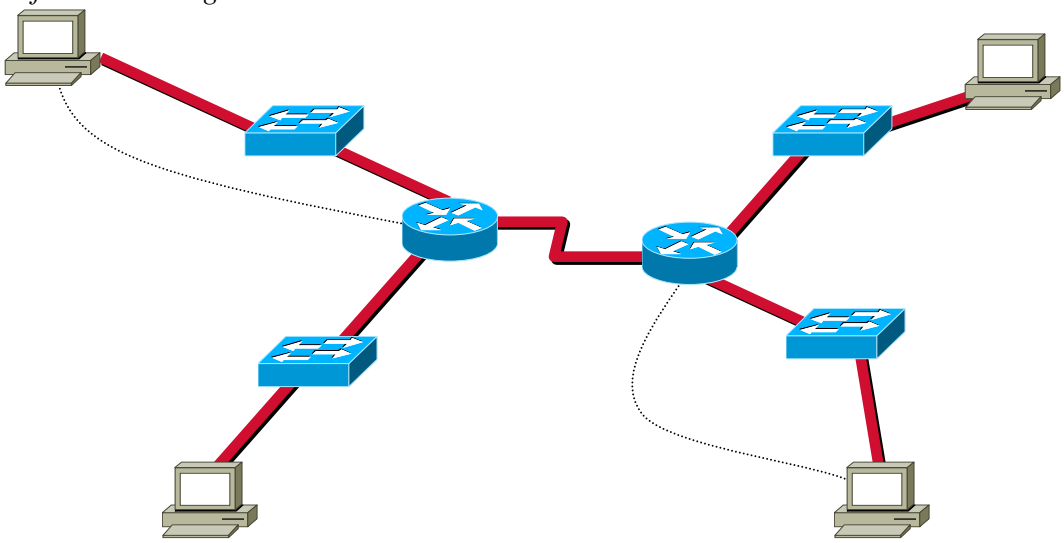
Scenario 2:

Routers		
Hostnames	Phiber	Optik
S0	10.2.4.1/24 (DCE)	n/a
S1	n/a	10.2.4.2/24 (DTE)
E0	192.168.1.1/24	192.168.5.1/24
E1	192.168.2.1/24	192.168.4.1/24
Workstations	A-E0	B-E0
IP	192.168.1.2	192.168.5.2
SM	255.255.255.0	255.255.255.0
GW	192.168.1.1	192.168.5.1
Workstations	A-E1	B-E1
IP	192.168.2.2	192.168.4.2
SM	255.255.255.0	255.255.255.0
GW	192.168.2.1	192.168.4.1

Scenario 3:

Hostnames	Phiber	Optik
S0	1.0.0.1/8 (DCE)	n/a
S1	n/a	1.0.0.2/8 (DTE)
E0	2.0.0.1/8	4.0.0.1/8
E1	3.0.0.1/8	5.0.0.1/8
Workstations	A-E0	B-E0
IP	2.0.0.2/8	4.0.0.2/8
SM	255.0.0.0	255.0.0.0
GW	2.0.0.1/8	4.0.0.1/8
Workstations	A-E1	B-E1
IP	3.0.0.1/8	5.0.0.2/8
SM	255.0.0.0	255.255.255.0
GW	3.0.0.1/8	5.0.0.1/8

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

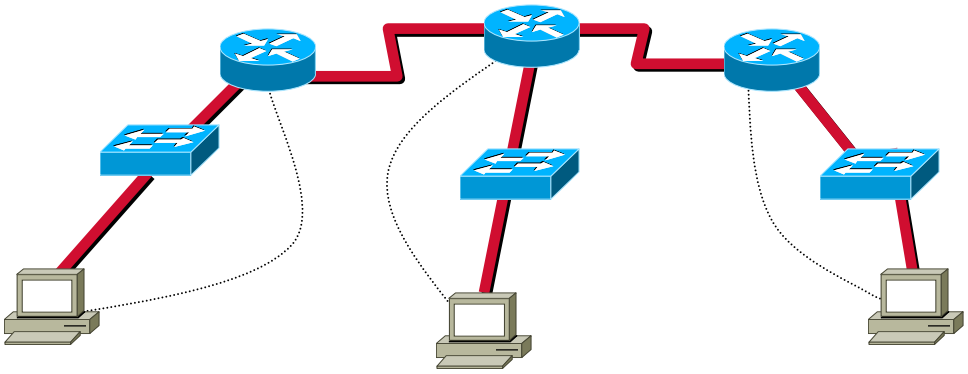
Phiber Optik was the leader of the Master's of Deception (MoD) hackers ring in New York City in the 1980's/early 1990's. Allegedly he master-minded the Martin Luther King day crash of AT&T's national phone service in 1990. Known for his daring actions and media stunts he appeared or was interviewed in 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:



Workstation "A" Workstation "B" Workstation "C"

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:

1. Cable 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 the router's networks.

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 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#
```

7. 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>
```

8. 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:

1. 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**

2. What would you expect to see on phreak? Try steps 1-6 over again on phreak.
3. Try this with class "B" private IP addresses that you choose.
4. Try this with class "A" private IP address that you choose.
5. 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.
6. Try this with class "C" public IP addresses that you choose.
7. Try this with class "B" public IP addresses that you choose.
8. Try this with class "A" public IP address that you choose.
9. Try mixing and matching private and public IP addresses.
10. 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.

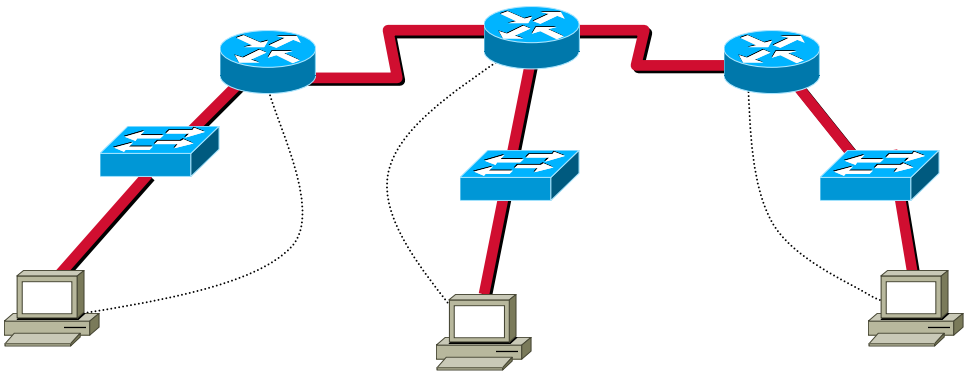
RIP metrics and the Limitations of RIP

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:



Workstation "A" Workstation "B" Workstation "C"

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:

1. Cable 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.
5. 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.
6. 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):

```
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
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.2.1.1              120      00:00:05
  Distance: (default is 120)
```

During:

```
scorpion#config t
scorpion(config-router)#timers basic 15 30 60 90
scorpion(config-router)#^Z
```

After (15 30 60 90:

```
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:
  Gateway         Distance      Last Update
  10.2.1.1              120      00:00:07
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?

Metric	Value
Hop count	
Update timer	
Invalid timer	
Hold-down timer	
Flush timer	

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.

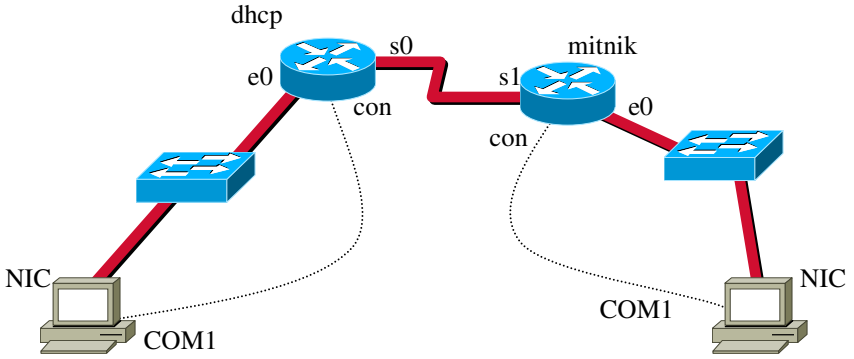
Dynamic Host Configuration Protocol (DHCP) Lab

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:



Workstation "A" Workstation "B"

Addressing:

Routers		
Hostnames	kevin	mitnik
E0	10.0.0.1/8	192.168.3.1/24
S0	192.168.1.2/24 (DCE)	n/a
S1	n/a	192.168.1.1/24 (DTE)
Workstations	A	B
IP	10.0.0.2	192.168.3.2
SM	255.0.0.0	255.255.255.0
GW	10.0.0.1	192.168.3.1

Step-by-Step Instructions:

1. Cable 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.
5. Setup the workstations with IP address, subnet masks, and gateways addresses. You will need to reboot the workstations.
6. Test connectivity from router to router.
7. Test connectivity from workstation A to workstation B from DOS.
8. Verify your RIP routes are being advertised.
9. Remove the IP address and gateway from workstation A and set it to obtain its address automatically. You will need to reboot.
10. 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
```

11. 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.
12. Test your connectivity between the two workstations.
13. Remove the IP address and gateway from workstation B and set it to obtain its address automatically. You will need to reboot.
14. 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
```

15. 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
```

16. Do a release and renew on workstation B's IP address.
17. 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 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

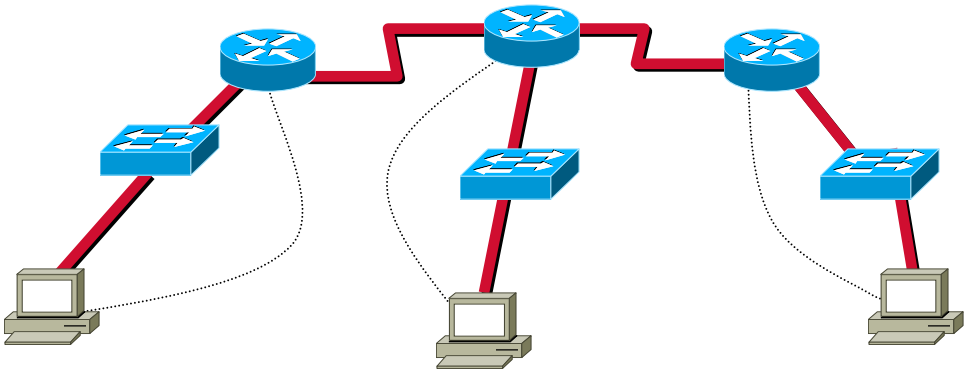
To *some*, Kevin Mitnik is an icon in the hacking community. In 1986 he was arrested for breaking into the Digital Equipment Corporation network. He was arrested in 1995 again for allegedly stealing 20,000 credit card numbers, but was actually convicted for illegal use of cellular numbers. He was a regular 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:



Workstation "A" Workstation "B" Workstation "C"

Addressing:

Routers				
Hostnames				
E0/0				
E0/1				
S0/0				
S0/1				
Workstations				
IP				
SM				
GW				

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.

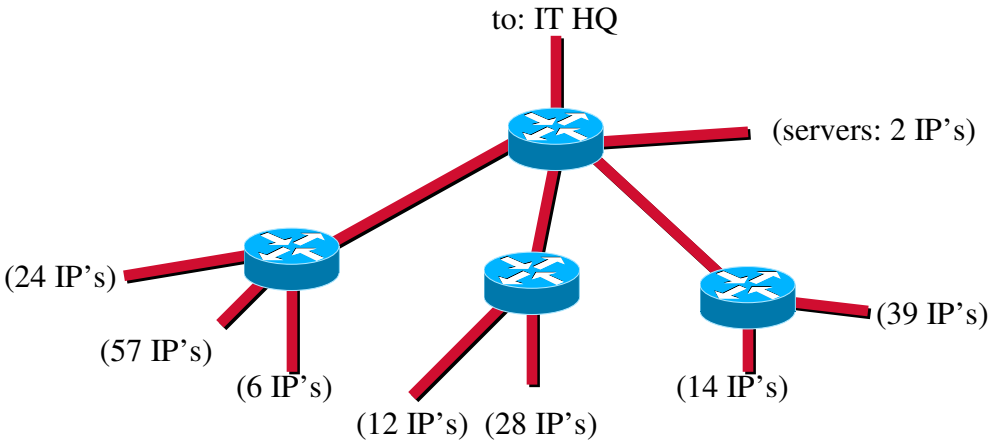
Paper Lab: Variable Length Subnet Masking (VLSM)

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 (http://www.henninger.net/downloads/ccna/tools/subnettable.pdf)

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**	212.14.17.224/**28**
212.14.17.64/**26**	212.14.17.8/**29**
212.14.17.128/**26**	212.14.17.4/**30**
212.14.17.32/**27**	212.14.17.240/**30**
212.14.17.192/**27**	212.14.17.244/**30**
212.14.17.16/**28**	212.14.17.248/**30**

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.

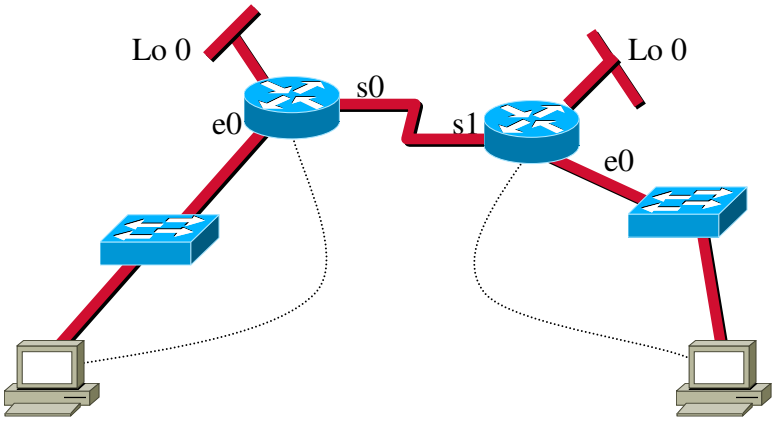
Static and Dynamic Routes with Discontiguous RIP Networks

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:



Workstation "A" Workstation "B"

Addressing:

Routers

Hostnames	Phiber	Optik
S0	161.20.4.1/30 (DCE)	n/a
S1	n/a	161.20.4.2/30 (DTE)
L0	161.20.3.1/30	161.20.5.1/30
E0	161.20.2.1/24	161.20.1.1/24

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 - 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
optik#
```

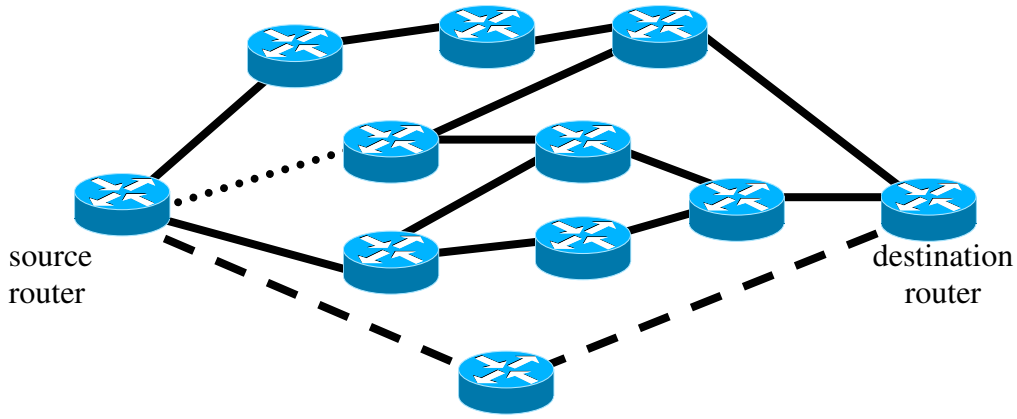
5. So let's fix that little problem here:

```
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 " – – – – " 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.

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 ?
  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!

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 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.
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

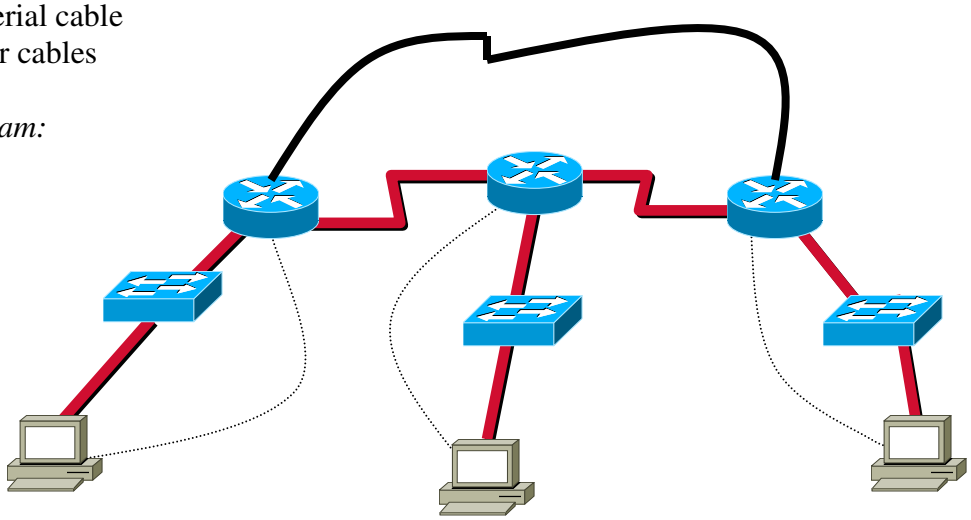
Phiber Optik was the leader of the Master's of Deception (MoD) hackers ring in New York City in the 1980's/early 1990's. Allegedly he master-minded the Martin Luther King day crash of AT&T's national phone service in 1990. Known for his daring actions and media stunts he appeared or was interviewed in many publications including Harper's, Esquire, and the New York Times. Don't worry…he got busted. Turk 182!

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:



Workstation "A" Workstation "B" Workstation "C"

Addressing:

Routers			
Hostnames	Prophet	Knight	Lightning
E0	172.16.1.1/16	172.16.2.1/16	172.16.3.1/16
S0	10.1.1.1/24 (DCE)	10.2.1.1/24 (DCE)	10.3.1.2/24 (DTE)
S1	10.3.1.1/24 (DTE)	10.1.1.2/24 (DTE)	10.2.1.2/24 (DTE)
Workstations	a	b	
IP	172.16.1.2/16	172.16.2.2/16	172.16.3.2/16
SM	255.255.0.0	255.255.0.0	255.255.0.0
GW	172.16.1.1	172.16.2.1	172.16.3.1

Step-by-Step Instructions:
1. Cable the lab as shown except for the serial line between Prophet and Lightning.
2. Complete the basic router setup on each router.
3. Configure the interfaces on each router.
4. Configure the routing protocol and advertise the router's networks.

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.
6. Test connectivity from workstation A to workstation C.
7. Turn on debug ip rip on each router.
8. 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 s0/0
prophet(config-if)#no ip split-horizon
```

9. 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 interface or just disconnect that serial line. This problem is known as "counting to infinity."

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.

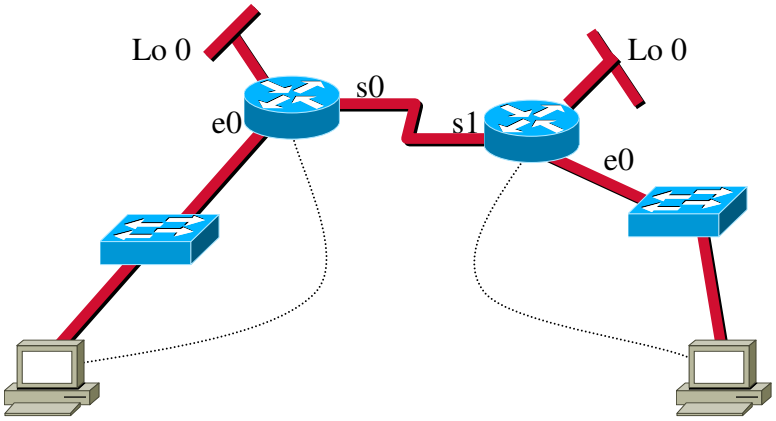
RIP Version 2 and Redistribution with RIP

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:



Workstation "A" Workstation "B"

Addressing:

Routers

Hostnames	Phiber	Optik
S0	161.20.4.1/30 (DCE)	n/a
S1	n/a	161.20.4.2/30 (DTE)
L0	161.20.3.1/30	161.20.5.1/30
E0	161.20.2.1/24	161.20.1.1/24

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 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
```

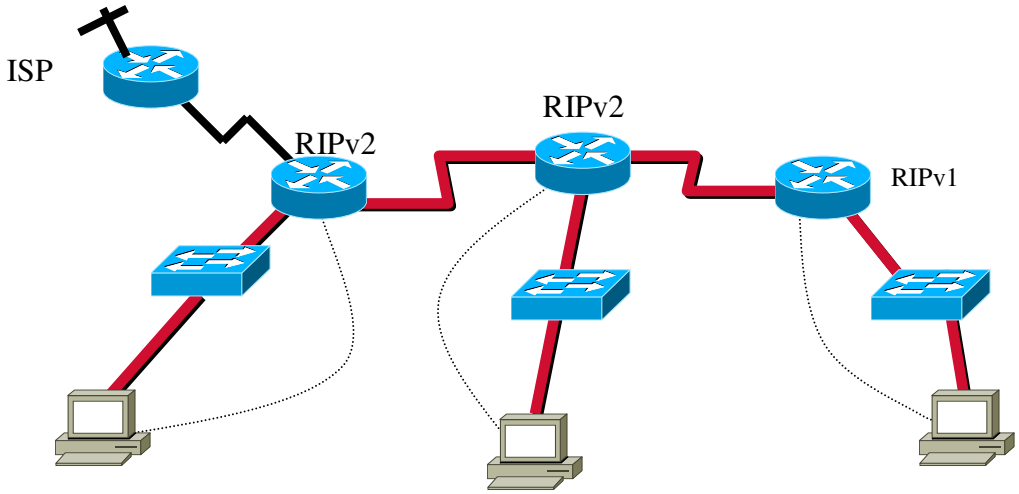
6. Now you should be able to ping from each workstation to the other workstation, to the loopbacks on both routers and everywhere. BAM! Problem solved much easier than with static routes. Yeah…it's that easy. ☺
7. Now let's take it up another level and add some more routers to our network (look for the 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.



Addressing:

Routers Hostnames	ISP	RIPv1
E0	n/a	192.168.1.1/24
L0	172.16.1.1/16	n/a
S0	220.221.222.253/30(DCE)	n/a
S1	n/a	161.20.6.2/24 (DTE)

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!

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.

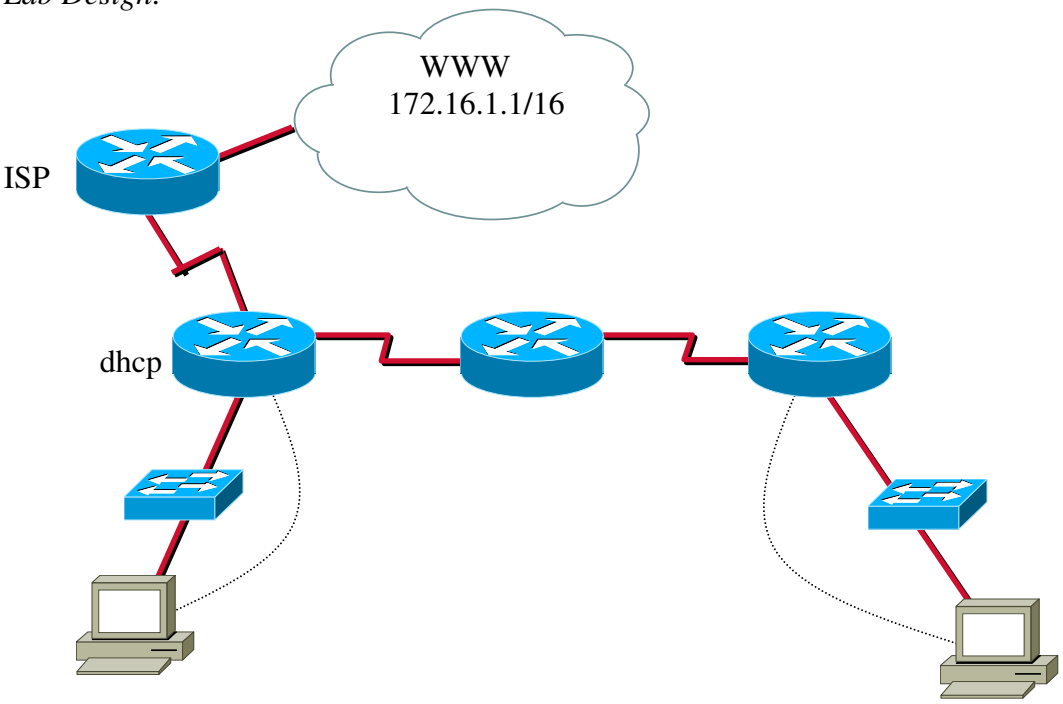
Prompt	Command	Shortcut	Description
	setup		
	help		
	?		
	enable		
	disable		
	exit		
	history		
	show		
	configuration		
	terminal		
	hostname		
	copy		
	running-configuration		
	start-configuration		
	write memory		
	show buffers		
	show flash		
	show interface		
	show memory		
	show protocols		
	show processes		
	show run		
	show start		
	show stacks		
	show tech		
	show version		
	password cisco		
	login		
	line vty		
	line con		
	line aux		
	logging synchronous		
	exec-timeout		
	enable password		

Prompt	Command	Shortcut	Description
	enable secret		
	ip host		
	no shut		
	interface e0/0		
	clockrate		
	ip address		
	router rip		
	network		
	ping		
	traceroute		
	show cdp		
	sh ip route		
	debug ip icmp		
	debug ip rip		
	undebug		
	show controller s0/0		
	interface loopback 0		
	telnet		
	show sessions		
	ip dhcp pool		
	ip dhcp excluded-address		
	ip helper-address		
	default-router		
	ip route		
	version 2		

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:

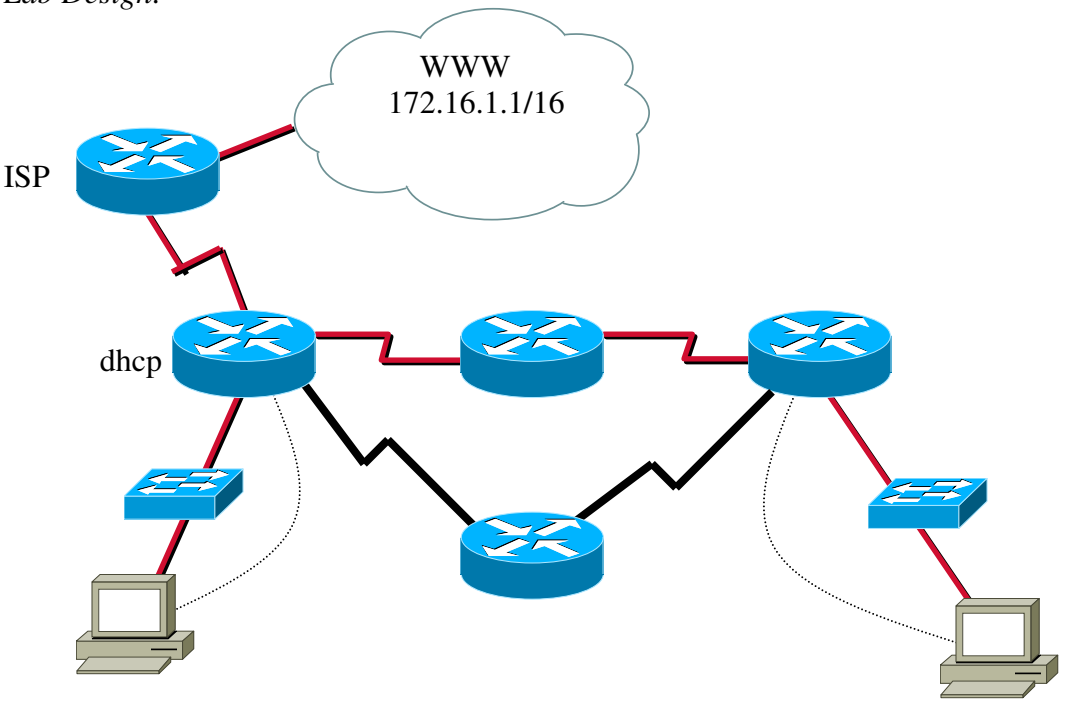


You are the network administrator for a medium-sized manufacturing company in Atlanta. They house all of their operations in three buildings on a city block. Each building has a router and six switches. There is a connection from one building to the ISP that is also used as a DHCP router/server. The 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

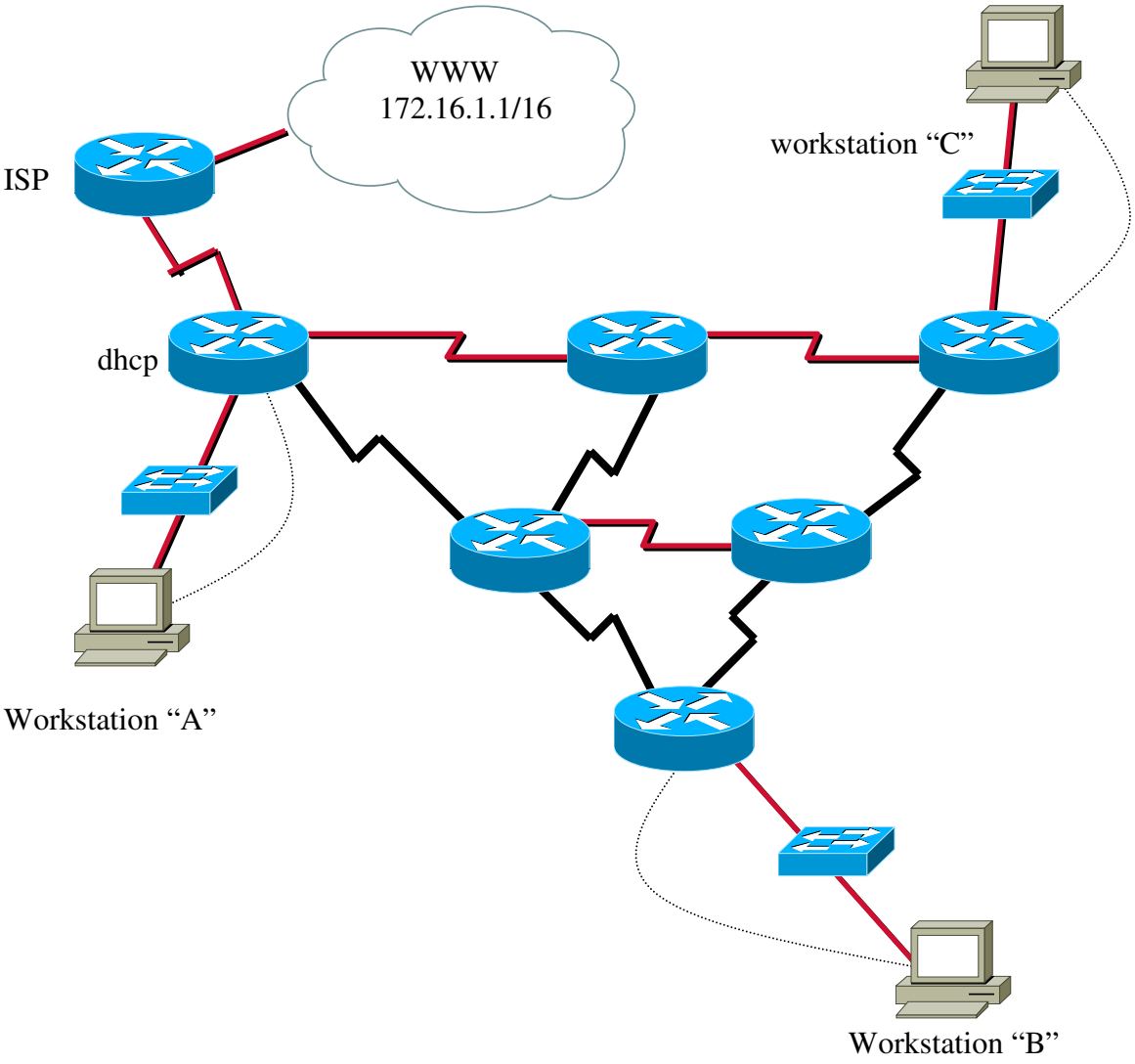
Variants:
Class "A," "B," or "C" private IP addresses only.
Mixed "A," "B," or "C" private IP addresses.
Class "A," "B," or "C" public IP addresses only.
Mixed "A," "B," or "C" public IP addresses.
Mixed public and private addressing.

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:



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.

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

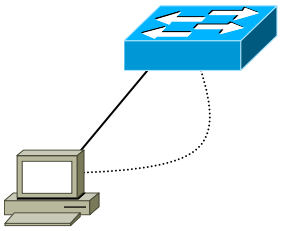
Part 3:
Switching

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
(1) straight through cable

Lab Design:



192.168.1.1/24

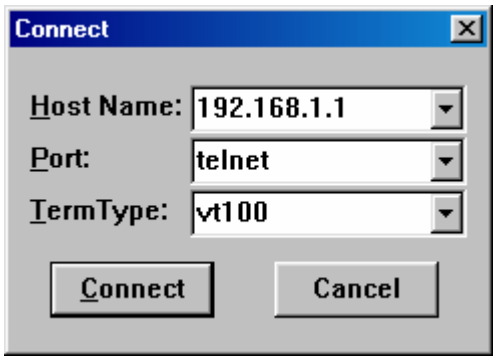
192.168.1.2/24
192.168.1.1 gw

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:

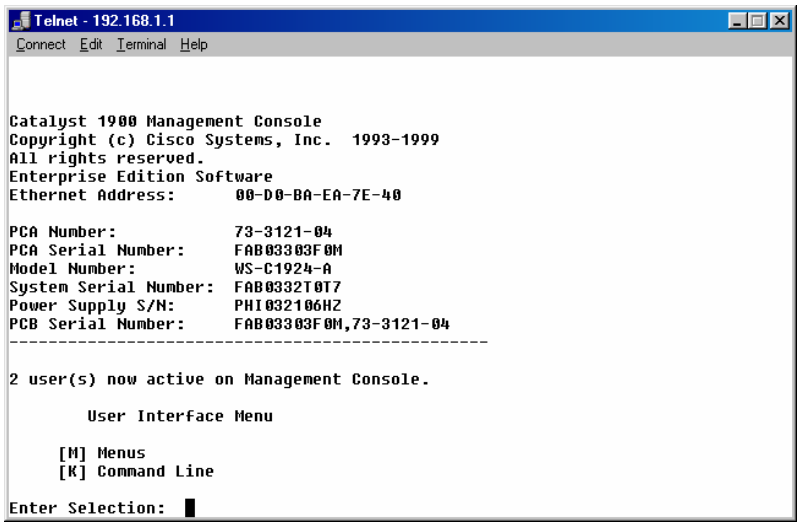
1. [I] IP configuration
2. [I] IP address
    a. 192.168.1.1
3. [S] Subnet mask
    a. 255.255.255.0
4. then, like our routers, we need a password in order to be able to telnet into this device:
    a. [X] Exit to previous menu
5. [M] Menus
6. [C] Console Settings
7. [M] Modify password
    a. cisco
    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. Its really cool. Open telnet by using Start then Run and typing telnet. The telnet window should open. Then click on "connect" and "remote session." When the pop up window opens type in the IP address of the switch and click on "Connect." You should see something like this:

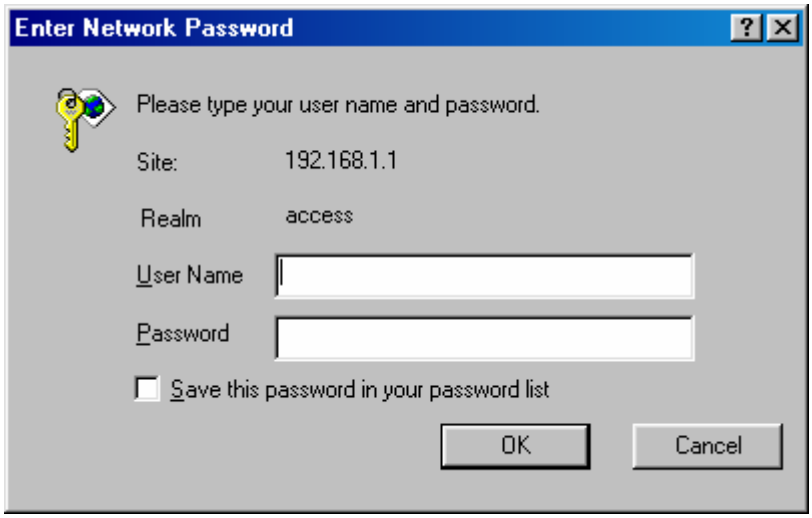


After only a couple of seconds you should see something like this:



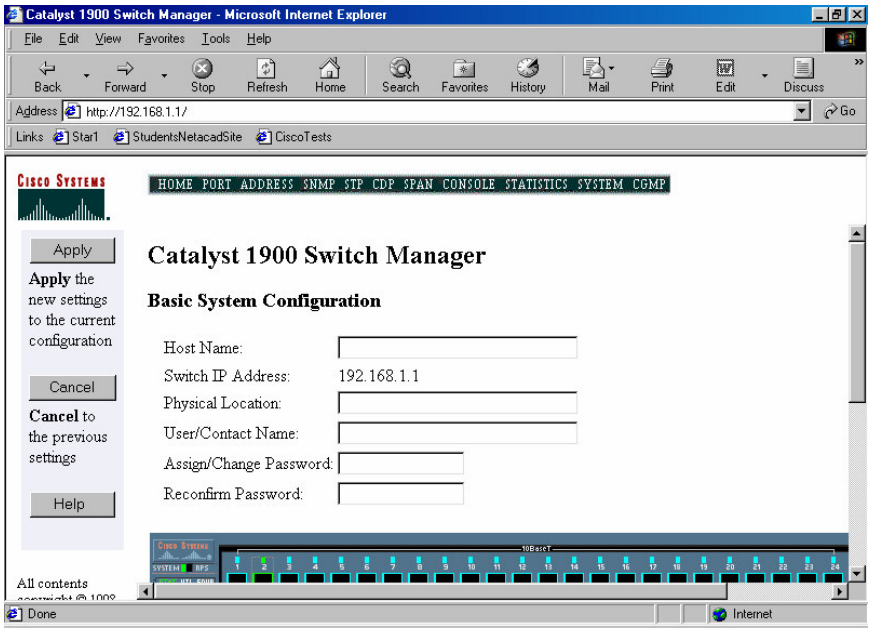
Notice how you no longer have the IP configuration option available.

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 this:



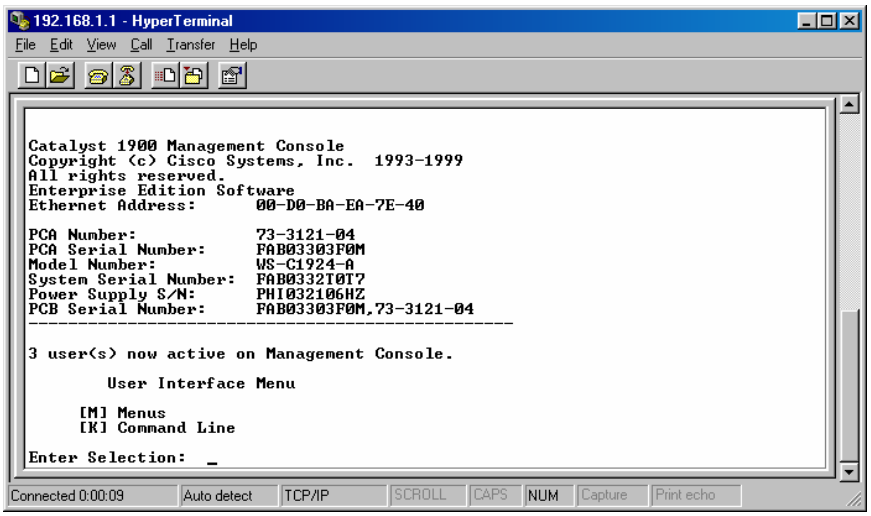
Remember how we just put in a password? Yup…we use it only…no user name required.

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 router. You will see this (next page):

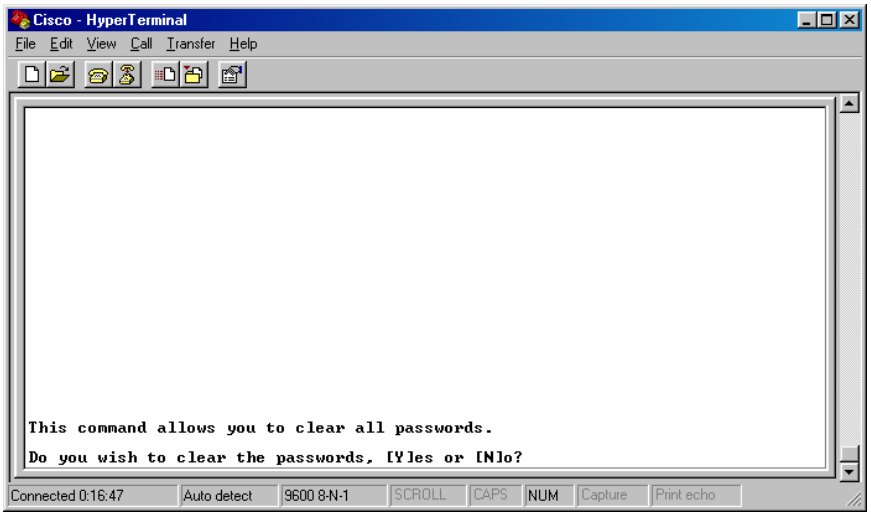


Resetting a switch:

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…its 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.

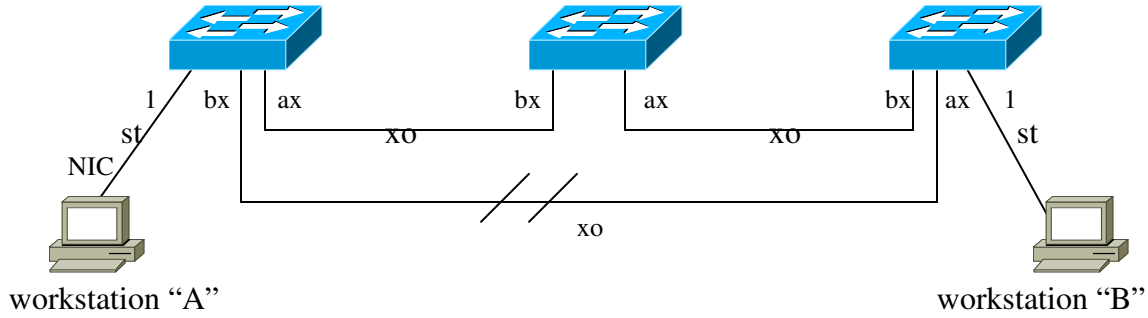
So What Have I Learned Here?

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.

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.

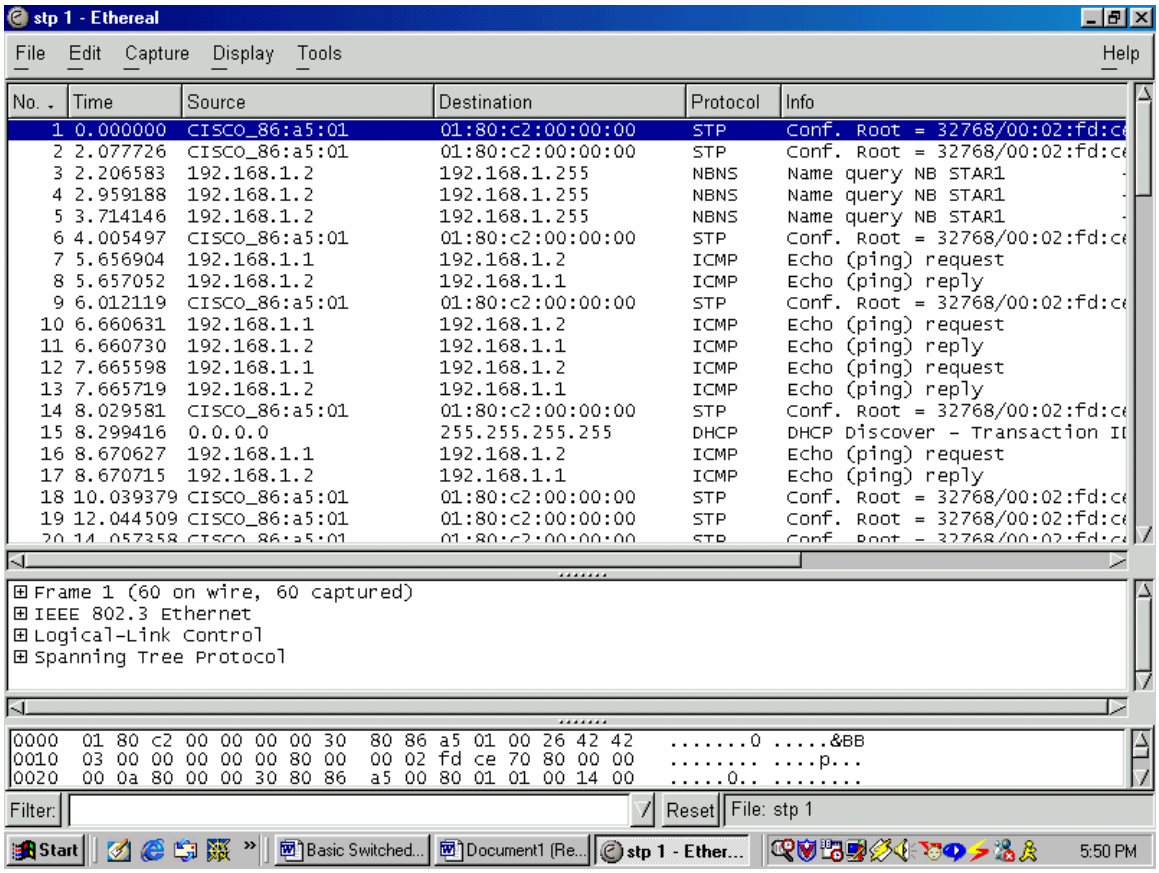
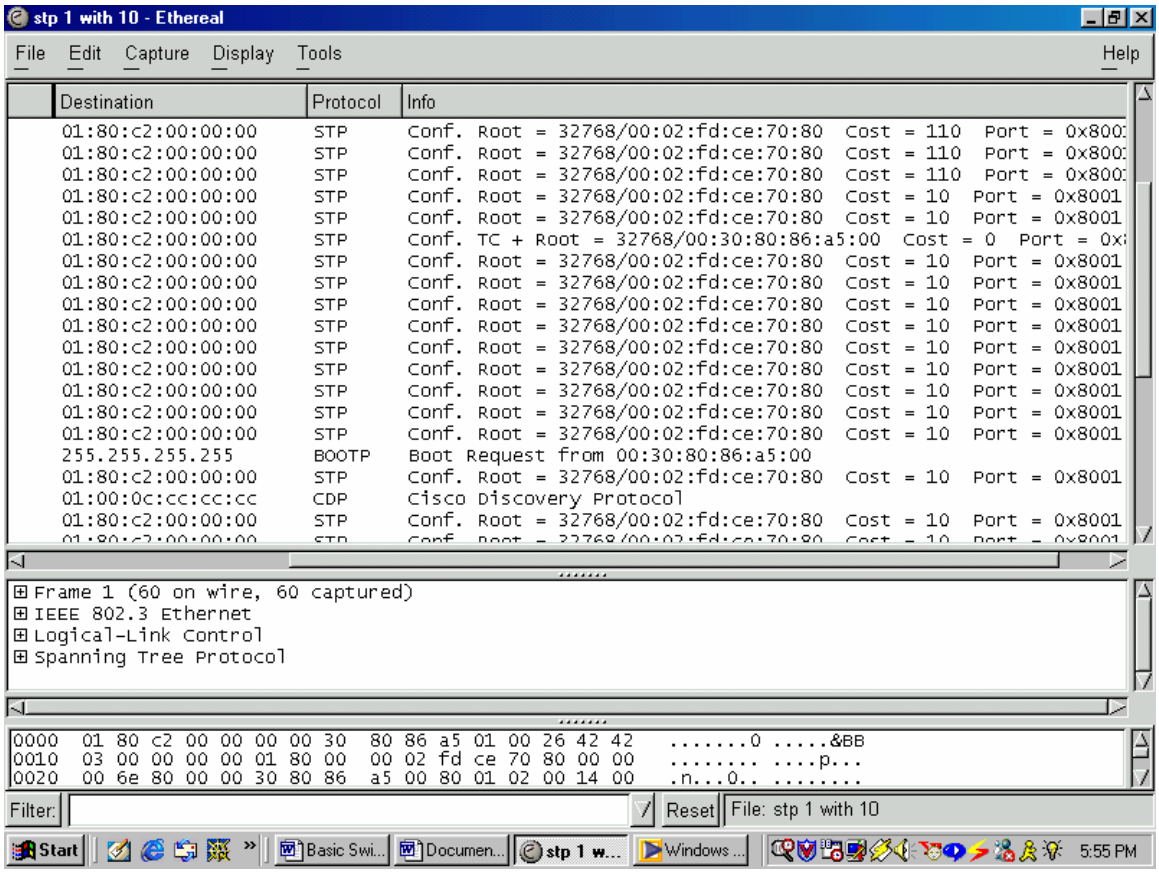


Figure 1—Capture for ping and STP. (note: complete icmp request and replies).

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.



```
Destination          Protocol  Info
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 110  Port = 0x800
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 110  Port = 0x800
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 110  Port = 0x800
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 10  Port = 0x8001
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 10  Port = 0x8001
01:80:c2:00:00:00    STP       Conf. TC + Root = 32768/00:30:80:86:a5:00  Cost = 0  Port = 0x
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 10  Port = 0x8001
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 10  Port = 0x8001
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 10  Port = 0x8001
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 10  Port = 0x8001
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 10  Port = 0x8001
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 10  Port = 0x8001
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 10  Port = 0x8001
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 10  Port = 0x8001
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 10  Port = 0x8001
255.255.255.255      BOOTP     Boot Request from 00:30:80:86:a5:00
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 10  Port = 0x8001
01:00:0c:cc:cc:cc    CDP       Cisco Discovery Protocol
01:80:c2:00:00:00    STP       Conf. Root = 32768/00:02:fd:ce:70:80  Cost = 10  Port = 0x8001
```

```
⊞ Frame 1 (60 on wire, 60 captured)
⊞ IEEE 802.3 Ethernet
⊞ Logical-Link Control
⊞ Spanning Tree Protocol
```

```
0000  01 80 c2 00 00 00 00 30  80 86 a5 01 00 26 42 42   .......0 .....&BB
0010  03 00 00 00 00 01 80 00  00 02 fd ce 70 80 00 00   ........ ....p...
0020  00 6e 80 00 00 30 80 86  a5 00 80 01 02 00 14 00   .n...0.. ........
```

Figure 2—STP showing cost of 10.

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 back up lines. Geeze…you thought it took long before.

So What Have I Learned Here?

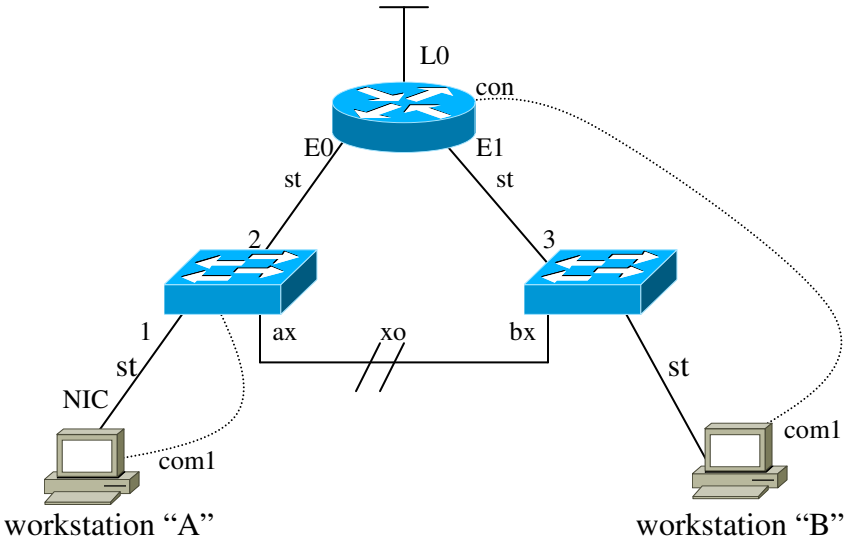
To set up redundant lines between switches we just need to know which ports to use for best service. It really doesn't matter which ones we use but certain ones are more preferred to others. In the next lab we will change settings.

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:

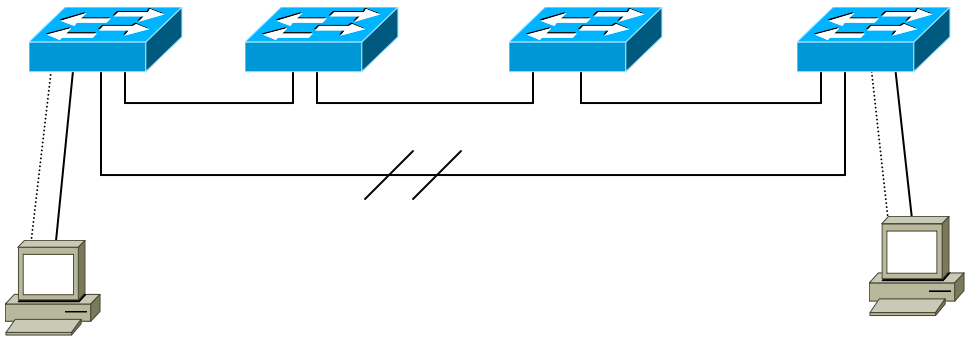
1. Cable the lab as shown. Ok. Now the fun starts. Use the 83.x.x.x network with a 16-bit mask. Oh don't get complacent with the easy numbers. Pick your own routing protocol to use.
2. Ping from workstation "A" to "B." Ping from each workstation to the loopback adapter. Use trace route for all three 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.

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 changing 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; 100Bast T 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"):

SW-A ____-____-____-____-____-____

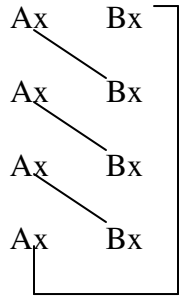
SW-B ____-____-____-____-____-____

SW-C ____-____-____-____-____-____

SW-D ____-____-____-____-____-____

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.



			root bridge		backup line	
SW-A	Ax	Bx	1st	2nd	3rd	4th
SW-B	Ax	Bx	1st	2nd	3rd	4th
SW-C	Ax	Bx	1st	2nd	3rd	4th
SW-D	Ax	Bx	1st	2nd	3rd	4th

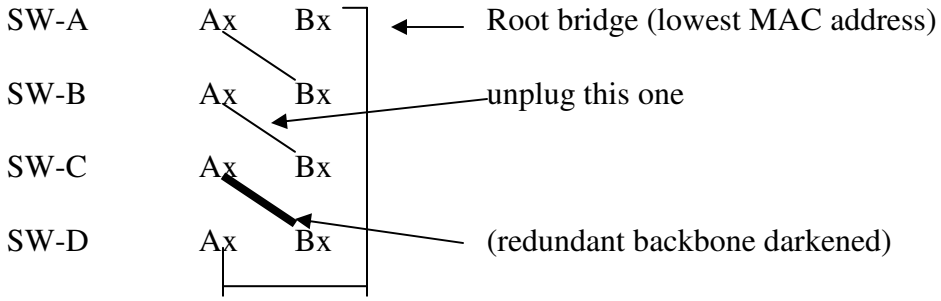
The backup line will be the line between the *highest* two MAC addresses (3rd and 4th). (The light on Ax for 3rd will be amber).

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.



   If you 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.
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.

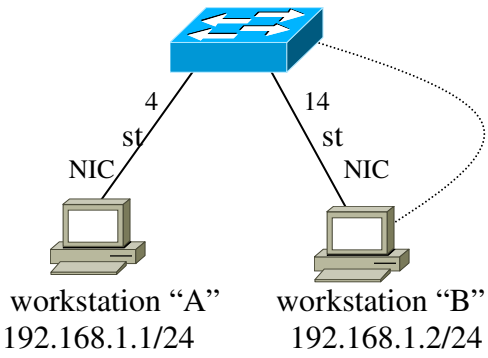
So What Did I Learn Here?
Now you can manually configure backbones between switches and automatically set priorities for backbone selection using the port configuration menu and costs. Just remember this is dependent upon the MAC addresses, with all other factors set to default. This lab also does not 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.

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:



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?

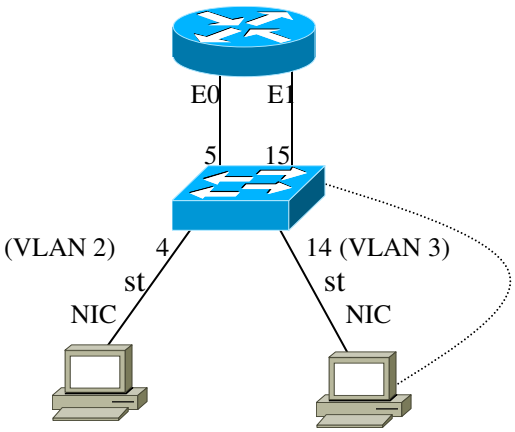
VLANs are nice to use in large networks. Instead of physically separating network users from each other with separate (and sometimes expensive devices) we can now do it logically without using added equipment. In the next lab we will add a router into our little lab design and see how it improves or messes up our network

Basic VLAN with One Router

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

Lab Diagram:



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.
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.
4. 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.

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.

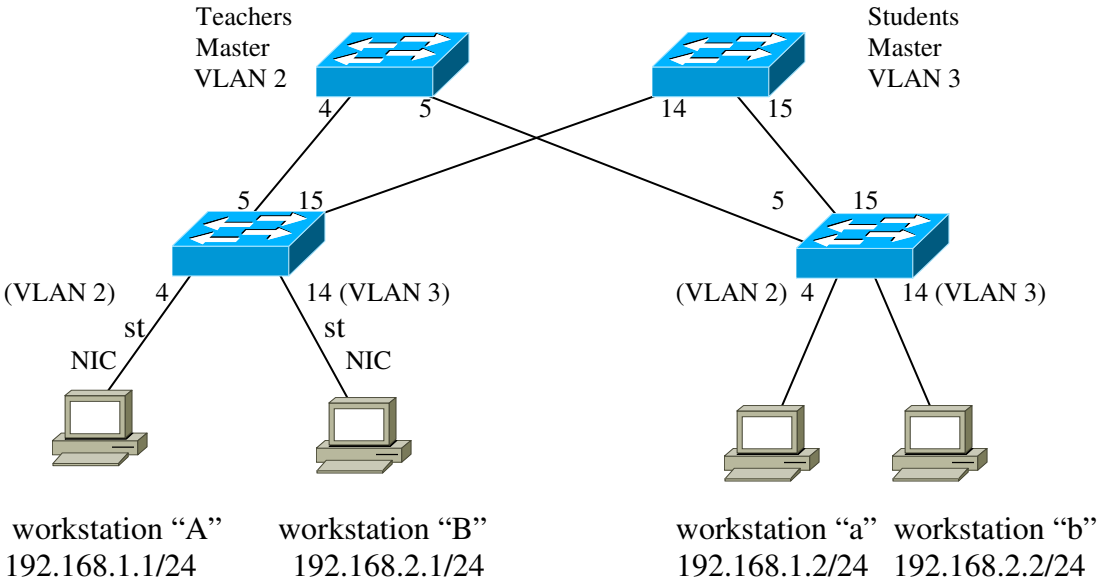
So What Have I Learned Here?

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.

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:



workstation "A" 192.168.1.1/24 | workstation "B" 192.168.2.1/24 | workstation "a" 192.168.1.2/24 | workstation "b" 192.168.2.2/24

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.

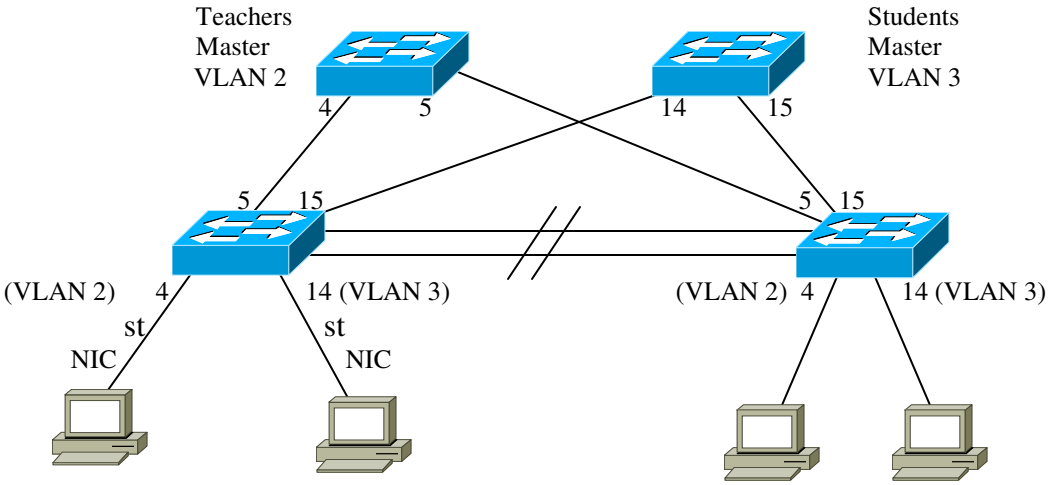
So What Have I Learned Here?
In this quick little lab you learned about setting up a partially meshed VLAN network. For most of the labs for this section you will build upon this design.

Mixing it up: VLAN's with STP

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:



workstation "A" 192.168.1.1/24 | workstation "B" 192.168.2.1/24 | workstation "a" 192.168.1.2/24 | workstation "b" 192.168.2.2/24

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.

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.

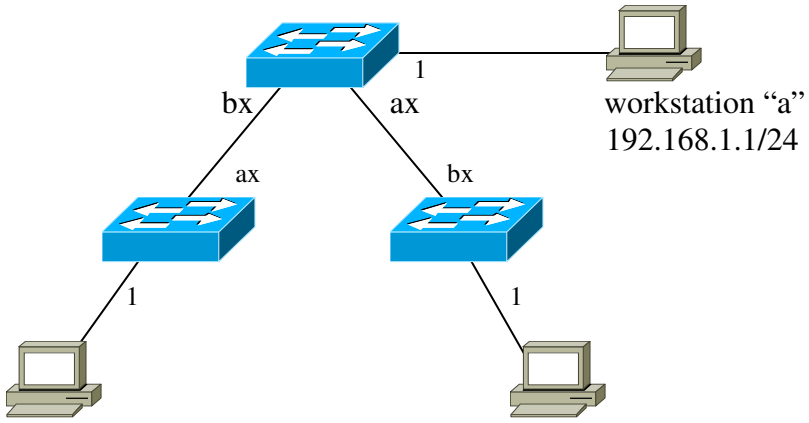
Continued:
Ok…great…you just got your wonderful network designed and implemented, so now you know why it needed to be adaptable: the "eccentric" president read an article in the "Harvard Business Review" (yeah…he could almost understand the big words) and wanted to implement a divisional team format. 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
(3) Crossover cables (xo)

Lab Diagram:



Background:
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:

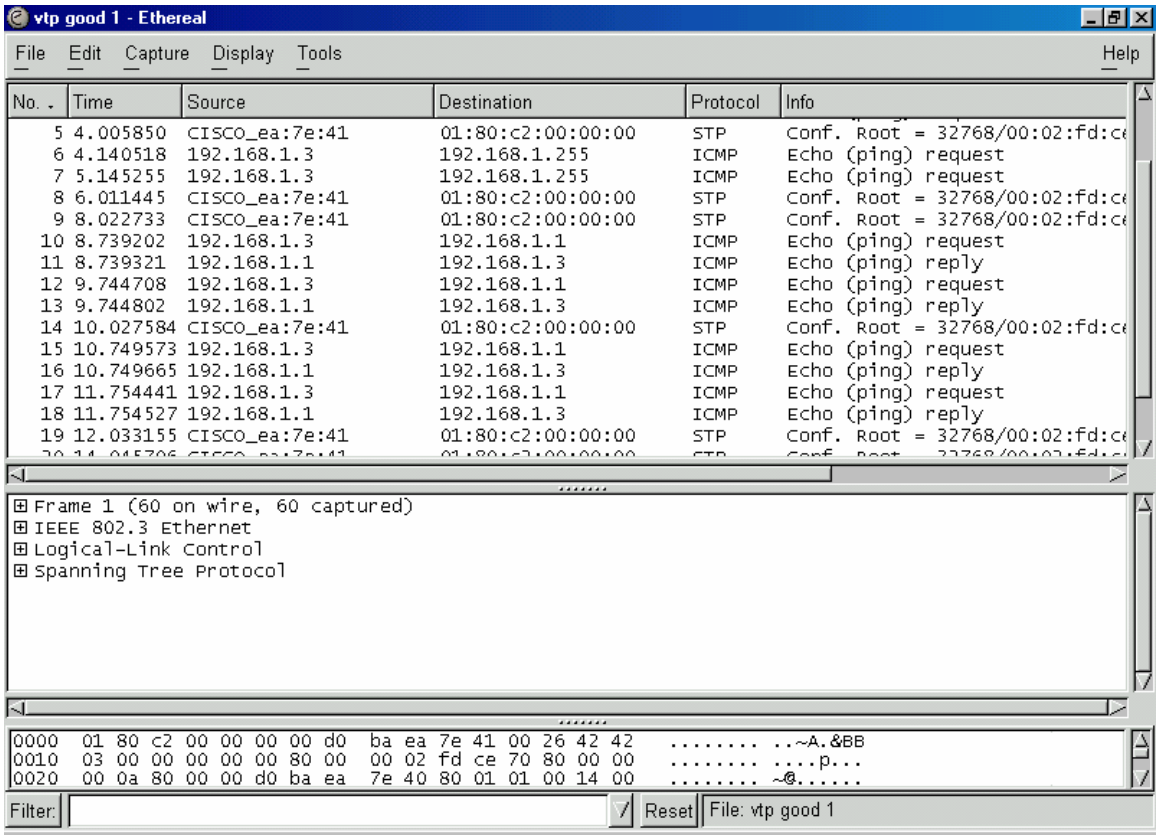


Figure 1—Good icmp request and replies seen.

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.

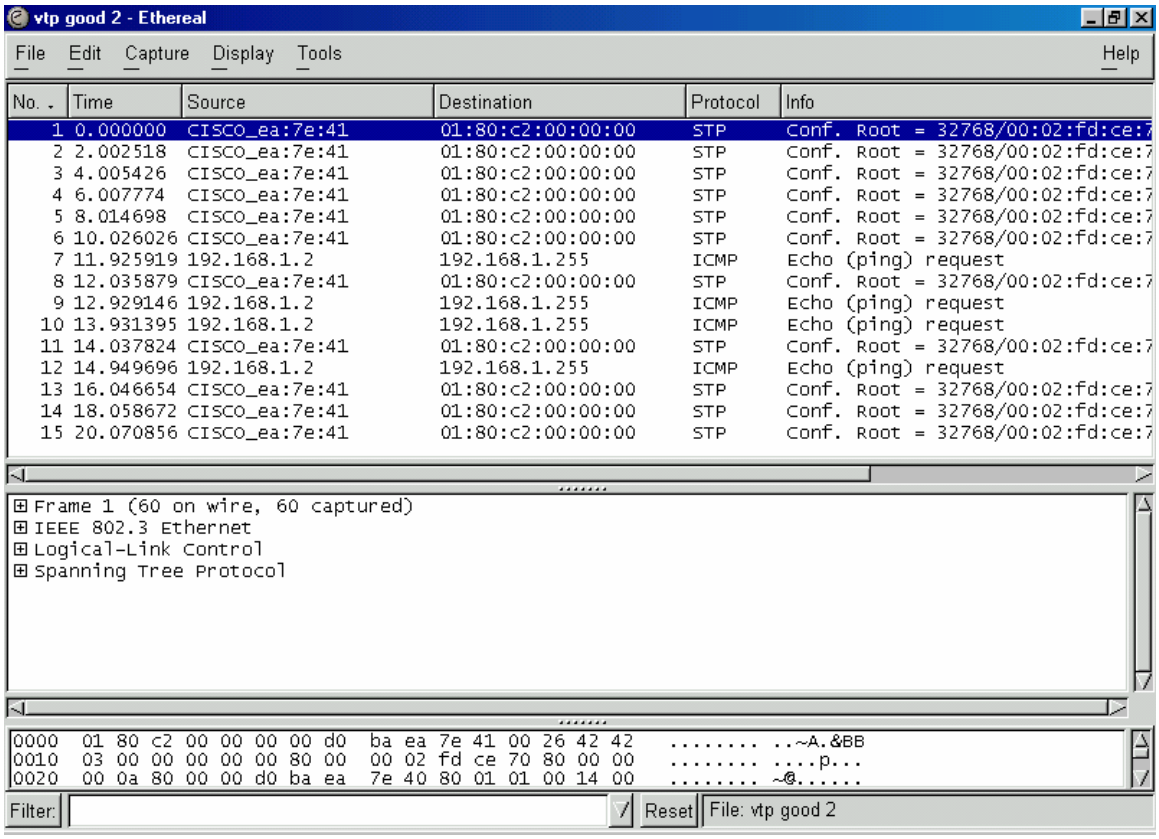


Figure 2—Only ping requests with VTP enabled.

12. Put VTP back on the switch. See the Switch Maintenance Lab for more in-depth instructions.

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.

Part 3 Command Review

Objective:
To list all commands utilized in Part 3 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. ☺

Prompt	Command	Shortcut	Description

Prompt	Command	Shortcut	Description

Whole Enchilada/Crazy Insano Lab #1 (WECIL): Switching

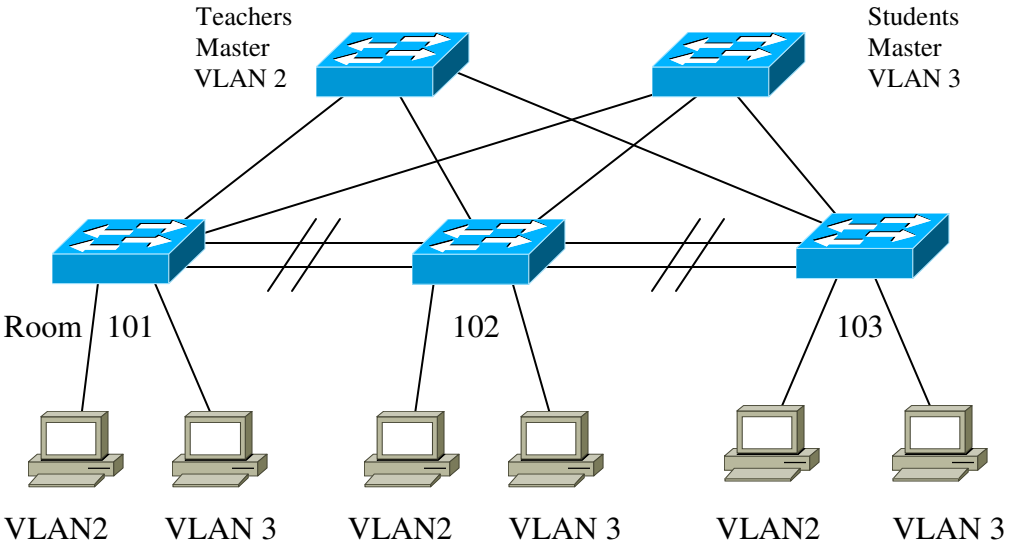
Objective:

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

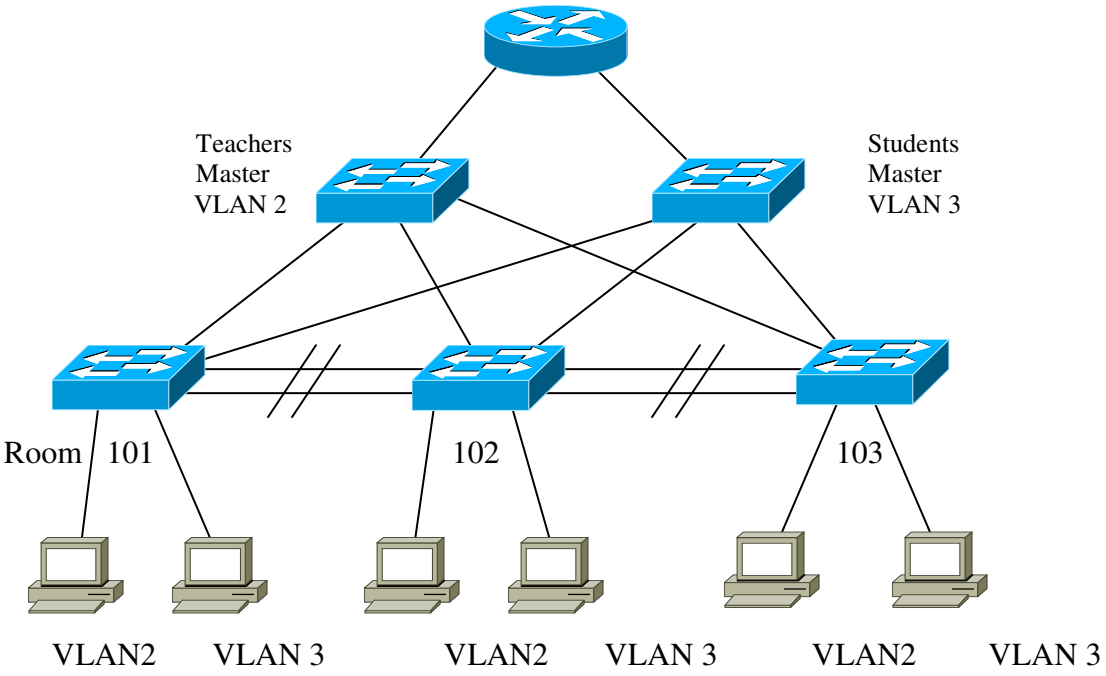
Objective:

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:

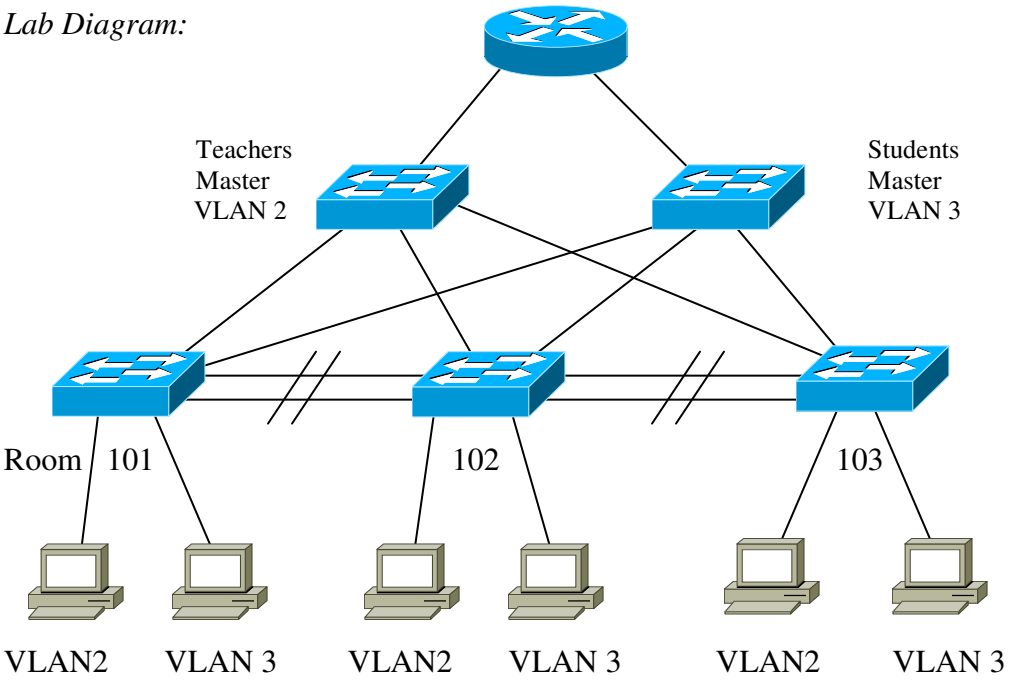
1. In this lab we will do the same lab but add a router to the mix. How does that change your IP addressing scheme? So the next time you design a switching network that may include routers in the future how would you design the IP scheme. Redraw your network.
2. All VLAN 2 devices should have communication 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

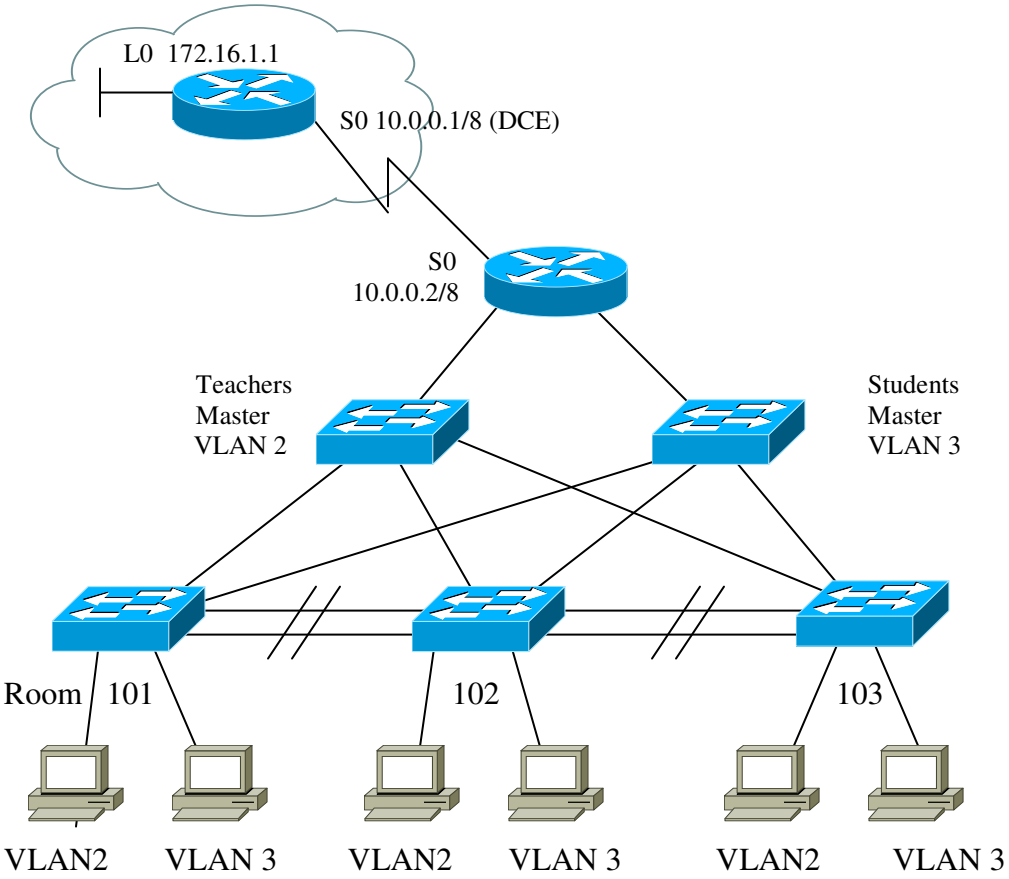
Objective:

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:

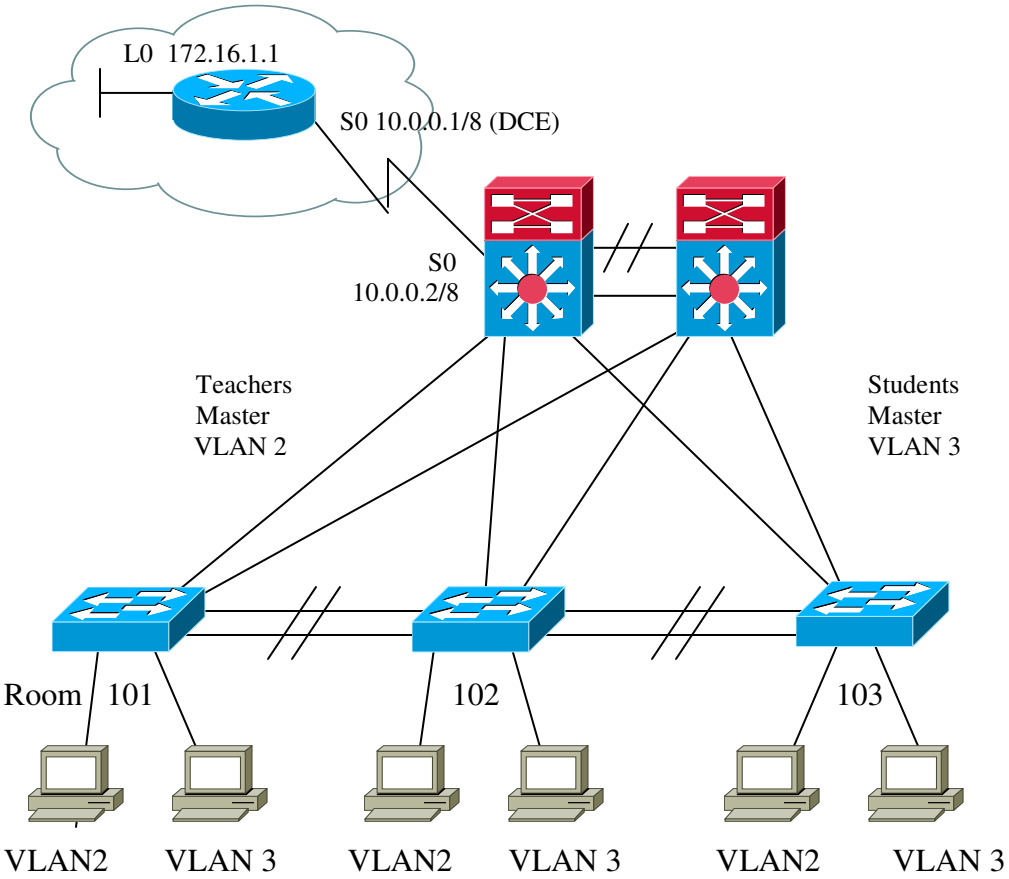
1. Let's repeat the last lab but add a web connection.
2. All VLAN 2 devices should have communication to all devices.
3. Test your redundant lines for VLAN 2.
4. All VLAN 3 devices should have communication to all devices.
5. Test your redundant lines for VLAN 3.
6. Each workstation should be able to ping the loopback on the ISP router.

Whole Enchilada/Crazy Insano Lab #5 (WECIL): Switching

Objective:

There is nothing to do here…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:



Part 4:
More on Routing

Paper Lab: CISCO Three-Layer Hierarchical Model

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.

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

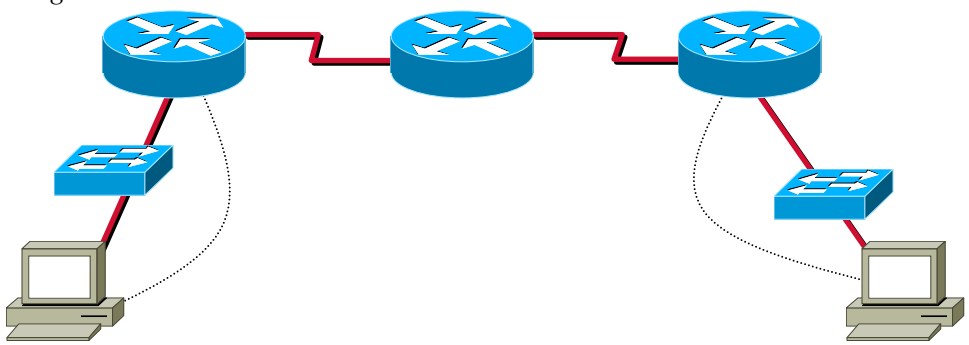
Match the CISCO networking device with its associated layer. Use a (1) for core-layer device, (2) for a distribution-layer device, or a (3) for an access-layer device.

Routers:	Layer:	Features:
700	_____	_____
800	_____	_____
1600	_____	_____
1720	_____	_____
2500	_____	_____
2600	_____	_____
3600	_____	_____
4000	_____	_____
7000	_____	_____
Switches:		
1548	_____	_____
1900	_____	_____
2900	_____	_____
4000	_____	_____
5000	_____	_____
6000	_____	_____
8000	_____	_____

Protocol Deathmatch: RIP versus RIPv2

Objectives:
To be able to discern between RIP and RIPv2 and when to use each. (A good review of part 2)

Lab Design:



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 using 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.
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?

Guest Router Name Derivation

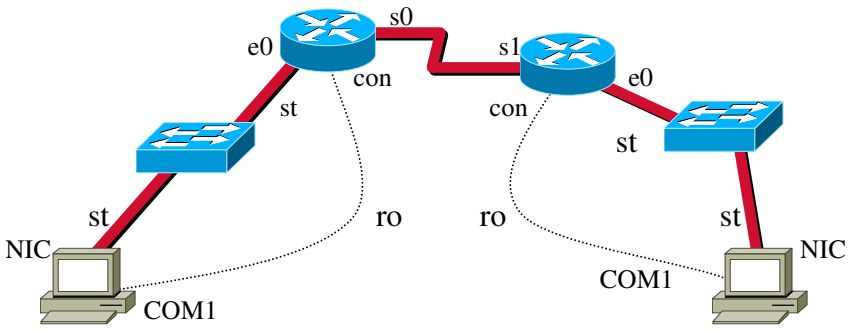
In 1991 Robert Morris became the first individual convicted for violating the 1986 Federal Computer Fraud and Abuse Act. He created an internet worm as part of a graduate school project whose sole purpose was to expose security vulnerabilities in networks so that network administrators could pro-actively 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."

Basic IGRP

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:



Workstation "A" Workstation "B"

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	B
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

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. You will learn more about these in a later lab.

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.
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#
```

```
C:\WINDOWS\Desktop>ping 184.36.67.3
Pinging 184.36.67.3 with 32 bytes of data:

Reply from 184.36.67.3: bytes=32 time=21ms TTL=126
Reply from 184.36.67.3: bytes=32 time=21ms TTL=126
Reply from 184.36.67.3: bytes=32 time=21ms TTL=126
Reply from 184.36.67.3: bytes=32 time=21ms TTL=126

Ping statistics for 184.36.67.3:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 21ms, Maximum =  21ms, Average =  21ms
```

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.

```
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

C    184.35.0.0/16 is directly connected, Serial0/1
C    184.36.0.0/16 is directly connected, Ethernet0/0
draper#

C:\WINDOWS\Desktop>ping 184.36.67.3
Pinging 184.36.67.3 with 32 bytes of data:

Reply from 184.34.67.1: Destination host unreachable.
Reply from 184.34.67.1: Destination host unreachable.
Reply from 184.34.67.1: Destination host unreachable.
Reply from 184.34.67.1: Destination host unreachable.

Ping statistics for 184.36.67.3:
   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>
```

Supplemental Lab or Challenge Activity:

1. Change the autonomous number on router 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

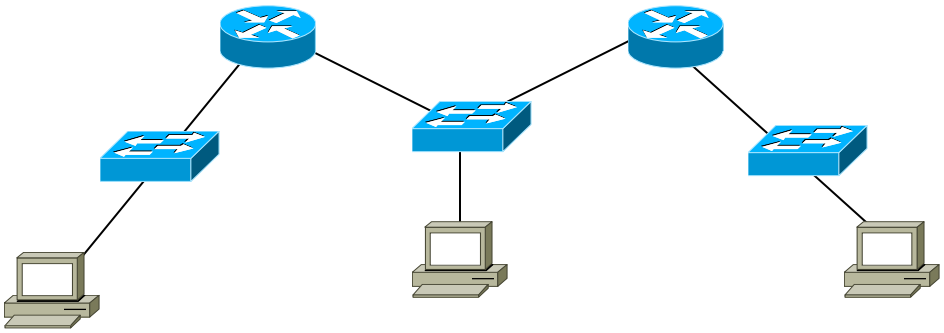
John Draper, a.k.a. "Captain Crunch," gained notoriety in the 1970's as a "phreaker" (phone hacker) when he figured out how pay phones work. He discovered when you put a dime in a payphone (calls in the 1970's used to be 10 cents) the telephone had an electromechanical converter that sent a 2600-hertz 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.

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			
Hostnames	Kevin	Paulsen	
E0	38.12.245.1/8	40.12.245.1/8	
E1	39.12.245.1	39.12.245.2/8	
Workstations	A	B	C
IP	38.12.245.2	40.12.245.2	39.12.245.3/8
SM	255.0.0.0	255.0.0.0	255.0.0.0
GW 1	38.12.245.1	40.12.245.1	39.12.245.1
GW 2	n/a	n/a	39.12.245.2

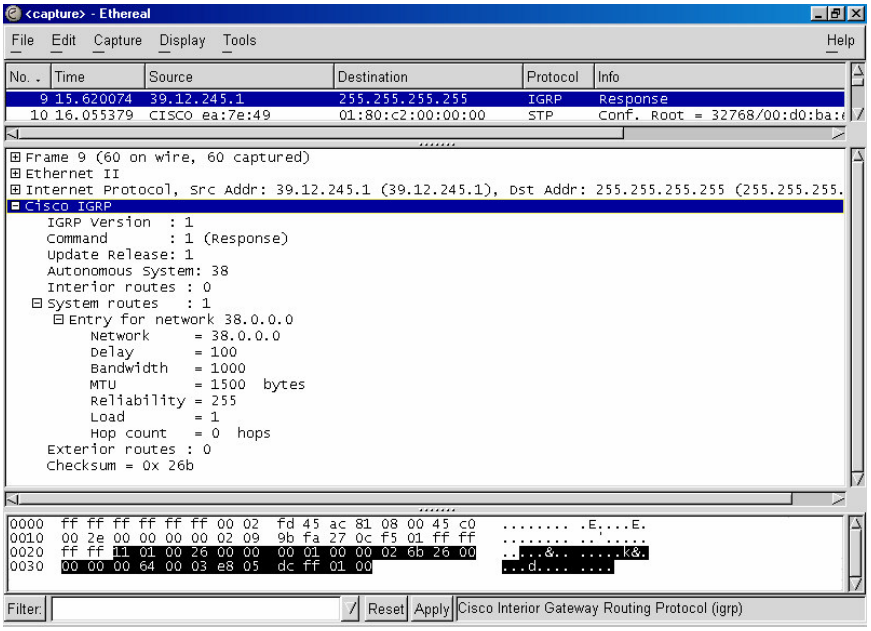
Background:
One of the disadvantages of using the Ethereal protocol inspector is that it will only capture packets on the subnet to which it is attached. In order to grab those IGRP packets we must set up a network that will allow us to do so. In the last lab we used a serial line between the two routers. 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 jim dandy.
3. Do a show ip route. It should look like this:

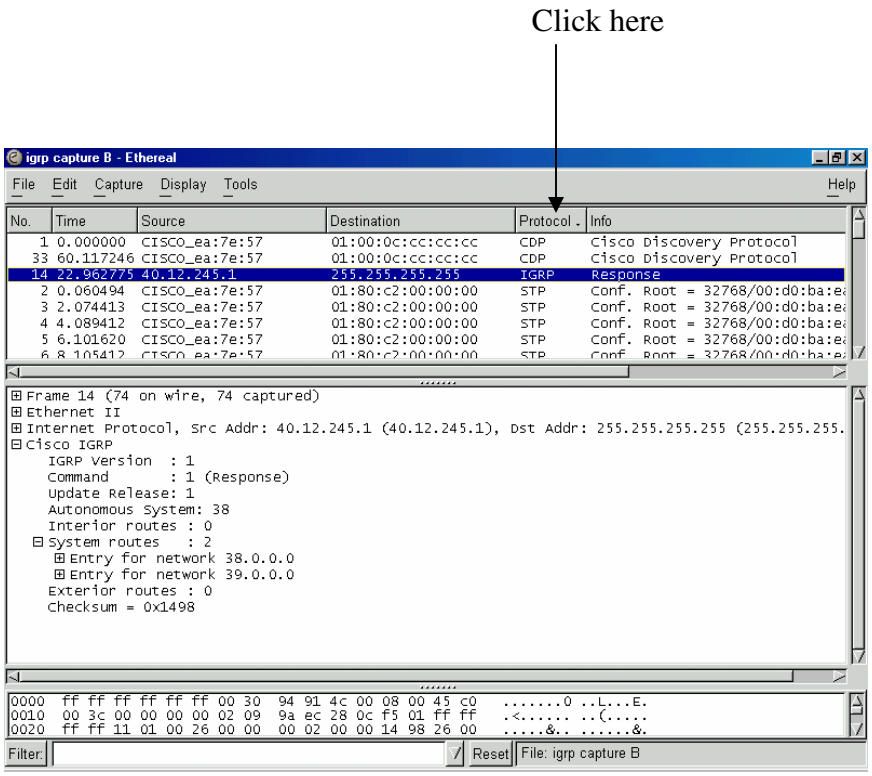
```
kevin#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       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!

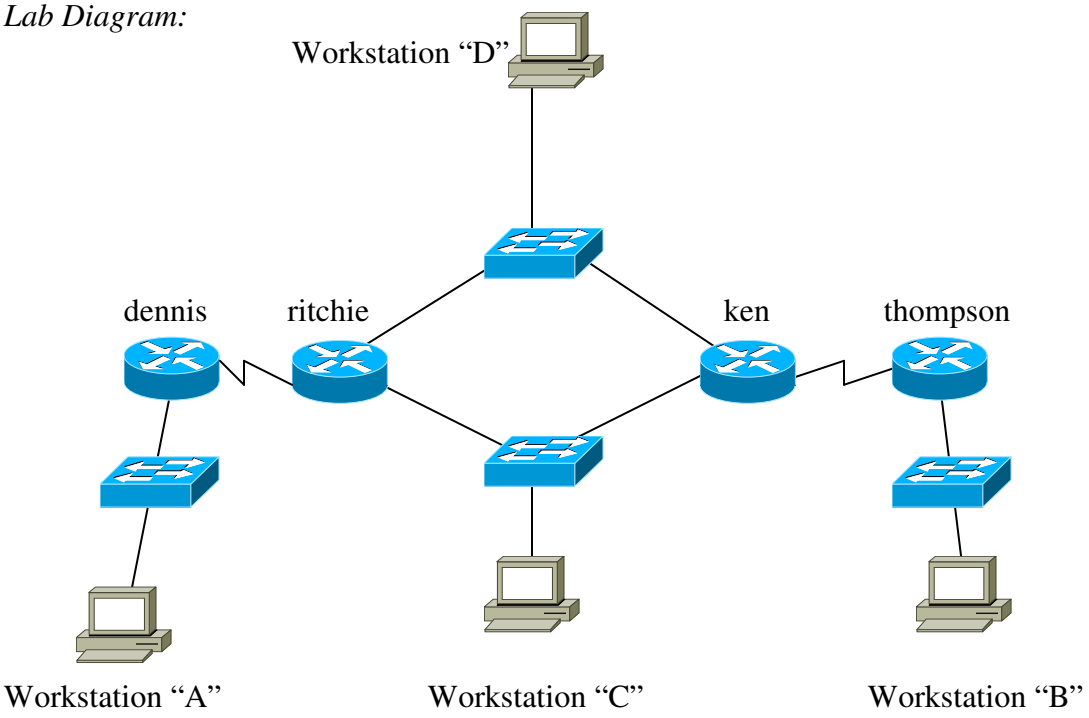
Guest Router Name Derivation

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.

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

Hostnames	dennis	ritchie
E0	200.150.100.1/24	202.150.100.1/24
E1	n/a	203.150.100.1/24
S0 (DCE)	201.150.100.1/24	n/a
S1	n/a	201.150.100.2/24

Routers

Hostnames	ken	Thompson
E0	202.150.100.2/24	200.150.101.1/24
E1	203.150.100.2/24	n/a
S0	n/a	201.150.101.1/24
S1	201.150.101.2/24	n/a

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:
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.

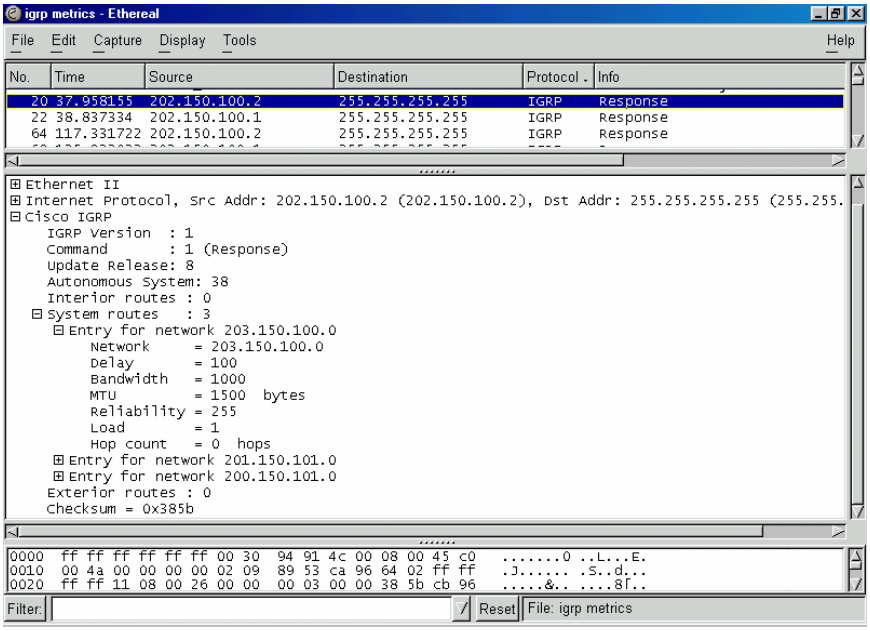
Step-By-Step Instructions:
1. Set up and cable the lab as shown. Give yourself enough time to do this. Don't rush through it otherwise your typos will cause headaches.
2. Test ping from each workstation to each other. This should be just fine.
3. Do a show ip route on each router. They should look like 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 201.150.100.2, 00:00:19, 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, 00:00:19, 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 202.150.100.2, 00:00:58, Ethernet0/0
                      [100/8576] via 203.150.100.2, 00:00:58, Ethernet0/1
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 202.150.100.2, 00:00:58, Ethernet0/0
                      [100/8676] via 203.150.100.2, 00:00:58, 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
       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    201.150.101.0/24 is directly connected, Serial0/1
I    200.150.100.0/24 [100/8676] via 202.150.100.1, 00:00:41, Ethernet0/0
                      [100/8676] via 203.150.100.1, 00:00:41, Ethernet0/1
I    201.150.100.0/24 [100/8576] via 202.150.100.1, 00:00:41, Ethernet0/0
                      [100/8576] via 203.150.100.1, 00:00:41, Ethernet0/1
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
thompson#

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
  2   25 ms   25 ms   25 ms  201.150.100.2
  3   25 ms   25 ms   26 ms  202.150.100.2
  4   49 ms   49 ms   50 ms  201.150.101.1
  5   60 ms   60 ms   60 ms  STAR10616125 [200.150.101.2]

Trace complete.

C:\WINDOWS\Desktop>
```

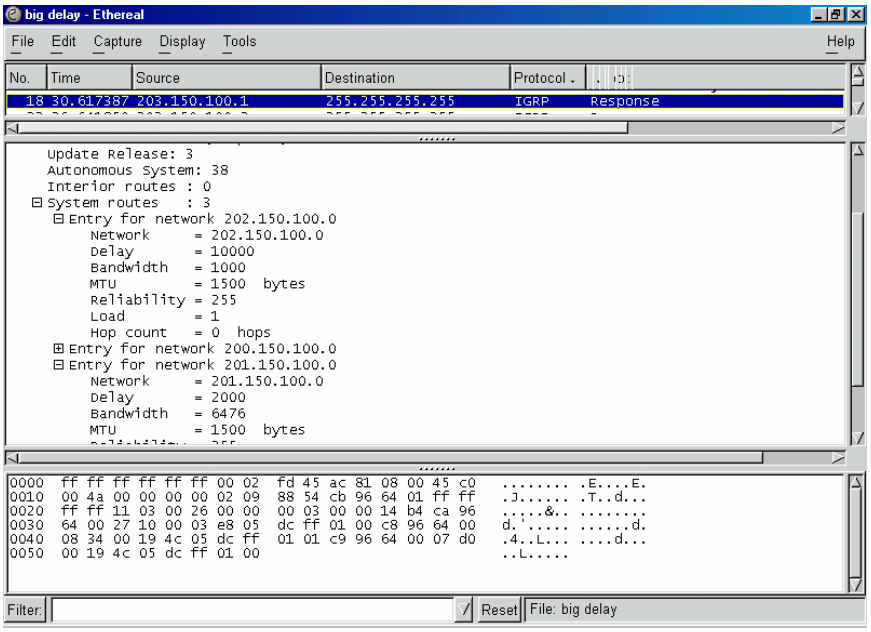
   The "crucial" step in our trace is in bold above. We can see the path is through the lower Ethernet path in our diagram. We can actually statically configure the Ethernet 1 interface (on ritchie) to pass the packets through Ethernet 1 by lowering (or raising) the specific metrics to make the 203.x..x.x. 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.
C:\WINDOWS\Desktop>
```

8. 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:

			203 route
change:	from		to
Bandwith	1000	E0	10000
OR	1000	E1	500
Delay	1000	E0	10000
OR	1000	E1	500
MTU	1500	E0	50
OR	1500	E1	2000*
RLY	255	E0	255**
OR	255	E1	100
Load	1	E0	255**
OR	1	E1	100**

* 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-jiggie. Don't erase your configurations…we will use the same one for the next lab.

Guest Router Name Derivation

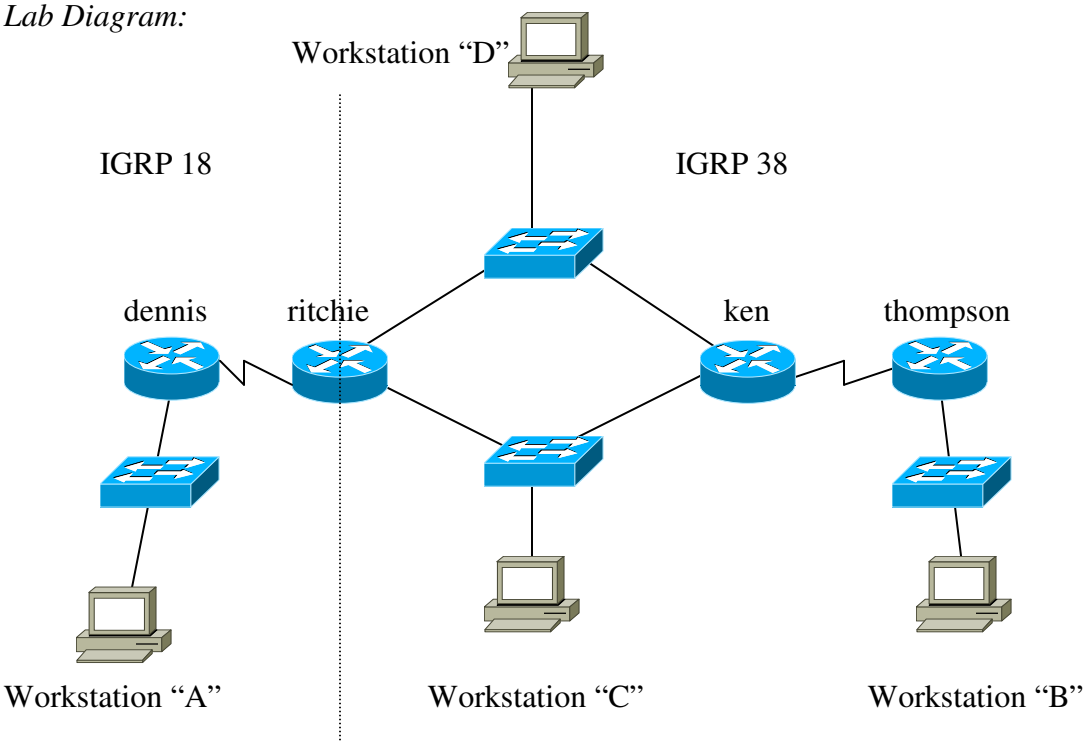
In 1969 Dennis Ritchie and Ken Thompson invented the UNIX operating System. If they only knew then what they were doing…creating software that would help put a man on the moon, transmit pictures back from Mars, and the solar system…oh, yeah…and give a green light to hackers everywhere. Nobody said anything was perfect.

Redistribution of IGRP and RIP

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

Lab Diagram:



Addressing:

Routers

Hostnames	dennis	ritchie
E0	200.150.100.1/24	202.150.100.1/24
E1	n/a	203.150.100.1/24
S0 (DCE)	201.150.100.1/24	n/a
S1	n/a	201.150.100.2/24

Routers

Hostnames	ken	Thompson
E0	202.150.100.2/24	200.150.101.1/24
E1	203.150.100.2/24	n/a
S0	n/a	201.150.101.1/24
S1	201.150.101.2/24	n/a

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
       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
                      [100/8576] via 202.150.100.2, 00:00:20, Ethernet0/0
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, Ethernet0/1
                      [100/8676] via 202.150.100.2, 00:00:20, Ethernet0/0
ritchie#
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    201.150.101.0/24 is directly connected, Serial0/1
I    200.150.100.0/24 [100/8676] via 202.150.100.1, 00:00:44, Ethernet0/0
                      [100/8676] via 203.150.100.1, 00:00:44, Ethernet0/1
I    201.150.100.0/24 [100/8576] via 202.150.100.1, 00:00:44, Ethernet0/0
                      [100/8576] via 203.150.100.1, 00:00:44, Ethernet0/1
I    200.150.101.0/24 [100/8576] via 201.150.101.1, 00:00:25, Serial0/1
ken#
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, Serial0/0
I    201.150.100.0/24 [100/10576] via 201.150.101.2, 00:00:38, Serial0/0
C    200.150.101.0/24 is directly connected, Ethernet0/0
thompson#
```

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. 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
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 [100/10000101] 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 "redistributed" as IGRP (**I**) routes to the right of the ritchie router.

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.

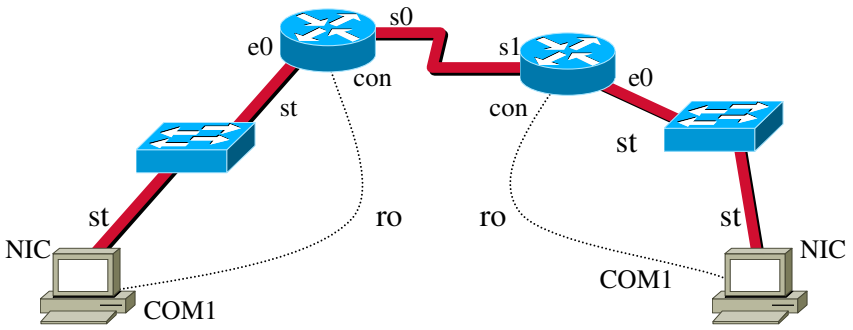
Guest Router Name Derivation

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.

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:



Workstation "A" Workstation "B"

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

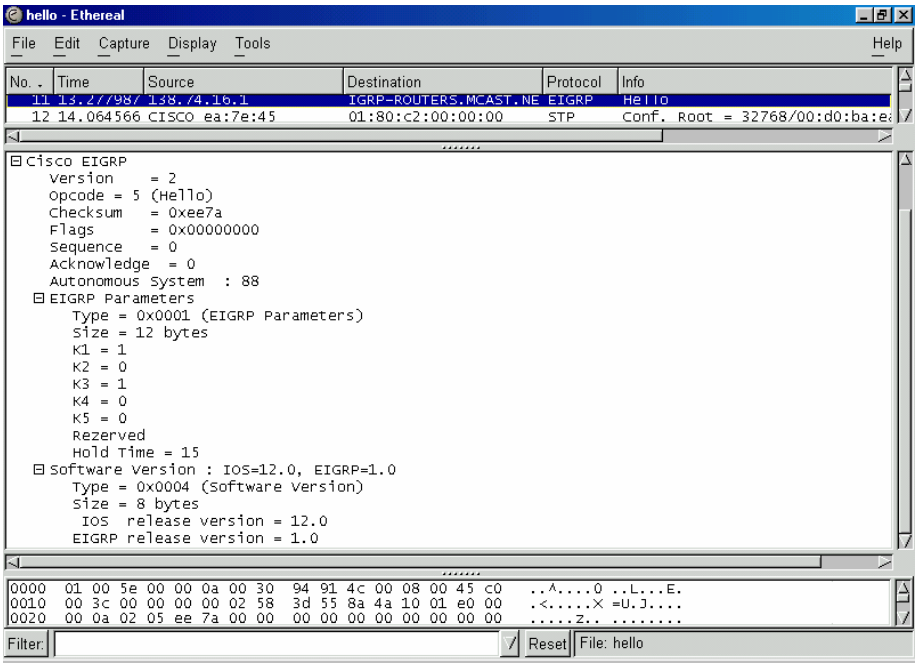
to add the extra metrics statements like with did with those. EIGRP and IGRP can be redistributed without those extra metric statements. How easy is that? Unlike IGRP, EIGRP supports Variable Length Subnet Masking (VLSM) so we do not have to be so concerned about the classful boundaries like we had to with IGRP (and RIP too). Instead of sending updates every x 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)#netwrork 14.32.0.0
games(config-router)#network220.34.98.0
```

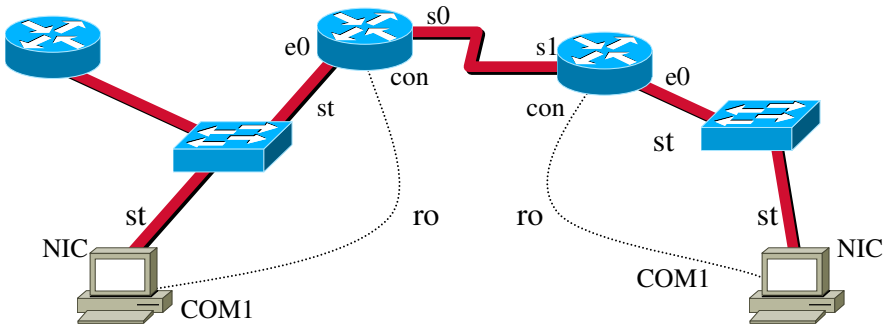
3. Try to ping from A to B. It should work just fine.
4. Start Ethereal on workstation A. After about 30 packets disconnect the serial line, wait a few seconds, and then plug it back in. Remember EIGRP will only send the tables when a change occurs, otherwise it just sends "hello" packets. We should 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.

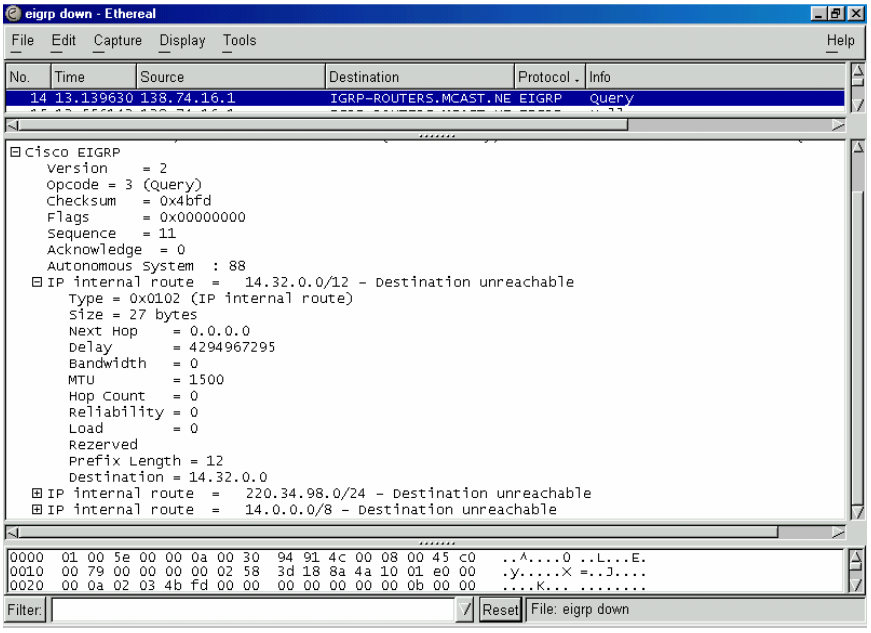
New Lab Diagram:



Workstation "A" Workstation "B"

New Router: "WOPR"
E0/0 138.74.16.2.20/20
L0 1.1.1.1/8

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 out put to a debug eigrp packets:

```
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/rely0/0
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:   AS 88, Flags 0x0, Seq 0/16 idbQ 0/0 iidbQ un/rely 0/0 peerQ
un/rely1/0
00:08:17: EIGRP: Enqueueing REPLY on Ethernet0/0 nbr 138.74.16.1
iidbQ un/rely 0
/1 peerQ un/rely 0/0 serno 9-11
00:08:17: EIGRP:  Requeued unicast on Ethernet0/0
00:08:17: EIGRP: Sending REPLY on Ethernet0/0 nbr 138.74.16.1
00:08:17:   AS 88, Flags 0x0, Seq 6/16 idbQ 0/0 iidbQ un/rely 0/0 peerQ
un/rely0/1 serno 9-11
00:08:17: EIGRP: Received ACK on Ethernet0/0 nbr 138.74.16.1
00:08:17:   AS 88, Flags 0x0, Seq 0/6 idbQ 0/0 iidbQ un/rely 0/0 peerQ
un/rely 0/1
00:08:31: EIGRP: Ethernet0/0 multicast flow blocking cleared
wopr#
```

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
wopr#
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
D       138.74.0.0/16 is a summary, 00:23:55, Null0
C       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
       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 [90/2323456] via 14.32.0.1, 00:01:18, Serial0/1
D    138.74.0.0/16 [90/2195456] via 14.32.0.1, 00:01:19, Serial0/1
     220.34.98.0/24 is variably subnetted, 2 subnets, 2 masks
C       220.34.98.16/28 is directly connected, Ethernet0/0
D       220.34.98.0/24 is a summary, 00:21:54, Null0
     14.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       14.0.0.0/8 is a summary, 00:01:23, Null0
C       14.32.0.0/12 is directly connected, Serial0/1
games#
```

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

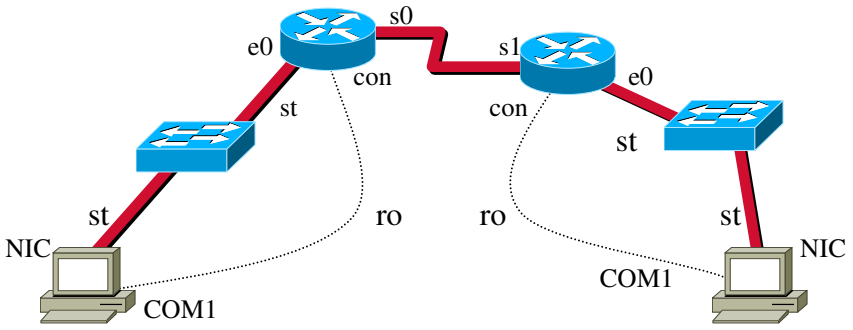
War Games is the first "great" hacker movie from 1984 staring Matthew Broderick, Alley Sheedy, and Dabney Coleman. In a round about way it created "idols" for young disenchanted computer geeks to become hackers. In the movie Matthew Broderick "hacked" into a military computer called "WOPR." Most of the 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.

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
(2) Switches
(4) Straight-through cables
(1) DCE/DTE serial cable
(2) rollover cables

Lab Diagram:



Workstation "A" Workstation "B"

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	B
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

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/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
wash#
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
leung#
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
```

```
wash#
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

leung#
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
wash#
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#

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.

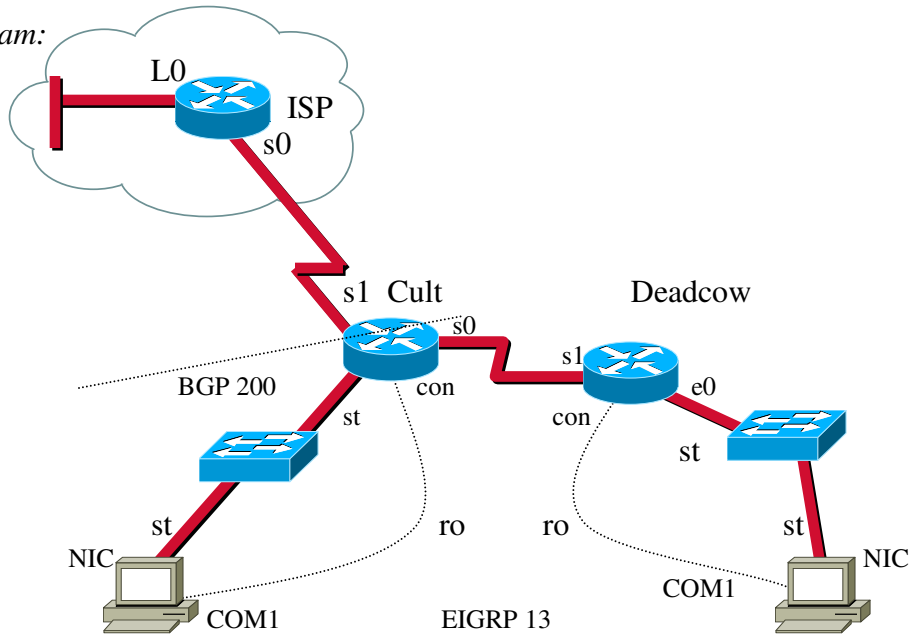
Guest Router Name

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!

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:



Workstation "A" Workstation "B"

Addressing:

Routers

Hostnames	ISP	Cult	Deadcow
E0	n/a	192.168.1.1/24	192.168.3.1/24
S0	210.1.1.1/24 (DCE)	192.168.2.1/24(DCE)	n/a
S1	n/a	210.1.1.2/24	192.168.2.2/24

Loopback 193.168.1.1/24 (L0)

Workstations	A	B
IP	192.168.1.3	192.168.3.3
SM	255.255.255.0	255.255.255.0
GW	192.168.1.2	192.168.3.2

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
```

```
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

B    193.168.1.0/24 [20/0] via 210.1.1.1, 00:22:06
C    192.168.1.0/24 is directly connected, Ethernet0/0
C    210.1.1.0/24 is directly connected, Serial0/0
C    192.168.2.0/24 is directly connected, Serial0/0
D    192.168.3.0/24 [90/2195456] via 192.168.2.2, 00:07:23, Serial0/0

deadcow#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 EX 193.168.1.0/24 [170/3097600] via 192.168.2.1, 00:03:57, Serial0/1
D    192.168.1.0/24 [90/2195456] via 192.168.2.1, 00:03:57, Serial0/1
C    192.168.2.0/24 is directly connected, Serial0/1
C    192.168.3.0/24 is directly connected, Ethernet0/0
deadcow#
```

6. We can use a command called show ip bgp to examine our bgp routes:

```
ISP#sh ip bgp
BGP table version is 65, local router ID is 193.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop            Metric     LocPrf  Weight Path
*> 192.168.1.0   210.1.1.2           0          0       200    ?
*> 192.168.2.0   210.1.1.2           0          0       200    ?
*> 192.168.3.0   210.1.1.2           2195456    0       200    ?
*> 193.168.1.0   0.0.0.0             0          32768          i
*> 210.1.1.0     0.0.0.0             0          32768          i
*                210.1.1.2           0          0       200    i
ISP#
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

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 192.168.1.0	0.0.0.0	0	32768		?
*> 192.168.2.0	0.0.0.0	0	32768		?
*> 192.168.3.0	192.168.2.2	2195456	32768		?
*> 193.168.1.0	210.1.1.1	0	0	100	i
* 210.1.1.0	210.1.1.1	0	0	100	i
*>	0.0.0.0	0	32768		i

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.

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.

	Link State	Distance Vector	Hybrid	Alg.
RIP		yes		
RIP version 2			no	
IGRP				
EIGRP				
BGP				
OSPF		no		SPF

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)

	Updates	Invalid	Hold-down	Flush
RIP	30 seconds			
RIP version 2				
IGRP		270		670
EIGRP				
BGP				
OSPF				

Which of the following are the *default* metrics for each routing protocol?

	Bandwidth	Reliability	Load	MTU	Delay	K-metrics	Hop count
RIP	No		No	No		No	
RIPv2		No		No			
IGRP		No			No		
EIGRP							
BGP							No
OSPF							No

Enable routing
If you type in:

```
router(config)#router rip
router(config-router)#network 172.16.1.1 255.255.255.0
```

then what will appear with a show run?

a. router rip
network 172.16.1.0

b. router rip
network 172.16.1.1

c. router rip
network 172.16.1.1 255.255.255.0

d. router rip
network 172.16.1.0 255.255.255.0

If you type in:

```
router(config)#router igrp 38
router(config-router)#network 172.16.1.1 255.255.255.0
```

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

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

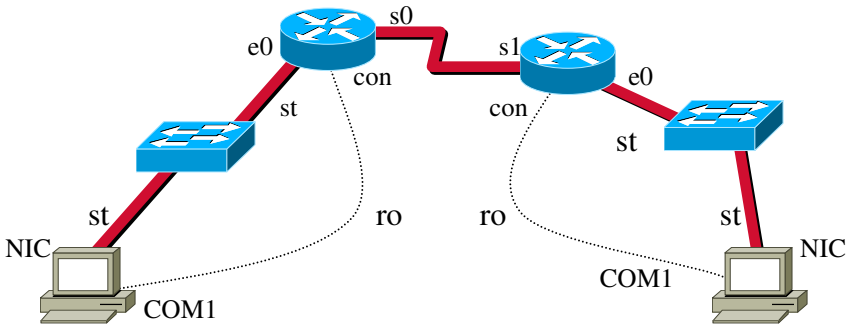
d. router eigrp 38
   network 172.16.1.0 255.255.255.0

Basic IP/IPX with Dynamic Routing

Objective:
To learn how to set up basic IP/IPX between two routers on a network using dynamic routing.

Tools and Materials:
(2) PC/workstations
(2) Routers
(2) Switches
(4) Straight-through cables
(1) DCE/DTE serial cable
(2) rollover cables

IP Lab Diagram:



Workstation "A" Workstation "B"

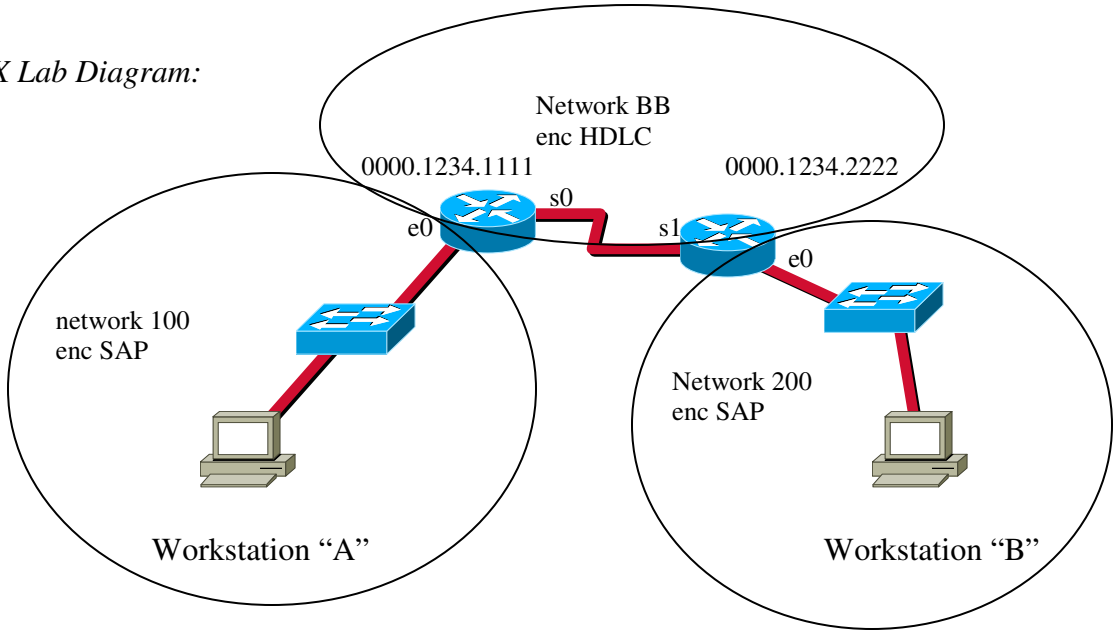
IP Addressing:

Routers		
Hostnames	Steve	Gibson
E0	192.168.1.1/24	192.168.3.1/24
S0	192.168.2.1/24 (DCE)	n/a
S1	n/a	192.168.2.2/24 (DTE)
Workstations	A	B
IP	192.168.1.2	192.168.3.2
SM	255.255.255.0	255.255.255.0
GW	192.168.1.1	192.168.3.1

Step-By-Step Instructions:

1. Set up and cable the lab as shown. Enable IGRP as a routing protocol using autonomous system number 16.
2. Test ping from workstation a to b.

IPX Lab Diagram:



3. Now that we know everything works fine lets add in our IPX with dynamic (automatic) routing. We found out from our Novell geek that we need to use 808.2 (Ethernet—SAP) for our IPX routing:

```
steve(config)#ipx routing 0000.1234.1111

steve(config)#int e0/0
steve(config-if)#ipx network 100 enc SAP

steve(config)#int s0/0
steve(config-if)#ipx network BB
```

The first line enables IPX routing on our router with the router having an IPX number of 0000.1234.1111. In the next two groups of commands we bind IPX network numbers and their frame types to the interfaces. For dynamic routing that is about it…here are the commands for the other router too:

```
gibson(config)#ipx routing 0000.1234.2222
gibson(config)#int e0/0
gibson(config-if)#ipx network 200 enc SAP

gibson(config)#int s0/1
gibson(config-if)#ipx network BB
```

4. Now we can test ping our network. This requires the use of the network number plus the ipx number. Since we do not have IPX enabled on our workstations we will have to rely upon ping from one router to the other. Let's see it in action!

gibson#ping ipx 0000.1234.1111
% Unrecognized host or address, or protocol not running.

See? We have to add the network (BB) number to ping with IPX:

gibson#ping ipx BB.0000.1234.1111

Type escape sequence to abort.
Sending 5, 100-byte IPXcisco Echoes to BB.0000.1234.1111, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/33/36 ms
gibson#

5. Let's look at some of the IPX show commands.

steve#sh ipx route
Codes: C - Connected primary network, c - Connected secondary network
S - Static, F - Floating static, L - Local (internal), W - IPXWAN
R - RIP, E - EIGRP, N - NLSP, X - External, A - Aggregate
s - seconds, u - uses, U - Per-user static

2 Total IPX routes. Up to 1 parallel paths and 16 hops allowed.

No default route known.

C **BB (HDLC),** Se0/0
C **100 (SAP),** Et0/0

Don't you just love it when everything works? We can see that we have used dynamic routing and have both our routes being advertised properly!

steve#sh ipx traffic
System Traffic for 0.0000.0000.0001 System-Name: steve
Rcvd: 33 total, 0 format errors, 0 checksum errors, 0 bad hop count,
3 packets pitched, 33 local destination, 0 multicast
Bcast: 18 received, 34 sent
Sent: 51 generated, 0 forwarded
0 encapsulation failed, 0 no route
SAP: 2 Total SAP requests, 0 Total SAP replies, 0 servers
2 SAP general requests, 0 ignored, 0 replies
0 SAP Get Nearest Server requests, 0 replies
0 SAP Nearest Name requests, 0 replies
0 SAP General Name requests, 0 replies
0 SAP advertisements received, 0 sent

0 SAP flash updates sent, 0 SAP format errors
RIP: 2 RIP requests, 0 ignored, 2 RIP replies, 2 routes
11 RIP advertisements received, 24 sent
4 RIP flash updates sent, 0 RIP format errors
Echo: Rcvd 10 requests, 5 replies
Sent 5 requests, 10 replies
0 unknown: 0 no socket, 0 filtered, 0 no helper
0 SAPs throttled, freed NDB len 0
Watchdog:
0 packets received, 0 replies spoofed
Queue lengths:
IPX input: 0, SAP 0, RIP 0, GNS 0
SAP throttling length: 0/(no limit), 0 nets pending lost route reply
Delayed process creation: 0
EIGRP: Total received 0, sent 0
Updates received 0, sent 0
Queries received 0, sent 0
Replies received 0, sent 0
SAPs received 0, sent 0
Trace: Rcvd 0 requests, 0 replies
Sent 0 requests, 0 replies

For good measure let's look on the ipx route on the other router too:

gibson#sh ipx route
Codes: C - Connected primary network, c - Connected secondary network
S - Static, F - Floating static, L - Local (internal), W - IPXWAN
R - RIP, E - EIGRP, N - NLSP, X - External, A - Aggregate
s - seconds, u - uses, U - Per-user static
2 Total IPX routes. Up to 1 parallel paths and 16 hops allowed.
No default route known.
C **BB (HDLC),** Se0/1
C **200 (SAP),** Et0/0

6. Now let's check out some debug commands with IPX:

gibson#debug ipx ?

all	IPX activity (all)
compression	IPX compression
eigrp	IPX EIGRP packets
ipxwan	Novell IPXWAN events
nlsp	IPX NLSP activity
packet	IPX activity
redistribution	IPX route redistribution
routing	IPX RIP routing information
sap	IPX Service Advertisement information
spoof	IPX and SPX Spoofing activity

```
gibson#debug ipx sap ?
  activity  IPX Service Advertisement packets
  activity  IPX Service Advertisement packets

gibson#debug ipx sap activity
IPX service debugging is on
gibson#
00:21:23: IPXSAP: positing update to 200.ffff.ffff.ffff via Ethernet0/0
(broadcast) (full)
00:21:30: IPXSAP: positing update to BB.ffff.ffff.ffff via Serial0/1
(broadcast) (full)
00:22:23: IPXSAP: positing update to 200.ffff.ffff.ffff via Ethernet0/0
(broadcast) (full)
00:22:30: IPXSAP: positing update to BB.ffff.ffff.ffff via Serial0/1
(broadcast) (full)
gibson#undebug all
```

Notice our SAP packets here and how often they are broadcast. These are what makes Novell a "chatty" network. Remember these SAP packets act like little children demanding attention: "Here I am! Here I am! Here I am!" Get it? Good.

Supplemental Lab or Challenge Activity:

1. Try adding Novell-ether as an encapsulation. Do you have to add it for every interface or is it on by default?
2. Add in a third router and keep the IPX party going!
3. Try switching the mask to 20 bits. What changes have to be made in your planning? Ok…now go do it!

So What Have I Learned Here?

In this lab you learned how to implement dynamic IPX routing along with dynamic IP routing. We just did this to make things easier even though it sounds complicated. There is not much IPX left in the "real world…" In some schools, some banks, and some companies but it seems to being dying out. Too bad…its really nice…it's even similar to IPv6…coincidence? I think not.

Guest Router Name

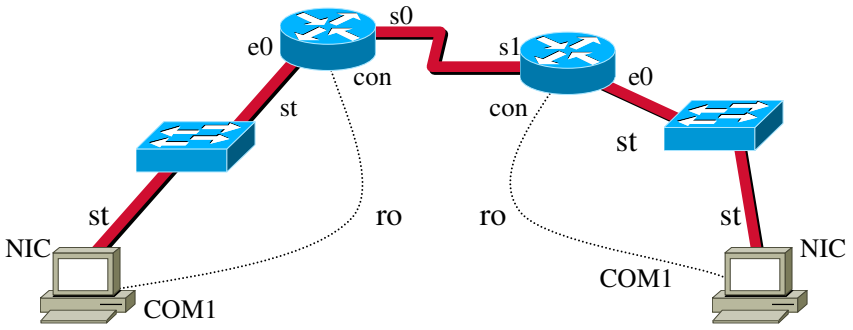
Steve Gibson runs a nice security website: http://www.grc.com His site will scan your system (free) and tell you of any security leaks or potential port problems. His site says they will not keep any information. Heck, it's a good start any many of my security colleagues frequently use the site.

Basic IP/IPX with Static Routing

Objective:
To learn how to set up basic IP/IPX between two routers on a network using static routing.

Tools and Materials:
(2) PC/workstations
(3) Routers
(2) Switches
(4) Straight-through cables
(2) DCE/DTE serial cable
(2) rollover cables

IP Lab Diagram:



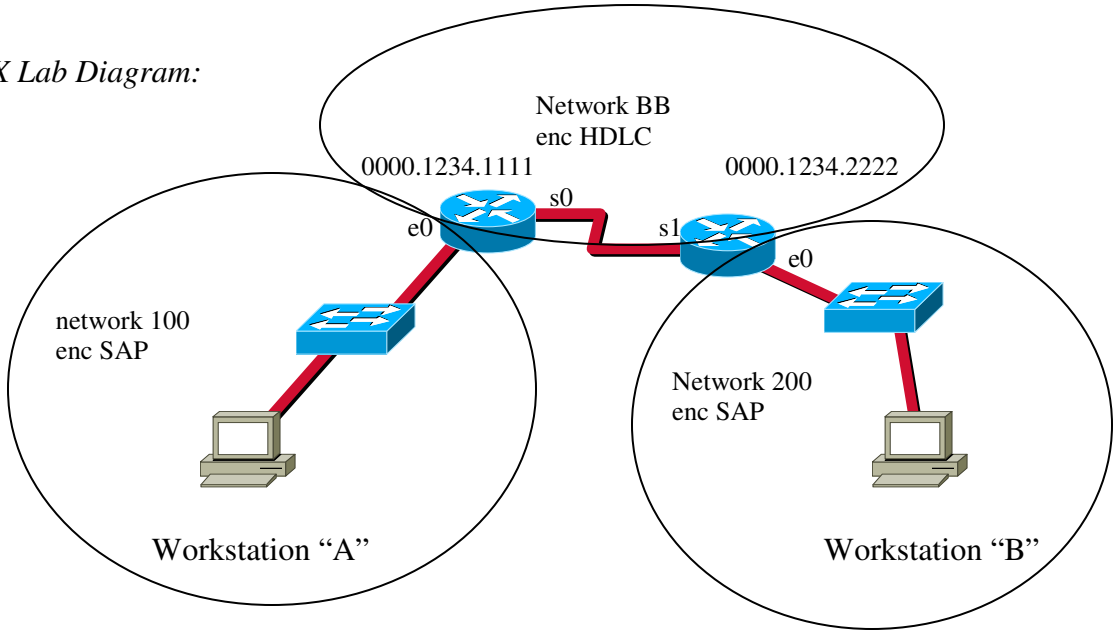
Workstation "A" Workstation "B"

IP Addressing:

Routers		
Hostnames	Steve	Gibson
E0	192.168.1.1/24	192.168.3.1/24
S0	192.168.2.1/24 (DCE)	n/a
S1	n/a	192.168.2.2/24 (DTE)
Workstations	A	B
IP	192.168.1.2	192.168.3.2
SM	255.255.255.0	255.255.255.0
GW	192.168.1.1	192.168.3.1

Step-By-Step Instructions:
1. Set up and cable the lab as shown. Enable IGRP as a routing protocol using autonomous system number 16. Use the same IPX settings from the last lab.
2. Test ping from workstation a to b.

IPX Lab Diagram:



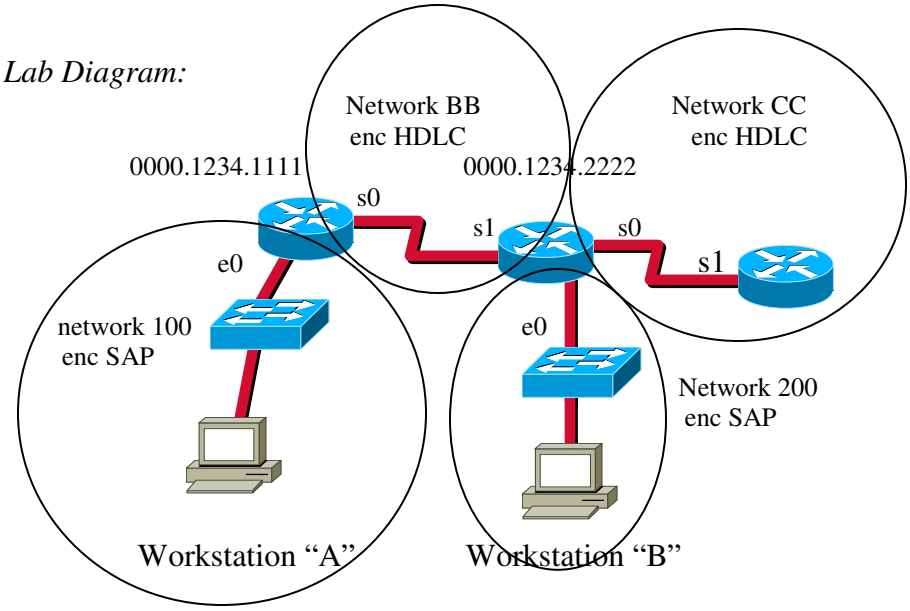
3. Now we can test ping our IPX network. This requires the use of the network number plus the ipx number. Since we do not have IPX enabled on our workstations we will have to rely upon ping from one router to the other. Let's see it in action!

```
gibson#ping ipx BB.0000.1234.1111

Type escape sequence to abort.
Sending 5, 100-byte IPXcisco Echoes to BB.0000.1234.1111, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/33/36 ms
gibson#
```

4. So how come it worked? Oh yeah. For any sort of static routing to take place we really have to use a third router here, otherwise everything is automatically routed over directly connected lines. So let's add a third router in:

```
gibson(config)#int s0/0
gibson(config-if)#ip address 192.168.4.1 255.255.255.0
gibson(config-if)#clockrate 56000
gibson(config-if)#network CC

staticIPX(config)#ipx routing 0000.1234.3333
staticIPX(config)#int s0/1
staticIPX(config-if)#ip address 192.168.4.2 255.255.255.0
staticIPX(config-if)#network CC
staticIPX(config-if)#no shut
```

IPX Lab Diagram:



5. Now let's try to ping from our new router all the way through:

```
staticIPX#ping ipx bb.0000.1234.2222

Type escape sequence to abort.
Sending 5, 100-byte IPXcisco Echoes to BB.0000.1234.2222, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/64/65 ms
staticIPX#
```

That's ok…we expected this…it is directly connected.

```
staticIPX#ping ipx bb.0000.1234.1111

Type escape sequence to abort.
Sending 5, 100-byte IPXcisco Echoes to BB.0000.1234.1111, timeout is 2
seconds:
.....
Success rate is 0 percent (0/5)
steve#
```

6. Next we need to add those static IPX routes in:

```
staticIPX(config)#ipx route BB CC.0000.1234.2222

steve(config)#ipx route CC BB.0000.1234.2222
```

Since gibson is directly connected to both we do not need any static routes here…if we added more routers into our network then we would need more.

7. Now let's try that ping IPX again:

```
staticIPX#ping ipx bb.0000.1234.1111

Type escape sequence to abort.
Sending 5, 100-byte IPXcisco Echoes to BB.0000.1234.1111, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/64/65 ms
staticIPX#
```

8. Let's also "See" those static IPX routes (highlighted):

```
staticIPX#sh ipx route
Codes: C - Connected primary network,    c - Connected secondary network
      S - Static, F - Floating static, L - Local (internal), W - IPXWAN
      R - RIP, E - EIGRP, N - NLSP, X - External, A - Aggregate
      s - seconds, u - uses, U - Per-user static

2 Total IPX routes. Up to 1 parallel paths and 16 hops allowed.
No default route known.
C        CC (HDLC),          Se0/1
S        BB via      CC.0000.1234.2222,        Se0/1

gibson#sh ipx route
Codes: C - Connected primary network,    c - Connected secondary network
      S - Static, F - Floating static, L - Local (internal), W - IPXWAN
      R - RIP, E - EIGRP, N - NLSP, X - External, A - Aggregate
      s - seconds, u - uses, U - Per-user static
4 Total IPX routes. Up to 1 parallel paths and 16 hops allowed.
No default route known.

C        BB (HDLC),          Se0/1
C        CC (HDLC),          Se0/0
C        200 (SAP),          Et0/0
S        100 via      BB.0000.1234.1111,        Se0/1

steve#sh ipx route
Codes: C - Connected primary network,    c - Connected secondary network
      S - Static, F - Floating static, L - Local (internal), W - IPXWAN
      R - RIP, E - EIGRP, N - NLSP, X - External, A - Aggregate
      s - seconds, u - uses, U - Per-user static

3 Total IPX routes. Up to 1 parallel paths and 16 hops allowed.
```

No default route known.

C BB (HDLC), Se0/0
C 100 (SAP), Et0/0
S CC via BB.0000.1234.2222, Se0/0
steve#

Supplemental Lab or Challenge Activity:

1. Try adding Novell-ether as an encapsulation.
2. Add in a fourth router and keep the IPX party going!
3. Try switching the mask to 20 bits. What changes have to be made in your planning? Ok…now go do it!

So What Have I Learned Here?

In this lab you learned how to implement static IPX routing along with dynamic IP routing. Next let's take a few moments to look more closely at our IPX show commands, debug commands, and encapsulation types.

Guest Router Name

Steve Gibson runs a nice security website: http://www.grc.com His site will scan your system (free) and tell you of any security leaks or potential port problems. His site says they will not keep any information. Heck, it's a good start any many of my security colleagues frequently use the site.

Paper Lab: Wildcard Masks

Objective:
To learn how to create wildcard masks for use with Access Control Lists.

Background:
People confuse wildcard masks with subnet masks all the time. They are similar after all because they both are masks but they really are different. A wildcard mask helps an access control list determine which ip addresses to implement the access control list commands upon. A nice, neat, simple rule: zero's denote "exact" match bits…think of that little razor knife: an "exact-o" knife. Let's dig in to an example:

1. Write a wildcard mask for a host ip address of 172.16.2.34:

   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:

   10101100.00010000.00000010.00100010

   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…)

   00000000.00000000.00000000.00000000

   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.

2. Write a wildcard mask for 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 here. 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.**001**00010
   *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:

   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.

3. Finally, unlike subnet masks, wildcard masks do not have to be contiguous (all zeros in a row)…we can mask out certain ips. Write a wildcard mask for a the odd numbered ips 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.**001**00010
   *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 except the last bit (this determines odd or even…it's the "1" bit):

   00000000.00000000.00000000.00011110

   Therefore, when we convert this wildcard mask back to decimal we get a wildcard mask of 0.0.0.30 for our subnet wildcard mask. 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.

Supplemental Labs or Challenge Activities:

1. 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.
2. Write a wildcard mask that will mask the host portions of the 192.168.1.0/24 network. We are looking for an exact match of the host portions only.
3. 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.
4. Write a wildcard mask that will mask the upper half of the 10.128.0.0/11 network. Here we will mask 129-255 in the second octet and 0-255 in the third and fourth octets (so we need exact matches for them).
5. Write a wildcard mask that will mask the lower half of the 200.210.128.0/27 network. Here we need to mask the lower half of the host portion for the 128 subnet.

as I said these are subjective in respect to the needs of the access control list

Paper Lab: Access Control Lists

Objective:
To learn the fundamentals of writing standard, extended, and named Access Control List statements.

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. 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. 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. You will learn about 3 of the basic ACL's: Standard, Extended, and Named. Two of the other ACL's you will learn about in CCNP school are Dynamic (a.k.a "Lock and Key") and Reflexive. A standard ACL controls access using an IP address or range of addresses. An extended ACL controls access to specific ports for IP addresses. A named ACL uses a name instead of a number to do the same thing as standard or extended ACL's

We have some very simple rules to follow when creating ACL's on your router. We have already discussed the first:

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.

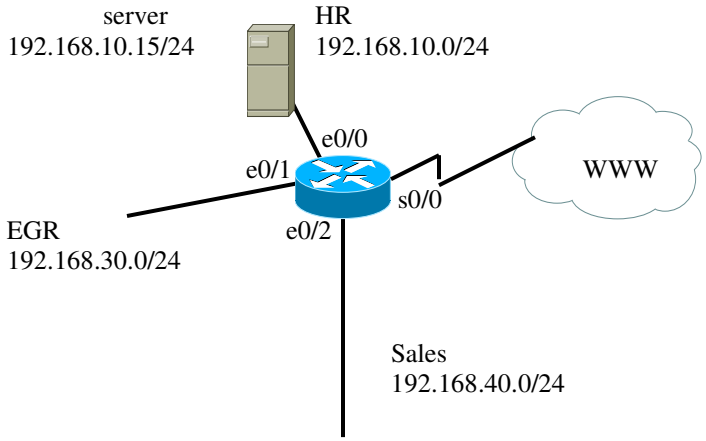
Access Control Lists are also numbered. We have different numbers for our different purposes, protocols, and types of ACL's. Let's look at those numbers now:

1-99	IP standard
100-199	IP extended
200-299	Protocol type-code
300-399	DECnet
400-499	XNS standard
500-599	XNS extended
600-601	Appletalk
700-799	48-bit MAC address
800-899	IPX standard
900-999	IPX extended
1000-1099	IPX SAP
1100-1199	Extended 48-bit MAC address
1200-1299	IPX summary address

Table 1—ACL numbering.

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 server, but allow them access to the marketing network and the WWW.



Now lets create our ACL:

```
Router(config)#access-list 1 deny 192.168.40.0 0.0.0.255
Router(config)#access-list 1 permit ip 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's (in this case a network) and the wildcard mask. In this mask we wanted to exactly match the network and subnet portion and didn't really care about the host portions. Therefore our mask became 0.0.0.255 (nnnnnnnn.nnnnnnnn.ssssssss.hhhhhhhh).

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. 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:
   a. Router(config)#access-list 100 deny tcp 192.168.30.0 0.0.0.255 any eq 23
   b. Router(config)#access-list 100 permit ip any any

(2) Apply the ACL to an interface:
   a. Router(config)int e0/2
   b. 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:
   a. Router(config)#access-list 100 deny icmp 192.168.30.0 0.0.0.255
   b. Router(config)#access-list 100 permit icmp any any

(2) Apply the ACL to an interface:
   a. Router(config)int e0/2
   b. 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. 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:
   a. Router(config)#ip access-list standard no_salesHR
   b. Router(config-std-nacl)#deny 192.168.40.0 0.0.0.255
   c. Router(config-std-nacl)#permit ip any

(2) Apply the ACL to an interface:
   a. Router(config)#interface e0/0
   b. Router(config-if)#ip access-group no_salesHR out

Supplemental Lab or Challenge Activity:
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.

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 192.168.30.0 0.0.0.255
   Router(config)#access-list 101 permit ip any any
   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 101 deny tcp 192.168.0.0 0.0.255.255 any eq 23
   Router(config)#access-list 101 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

   Correct? _____
   Incorrect? _____
   What's wrong? _____
   _____
   _____
   _____

Now let's try to "free-hand" some ACL's

1. 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.
2. Write a named ACL to do the same thing.
3. Write an extended ACL to deny FTP access to everyone in the network.
4. Write a named ACL to allow only the EGR network to have www access.
5. 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?
6. Write an extended ACL to allow only the HR people with odd numbered ip addresses to have the ability to use FTP.

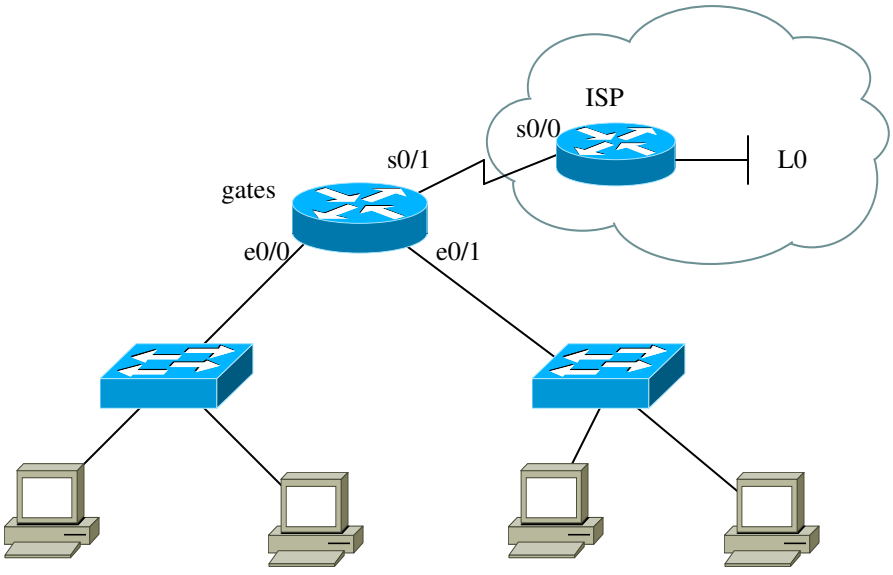
So What Have I Learned Here?

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.

Objective:
To implement a standard access control list on a simple network.

Tools and Materials:
(4) workstations
(6) straight-through cables
(2) routers
(1) DCE/DTE cable
(2) switches (or one switch with 2 VLAN's)

Lab Diagram:



	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

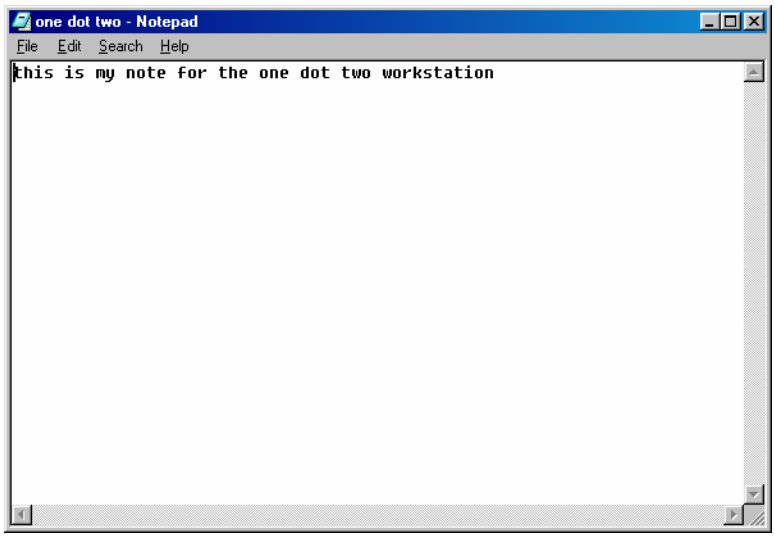
Addressing

Router	Gates	ISP
S0/0 (DCE)	n/a	192.168.2.1
S0/1 (DTE)	192.168.2.2	n/a
E0/0	192.168.1.1	n/a
E0/1	192.168.3.1	n/a
L0	n/a	172.16.1.1/16

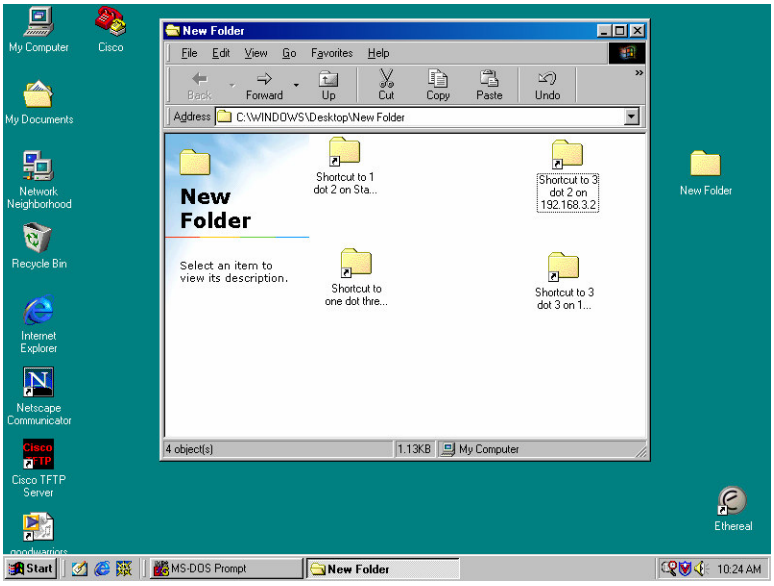
Step-By-Step Instructions:

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.
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:



Repeat for each workstation. Put a shortcut for each desktop folder on each workstation. It should look like this:



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:

    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

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)#int e0/1
    gates(config-if)#ip access-group 10 out

8. 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

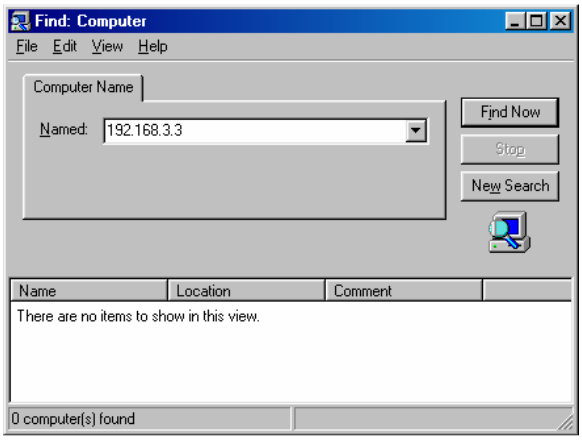
    Pinging 192.168.3.3 with 32 bytes of data:

    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.

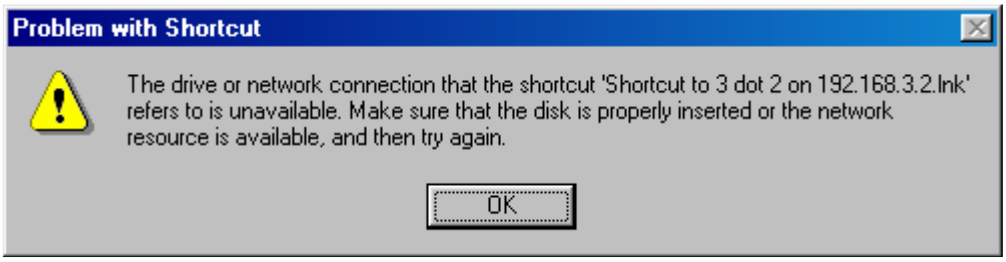
    Ping statistics for 192.168.3.3:
      Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
      Minimum = 0ms, Maximum = 0ms, Average = 0ms

    C:\WINDOWS\Desktop>

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. 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 ip 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!

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
gates#
```

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
18:32:06: IP: s=192.168.1.1 (local), d=192.168.1.2 (Ethernet0/0), len 56, sending
18:32:06: ICMP type=3, code=13
18:32:07: IP: s=192.168.1.2 (Ethernet0/0), d=192.168.3.2 (Ethernet0/1), len 60, access denied
18:32:07: ICMP type=8, code=0
18:32:07: IP: s=192.168.1.1 (local), d=192.168.1.2 (Ethernet0/0), len 56, sending
18:32:07: ICMP type=3, code=13
18:32:08: IP: s=192.168.1.2 (Ethernet0/0), d=192.168.3.2 (Ethernet0/1), len 60, access denied
18:32:08: ICMP type=8, code=0
18:32:08: IP: s=192.168.1.1 (local), d=192.168.1.2 (Ethernet0/0), len 56, sending
18:32:08: ICMP type=3, code=13
gates#

Type 8 with code 0 is for an ICMP echo.
Type 3 with code 13 is for "administratively prohibited."

Supplemental Lab or Challenge Activity:

1. Design a network using 3 or more routers using a different routing protocol than RIPv2. Vary the IP address classes. Deny certain subnets access to each other.
2. Design a network for a school that uses even-numbered IP addresses for teachers and odd-numbered IP addresses for students. Allow only the teachers to be able to use the WWW and telnet. Set up students for web access but without DNS capabilities.
3. Go out to CISCO and find out other type and codes for ACL's using the debug ip packet detail command.

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.

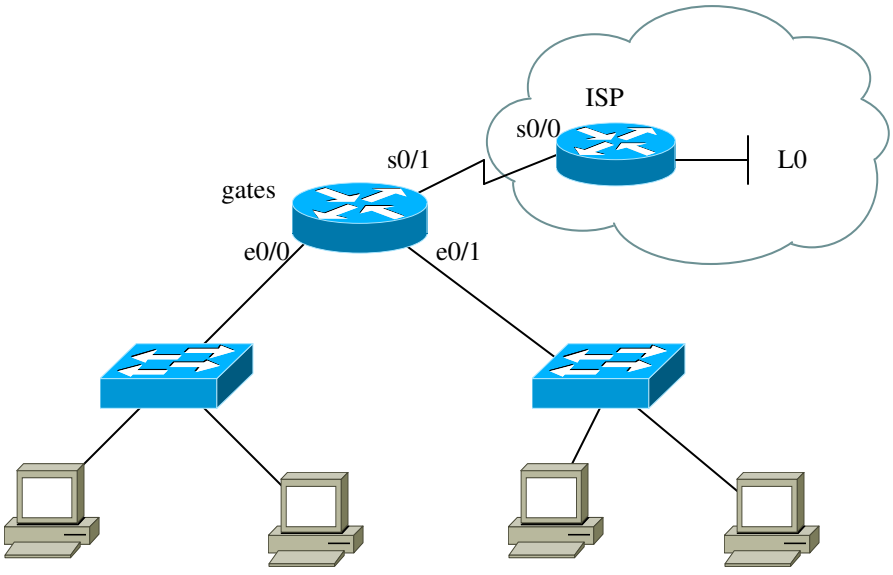
Guest Router Name

Gates…if you haven't guessed it by now…Bill Gates is one of the founders of Microsoft—the world's largest computer software empire. Microsoft is one of the most hated targets of hackers because of the closed source code. On the other hand they say Apple's are almost un-hackable…mainly because hackers do not use Apples, do not care about Apples, and never really will as long as Microsoft is around. TNT had a good biography on Bill Gates with Anthony M. Hall as Bill Gates. See it if you get the chance.

Objective:
To implement an extended access control list on a simple network.

Tools and Materials:
(4) workstations
(6) straight-through cables
(2) routers
(1) DCE/DTE cable
(2) switches (or one switch with 2 VLAN's)

Lab Diagram:



	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	Gates	ISP
S0/0 (DCE)	n/a	192.168.2.1
S0/1 (DTE)	192.168.2.2	n/a
E0/0	192.168.1.1	n/a
E0/1	192.168.3.1	n/a
L0	n/a	172.16.1.1/16

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.
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 "ip 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
```

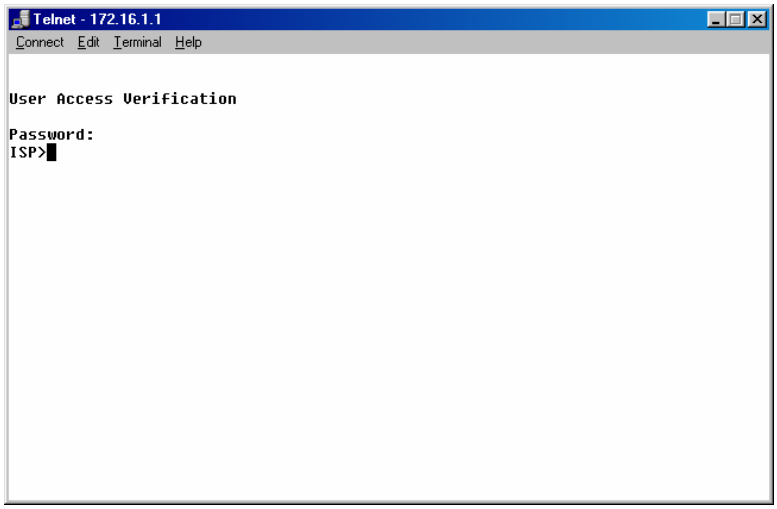
   Let's look at our statement. We set up ACL 138 to deny ICMP from (source) host 192.168.1.2 to (dest) 192.168.3.0 (network) with a wildcard mask to match the network 0.0.0.255.

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:



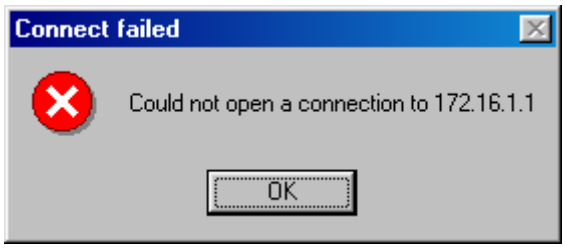
10. Now let's create the extended ACL:

```
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:



Supplemental Lab or Challenge Activity:

1. Design a network using 3 or more routers using a different routing protocol than RIPv2. Vary the IP address classes. Deny telnet access to everyone.
2. Go out to CISCO and find out other port numbers for extended ACL's.
3. Try using a protocol inspector to capture packets.

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.

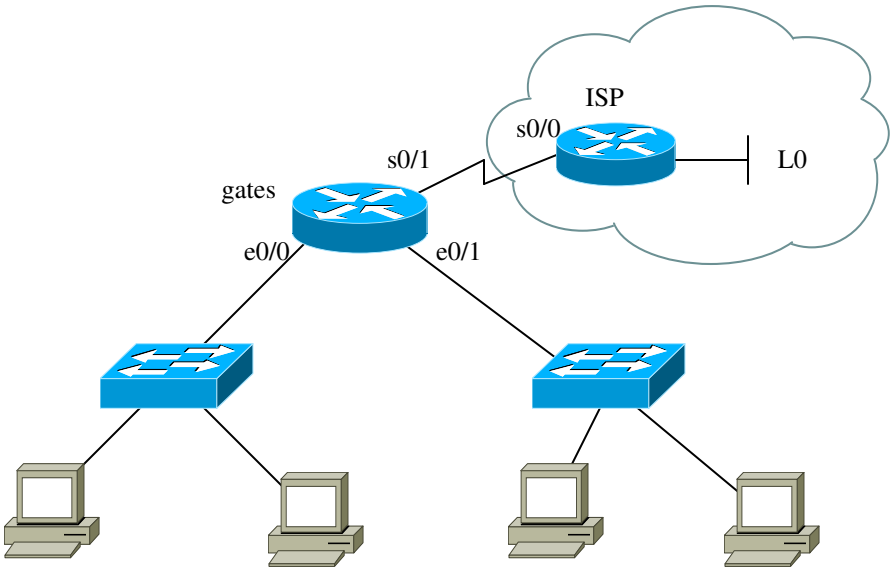
Guest Router Name

Gates…if you haven't guessed it by now…Bill Gates is one of the founders of Microsoft—the world's largest computer software empire. Microsoft is one of the most hated targets of hackers because of the closed source code. On the other hand they say Apple's are almost un-hackable…mainly because hackers do not use Apples, do not care about Apples, and never really will as long as Microsoft is around. TNT had a good biography on Bill Gates with Anthony M. Hall as Bill Gates. See it if you get the chance.

Objective:
To implement a named access control list on a simple network.

Tools and Materials:
(4) workstations
(6) straight-through cables
(2) routers
(1) DCE/DTE cable
(2) switches (or one switch with 2 VLAN's)

Lab Diagram:



	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	Gates	ISP
S0/0 (DCE)	n/a	192.168.2.1
S0/1 (DTE)	192.168.2.2	n/a
E0/0	192.168.1.1	n/a
E0/1	192.168.3.1	n/a
L0	n/a	172.16.1.1/16

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:

```
gates(config)#access-list extended no_ping
gates(config-ext-nac;)#deny icmp host 192.168.1.2 192.168.3.0 0.0.0.255
gates(config-ext-nacl)#permit icmp any any
gates(config-ext-nacl)#exit
gates(config)#ip access-list extended no_telnet
gates(config-ext-nacl)#deny tcp host 192.168.3.2 any eq 23
gates(config-ext-nacl)#permit tcp any any
```

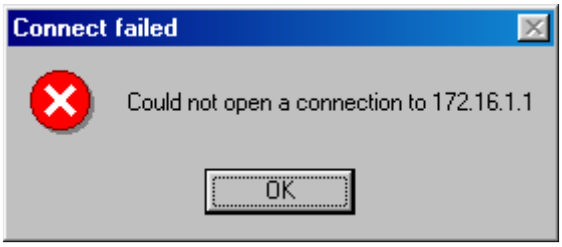
4. Step 2: apply the ACL to an interface. Since this is an extended ACL it should be placed nearest the source as possible using "in."

```
gates(config)#int e0/0
gates(config-if)#ip access-group 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:



Supplemental Lab or Challenge Activity:

1. Why would you want to use named ACL's instead of numbered ACL's?
2. Can you use the same name ACL on different routers in the same network?

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.

Guest Router Name

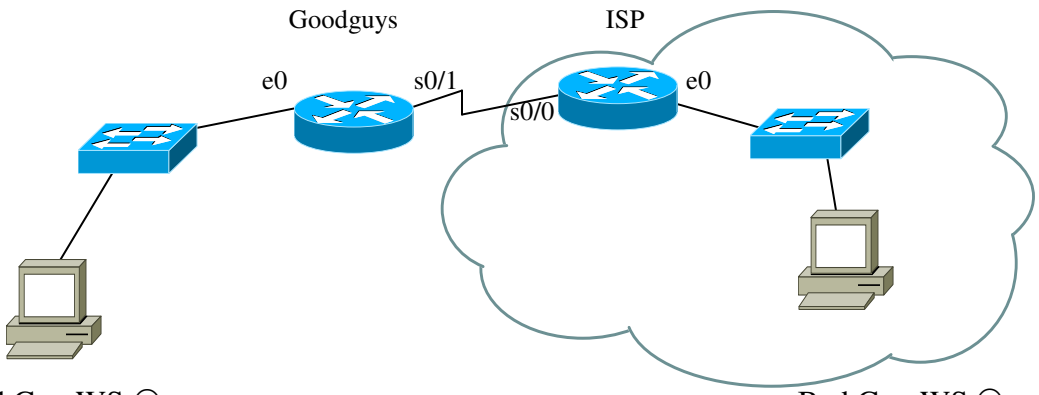
Gates…if you haven't guessed it by now…Bill Gates is one of the founders of Microsoft—the world's largest computer software empire. Microsoft is one of the most hated targets of hackers because of the closed source code. On the other hand they say Apple's are almost un-hackable…mainly because hackers do not use Apples, do not care about Apples, and never really will as long as Microsoft is around. TNT had a good biography on Bill Gates with Anthony M. Hall as Bill Gates. See it if you get the chance.

Making a Protocol Inspector with ACL's

Objective:
To learn how to use ACL's to build a mini-protocol inspector.

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:



Good Guy WS ☺
192.168.1.2/24

Bad Guy WS ☹
172.16.1.2/16

Addressing:

Router	goodguys	ISP
S0/0 (DCE)	n/a	220.100.50.1/24
S0/1 (DTE)	220.100.50.2/24	n/a
E0/0	192.168.1.1/24	172.16.1.1/16

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 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 caused 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 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

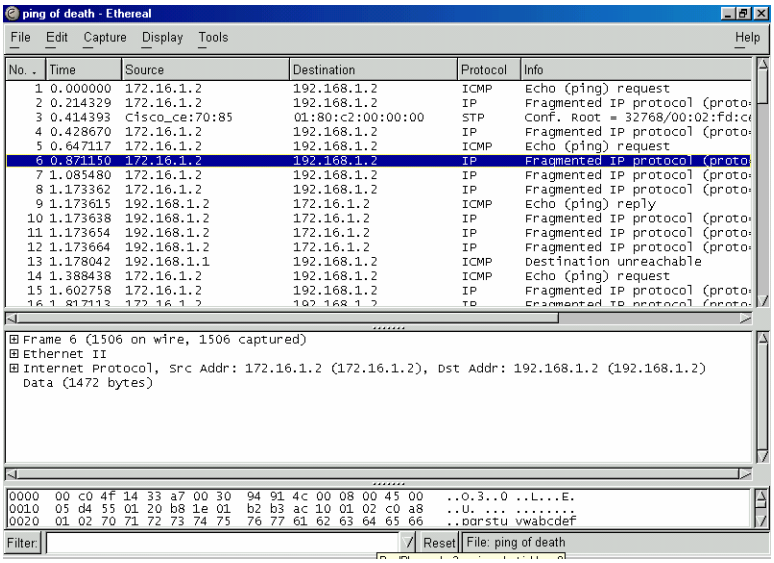
   You should see something like this:

   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
00:34:59: %SYS-5-CONFIG_I: Configured from console by console
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**
goodguys#

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.

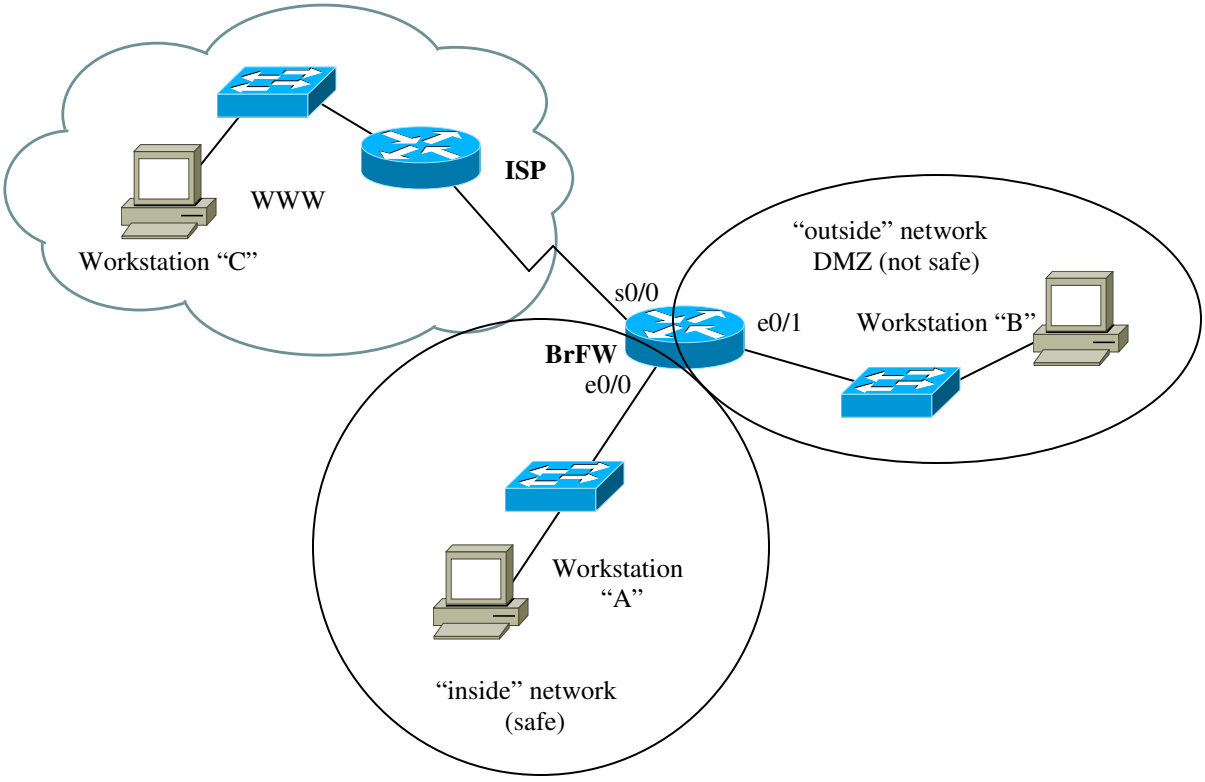
So What Have I Learned Here?

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. We are wrapping up this section and moving in to WAN's…this stuff is fun, isn't it?

Objective:
To learn how a router can be set up as a mini-firewall using 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:

Router	BrFW	ISP
S0/0	214.72.83.12/24 (DTE)	214.72.83.11/24 (DCE)
E0/0	10.0.1.1/24	50.0.1.1/24
E0/0	206.16.1.1/24	n/a

Workstations	A	B	C
IP	10.0.1.2/24	202.16.1.2/24	50.0.1.2/24
GW	10.0.1.1	202.16.1.1	50.0.1.1

Background:
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 is the front and rear views of a CISCO PIX Firewall.

"front view" of PIX 515

"rear view" of PIX 515

Not too different huh? In this lab you will learn about a fourth 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)#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:

```
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.

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?
3. What are dynamic access control lists? How could I use them here?

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.

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.

Prompt	Command	Shortcut	Description

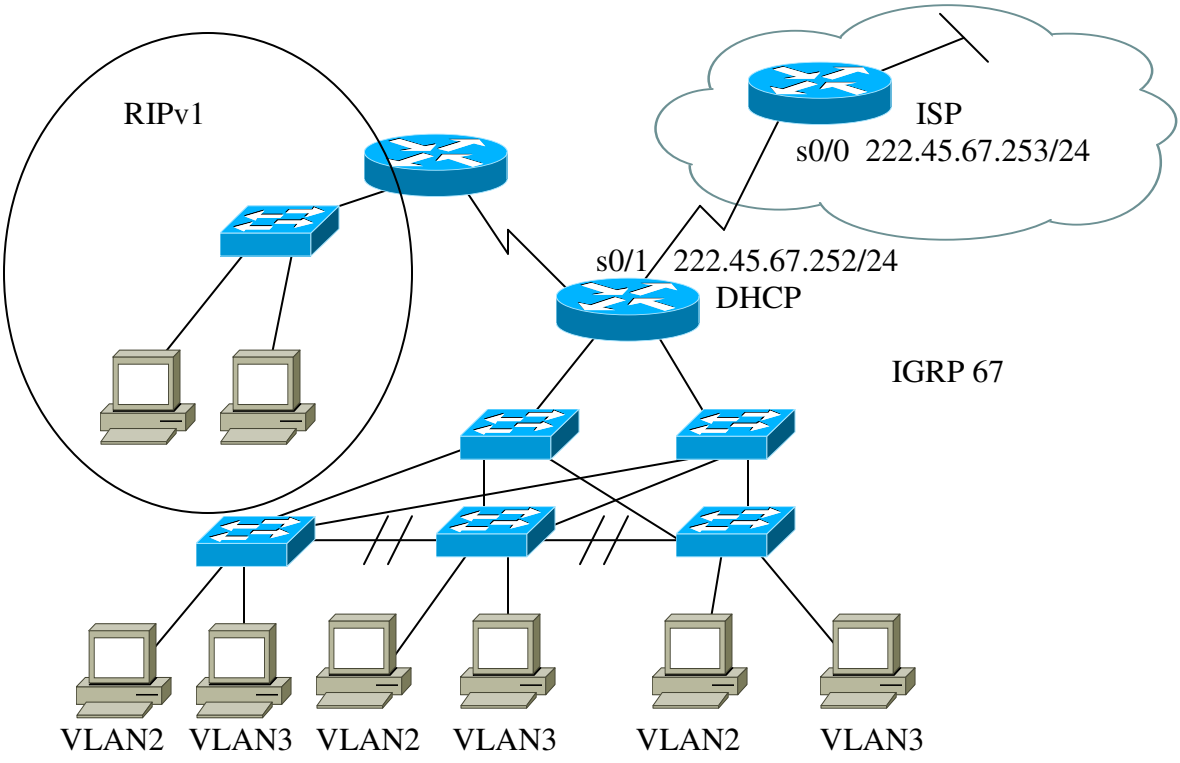
Prompt	Command	Shortcut	Description

Whole Enchilada/Crazy Insano Lab #1 (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:

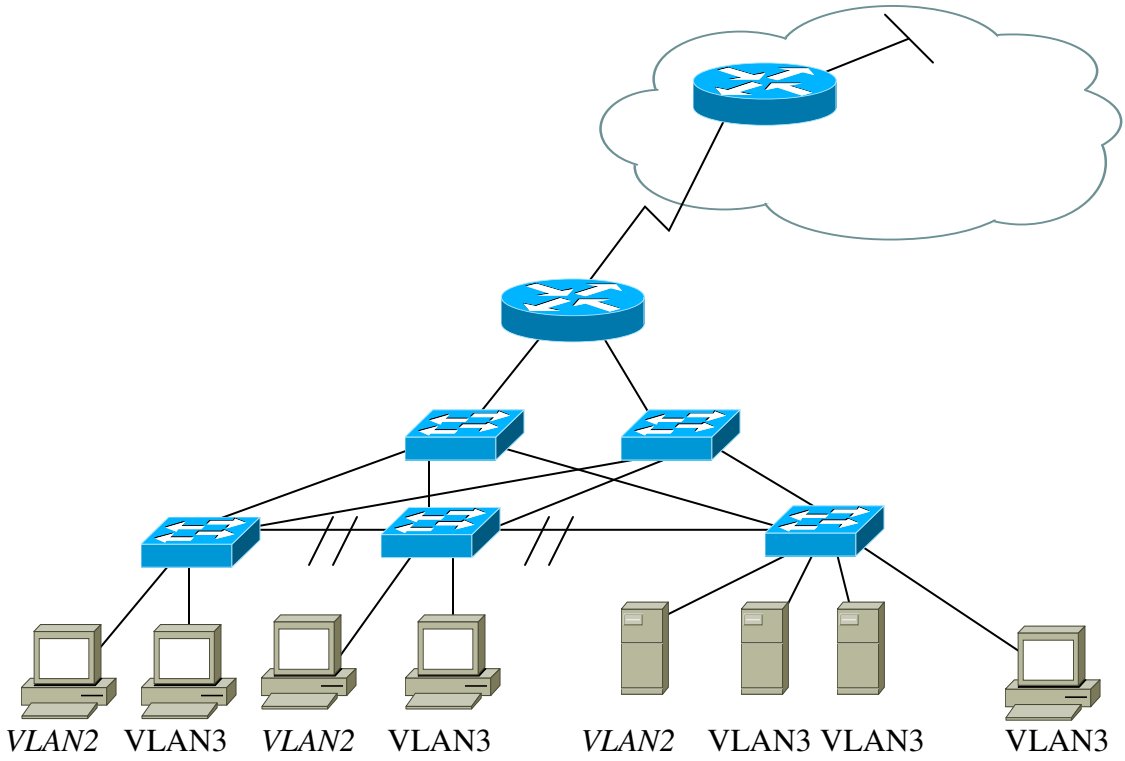


Whole Enchilada/Crazy Insano Lab #2 (WECIL): IP/IPX

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 5:
Wide Area Network Routing

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.

How much does it cost? $125 per attempt for each test.

What is a passing score? For CCNA 849 of 1000 is a passing score. This are about 45-55 questions to complete in 75 minutes.

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 eight sections: Bridging/Switching, OSI reference model & layered communications, network protocols, routing, WAN protocols, network management, lan design, and CISCO basics, IOS and network basics. 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, etc. You will need two picture ID's.

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.

For which test do I register? You are being prepared for the Routing and Switching tracks. Currently, for the CCNA you should register for CCNA 3.0 640-607.

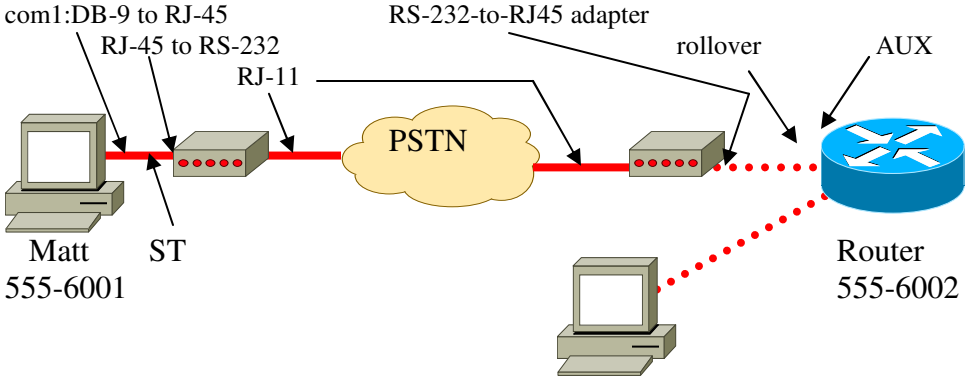
Visit http://www.cisco.com/warp/public/10/wwtraining/certprog/ for more information and to demo a course simulation.

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:



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?

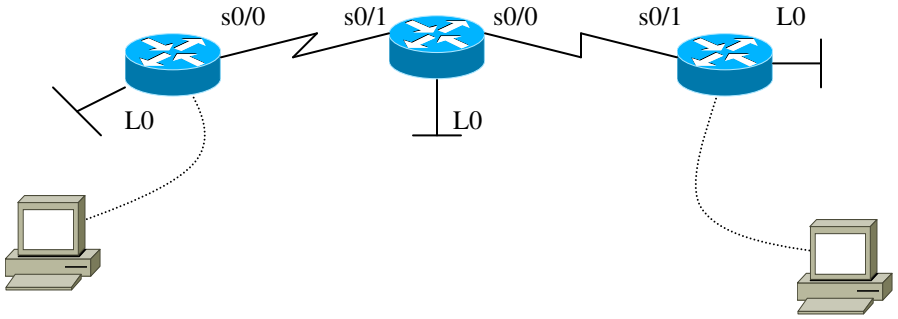
So What Did I Learn Here?

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.

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:



Name:	Terminus	Leftist	Urvile
S0/0	200.200.200.1/24	201.200.200.1/24	n/a
S0/1	n/a	200.200.200.2/24	201.200.200.2/24
L0	10.0.0.1/8	11.0.0.1/8	12.0.0.1/8

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 ?
```

You should see:

```
Router(config-if)#enc ?
  Atm-dxi          ATM-DXI encapsulation
  bstun            Block Serial tunneling (BSTUN)
  frame-relay      Frame Relay networks
  hdlc             Serial HDLC synchronous
  lapb             LAPB (X.25 Level 2)
  ppp              Point-to-Point protocol
  sdlc             SDLC
  sdlc-primary     SDLC (primary)
  sdlc-secondary   SDLC (secondary)
  smds             Switched Megabit Data Service (SMDS)
  stun             Serial Tunneling (STUN)
  x25              X.25
```

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. Geeze. 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, Serial0/0
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
terminus#
```

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

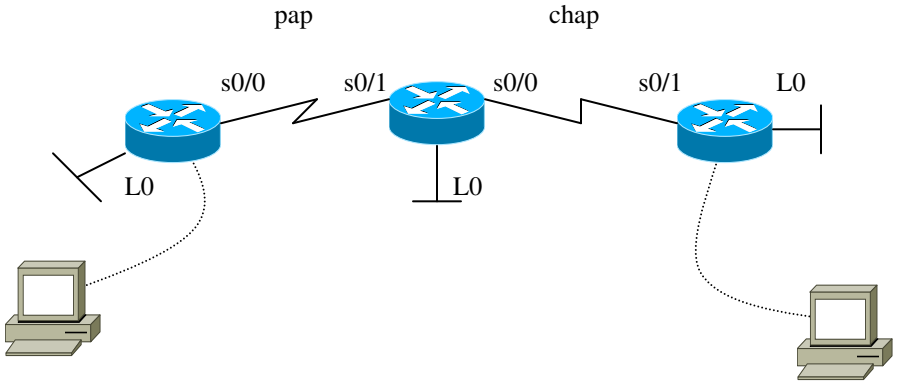
Terminus, Leftist, and Urvile were three hackers from the Legion of Doom, who lived in Georgia, that were busted in 1990 by the U.S. Secret Service in connection with the Martin Luther King Day AT&T long distance network crash. They were known as "switching gurus" and as "heavy hitters" within the LoD because they frequently accessed BellSouth's network. Apparently BellSouth, at that time, did not have very strict security in place.

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:



Name:	Terminus	Leftist	Urvile
S0/0	200.200.200.1/24	201.200.200.1/24	n/a
S0/1	n/a	200.200.200.2/24	201.200.200.2/24
L0	10.0.0.1/8	11.0.0.1/8	12.0.0.1/8

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 **PPP: Phase is ESTABLISHING, Active Open**
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
00:54:36: Se0/0 LCP: MagicNumber 0x04158C08 (0x050604158C08)
00:54:36: Se0/0 **LCP: State is Open**
00:54:36: Se0/0 PPP: **Phase is UP**

00:54:36: Se0/0 **IPCP: O CONFREQ [Closed]** id 3 len 10
00:54:36: Se0/0 **IPCP: Address 200.200.200.1** (0x0306C8C8C801)
00:54:36: Se0/0 **CDPCP: O CONFREQ [Closed]** id 3 len 4
00:54:36: Se0/0 **IPCP: I CONFREQ [REQsent]** id 4 len 10
00:54:36: Se0/0 **IPCP: Address 200.200.200.2** (0x0306C8C8C802)
00:54:36: Se0/0 **IPCP: O CONFACK [REQsent]** id 4 len 10
00:54:36: Se0/0 **IPCP: Address 200.200.200.2** (0x0306C8C8C802)
00:54:36: Se0/0 **CDPCP: I CONFREQ [REQsent]** id 4 len 4
00:54:36: Se0/0 **CDPCP: O CONFACK [REQsent]** id 4 len 4
00:54:36: Se0/0 **IPCP: I CONFACK [ACKsent]** id 3 len 10
00:54:36: Se0/0 **IPCP: Address 200.200.200.1** (0x0306C8C8C801)
00:54:36: Se0/0 **IPCP: State is Open**
00:54:36: Se0/0 **CDPCP: I CONFACK [ACKsent]** id 3 len 4
00:54:36: Se0/0 **CDPCP: State is Open**
00:54:36: Se0/0 **IPCP: Install route to 200.200.200.2**
00:54:37: %LINEPROTO-5-UPDOWN: **Line protocol on Interface Serial0/0, changed state to up**
01:02:50: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up

Ok…all those numbers and stuff can be confusing. Let's strip that output down to just the information in bold 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 Serial0/0, changed state to down
11. **(line is reconnected)**
12. Interface Serial0/0, 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]
23. IPCP: I CONFREQ [REQsent]
24. IPCP: Address 200.200.200.2
25. IPCP: O CONFACK [REQsent]
26. IPCP: Address 200.200.200.2
27. CDPCP: I CONFREQ [REQsent]

28. CDPCP: O CONFACK [REQsent]
29. IPCP: I CONFACK [ACKsent]
30. IPCP: Address 200.200.200.1
31. IPCP: State is Open
32. CDPCP: I CONFACK [ACKsent]
33. CDPCP: State is Open
34. IPCP: Install route to 200.200.200.2
35. Line protocol on Interface Serial0/0, changed state to up

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.

```
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):

```
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 (0x050601DE98E6)
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 (0x0506042BA23D)
01:18:42: Se0/0 LCP: I CONFREJ [ACKsent] id 109 len 8
01:18:42: Se0/0 LCP:    AuthProto PAP (0x0304C023)
(redundant lines removed)
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 (0x0506042BA23D)
01:18:42: Se0/0 LCP: I TERMACK [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 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)
21. 01:24:46: Se0/1 LCP: O CONFACK [REQsent] id 187 len 14
22. 01:24:46: Se0/1 LCP: AuthProto PAP (0x0304C023)
23. 01:24:46: Se0/1 LCP: MagicNumber 0x04318B4D (0x050604318B4D)
24. 01:24:46: Se0/1 LCP: I CONFACK [ACKsent] id 76 len 14
25. 01:24:46: Se0/1 LCP: AuthProto PAP (0x0304C023)
26. 01:24:46: Se0/1 LCP: MagicNumber 0x01E48949 (0x050601E48949)
27. 01:24:46: Se0/1 LCP: State is Open
28. 01:24:46: Se0/1 **PPP: Phase is AUTHENTICATING, by both**
29. 01:24:46: Se0/1 **PAP: O AUTH-REQ id 2 len 21 from "prophet"**
30. 01:24:46: Se0/1 **PAP: I AUTH-REQ id 2 len 21 from "prophet"**
31. 01:24:46: Se0/1 **PAP: Authenticating peer prophet**
32. 01:24:46: Se0/1 **PAP: O AUTH-ACK id 2 len 5**
33. 01:24:46: Se0/1 **PAP: I AUTH-ACK id 2 len 5**
34. 01:24:46: Se0/1 PPP: Phase is UP
35. 01:24:46: Se0/1 IPCP: O CONFREQ [Closed] id 8 len 10
36. 01:24:46: Se0/1 IPCP: Address 200.200.200.2 (0x0306C8C8C802)
37. 01:24:46: Se0/1 CDPCP: O CONFREQ [Closed] id 8 len 4
38. 01:24:46: Se0/1 IPCP: I CONFREQ [REQsent] id 7 len 10
39. 01:24:46: Se0/1 IPCP: Address 200.200.200.1 (0x0306C8C8C801)
40. 01:24:46: Se0/1 IPCP: O CONFACK [REQsent] id 7 len 10
41. 01:24:46: Se0/1 IPCP: Address 200.200.200.1 (0x0306C8C8C801)
42. 01:24:46: Se0/1 CDPCP: I CONFREQ [REQsent] id 7 len 4
43. 01:24:46: Se0/1 CDPCP: O CONFACK [REQsent] id 7 len 4
44. 01:24:46: Se0/1 IPCP: I CONFACK [ACKsent] id 8 len 10
45. 01:24:46: Se0/1 IPCP: Address 200.200.200.2 (0x0306C8C8C802)
46. 01:24:46: Se0/1 IPCP: State is Open
47. 01:24:46: Se0/1 CDPCP: I CONFACK [ACKsent] id 8 len 4
48. 01:24:46: Se0/1 CDPCP: State is Open
49. 01:24:46: Se0/1 IPCP: Install route to 200.200.200.1
50. 01:24:47: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed state to up

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.

```
leftist#debug ppp authentication
PPP authentication debugging is on
04:26:10: %LINK-3-UPDOWN: Interface Serial0/1, changed state to down
04:26:11: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed state to down
```

```
04:26:16: Se0/1 PPP: Treating connection as a dedicated line
04:26:16: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
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 PAP: O AUTH-ACK id 32 len 5
04:26:16: Se0/1 PAP: I AUTH-ACK id 32 len 5
04:26:17: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/1, changed state to up
leftist#
```

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 "undebug 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)#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

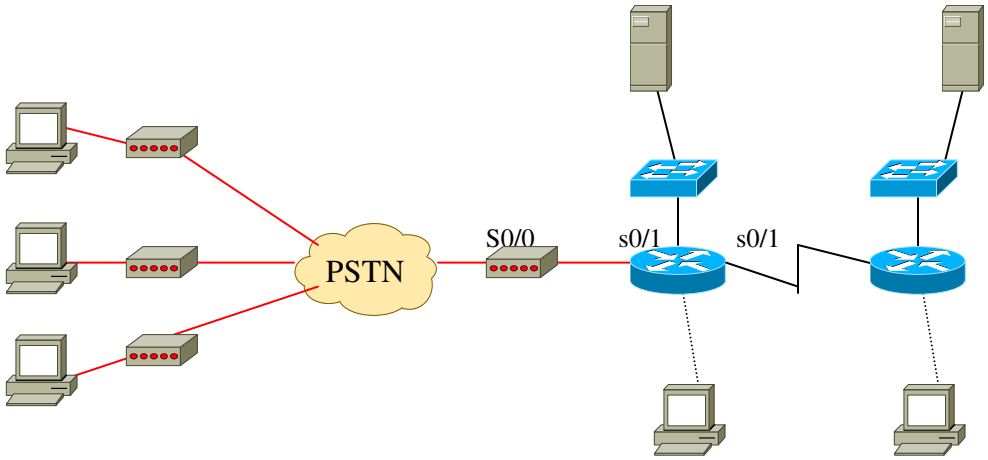
Terminus, Leftist, and Urvile were three hackers from the Legion of Doom, who lived in Georgia, that were busted in 1990 by the U.S. Secret Service in connection with the Martin Luther King Day AT&T long distance network crash. They were known as "switching gurus" and as "heavy hitters" within the LoD because they frequently accessed BellSouth's network. Apparently BellSouth, at that time, did not have very strict security in place.

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
(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:



Router	Dark	Lord
S0/0	192.168.1.1/24	n/a
S0/1	192.168.2.1/24	192.168.2.2/24
Loop0	1.1.1.1/8	2.2.2.2/8
IP Network	1.0.0.0/8	2.0.0.0/8
IPX Network	100 (802.3)	200 (802.3)
Router IPX	0000.AAAA.0001	0000.BBBB.0002
FileServer	1000.0000.0000.0001	2000.0000.0000.0002

S0/1-S0/1 IPX Network 192 (802.3)

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
lord(config-if)#int s0/1
lord(config-if)#ipx network 192
lord(config-if)#ipx sap-interval 0
```

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.

Guest Router Name Derivation

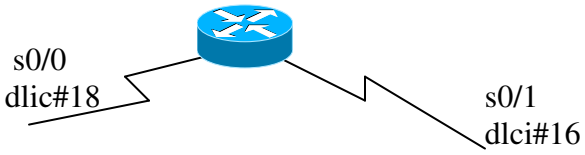
Jason Allen Diekman, a.k.a. "Shadow Knight" or "Dark Lord," was charged with hacking into Nasa, Oregon State Univeristy, and a San Francisco area ISP in 2002. He was sentenced to 21 months in Federal Prison, ordered to pay restitution of $87,736.29, and will have 3 years of probation, which includes 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
Frswitch(config-if)#frame-relay intf-type dce
Frswitch(config-if)#clockrate 56000
Frswitch(config-if)#no shut

Frswitch(config)#int s0/1
Frswitch(config-if)#enc frame-relay
Frswitch(config-if)#frame-relay intf-type dce
Frswitch(config-if)#clockrate 56000
Frswitch(config-if)#no shut
```

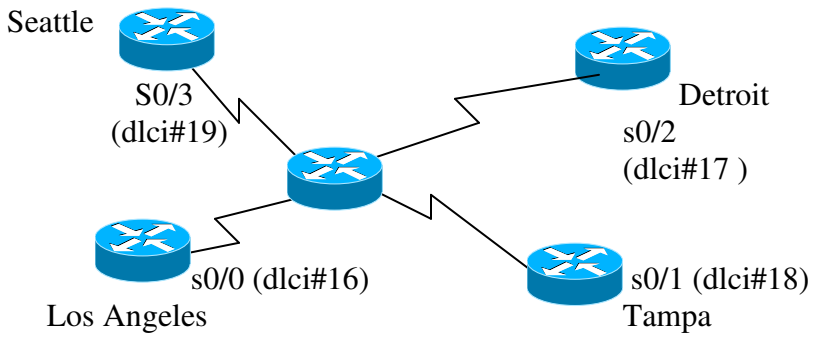
4. Then configure a dlci-switching table on EACH interface.

```
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
```

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:



So what have I learned here?

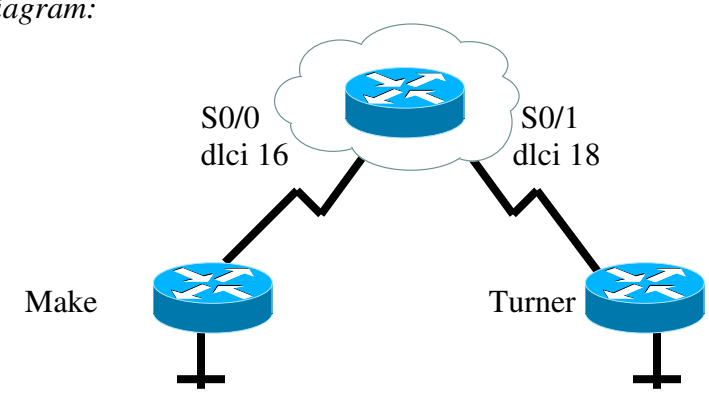
In this lab you learned how to turn a router into a frame relay switch. Would you do this inside a company? Almost always no. Your serial lines with HDLC can provide clocking to move the information. In the next couple of labs you will learn about configuring different topologies with frame relay networking.

Basic Frame Relay With Two Routers

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:



Router	Make	Turner
S0/0	192.168.1.2/24	192.168.1.1/24
Dce/dte	dte	dte
Dlci	16	18
Loop 0	2.2.2.2/8	1.1.1.1/8

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.

```
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.

```
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:

```
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
turner#

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
turner#
```

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
turner #
```

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 (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

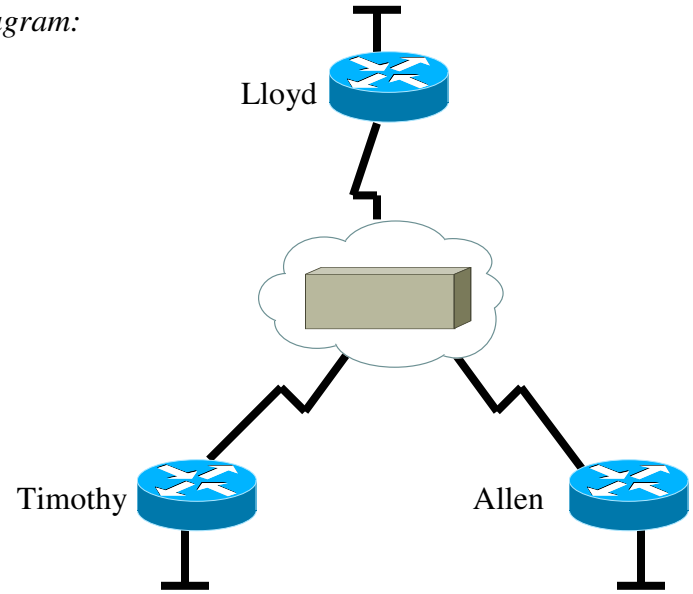
Frame Relay: Hub and Spoke with 3 routers

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 17 192.168.20.2-dlci 18	192.168.20.1-dlci 16	192.168.20.1-dlci16

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:"

```
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:

```
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:

```
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 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 changed 00:19:50
```

Notice you are on "timothy" 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. 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?

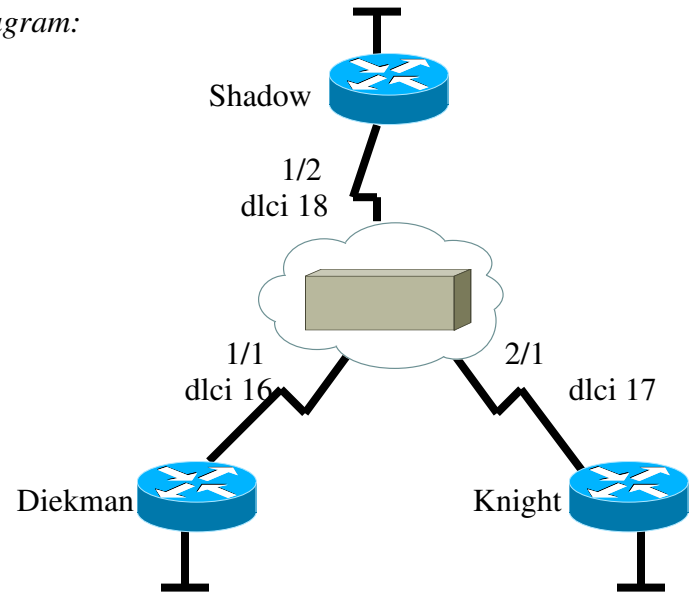
Fully-Meshed Frame Relay with 3 Routers and Sub-interfaces

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

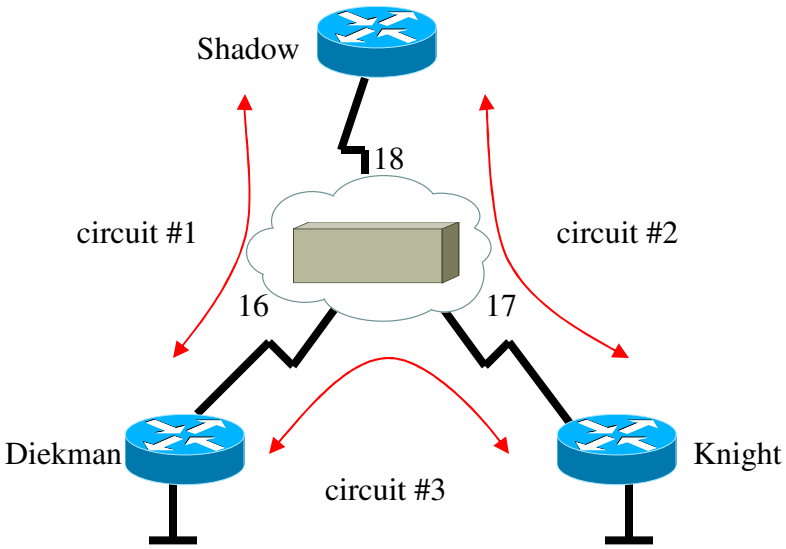
Lab Diagram:



Router	Diekman	Shadow	Knight
Serial int.	in text	in text	in text
Loop 0	1.1.1.1/8	2.2.2.2/8	3.3.3.3/8
Adtran	1/1	1/2	2/1
Dlci	16	18	17

Step-By-Step Instructions:

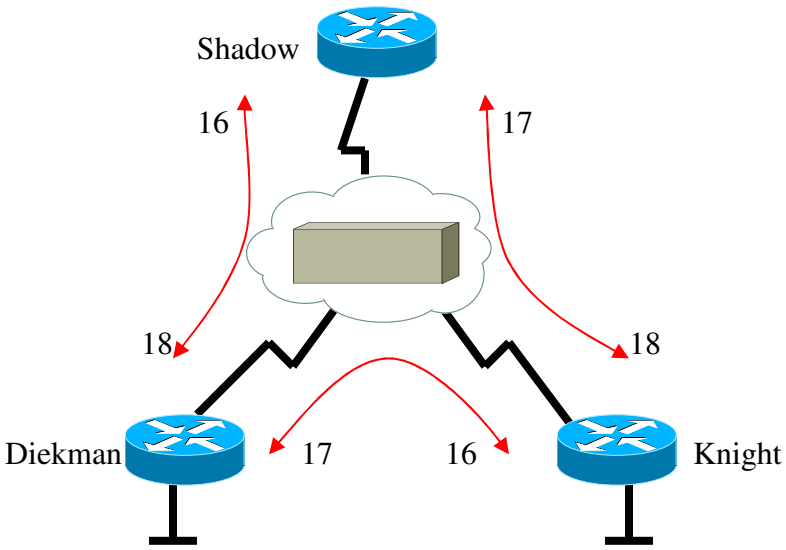
1. Cable the lab as shown and set up the basics on each router. Choose a routing protocol and set it up on each router (don't forget to advertise your networks). Add the loopback interface configurations.
2. To set up the routers for subinterfaces. The important thing here is to understand which ip addresses 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):

Router	Diekman	Shadow	Knight
S0/0	none	none	none
Adtran	1/1	1/2	2/1
Dlci	16	18	17
S0/0.16	none		
S0/0.17			none
S0/0.18		none	

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):

Router	Diekman	Shadow	Knight
S0/0	none	none	none
Adtran	1/1	1/2	2/1
Dlci	16	18	17
S0/0.16	none	192.168.1.2	
S0/0.17			none
S0/0.18	192.168.1.1	none	

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):

Router	Diekman	Shadow	Knight
S0/0	none	none	none
Adtran	1/1	1/2	2/1
Dlci	16	18	17
S0/0.16	none	192.168.1.2	
S0/0.17		192.168.2.1	none
S0/0.18	192.168.1.1	none	192.168.2.2

Finally, let's fill in the information for circuit #3 (192.168.3.0 network using .1 and .2):

Router	Diekman	Shadow	Knight
S0/0	none	none	none
Adtran	1/1	1/2	2/1
Dlci	16	18	17
S0/0.16	none	192.168.1.2	192.168.3.2
S0/0.17	192.168.3.1	192.168.2.1	none
S0/0.18	192.168.1.1	none	192.168.2.2

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-if)#ip address 192.168.3.2 255.255.255.0
knight(config-if)#no shut
knight(config-if)#frame-relay interface-dlci 16

knight(config-if)#int s0/0.18 point-to-point
knight(config-if)#ip address 192.168.2.2 255.255.255.0
knight(config-if)#no shut
knight(config-if)#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-if)#ip address 192.168.1.2 255.255.255.0
shadow(config-if)#no shut
shadow(config-if)#frame-relay interface-dlci 16

shadow(config-if)#int s0/0.17 point-to-point
shadow(config-if)#ip address 192.168.2.1 255.255.255.0
shadow(config-if)#no shut
shadow(config-if)#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-if)#ip address 192.168.3.1 255.255.255.0
diekman(config-if)#no shut
diekman(config-if)#frame-relay interface-dlci 17

diekman(config-if)#int s0/0.18 point-to-point
diekman(config-if)#ip address 192.168.1.1 255.255.255.0
diekman(config-if)#no shut
diekman(config-if)#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 changed 00:19:50
```

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 Univeristy, 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?

Frame Relay Operation and Troubleshooting

Objective:
To learn how to troubleshooting 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

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:

router#sh frame ?	
end-to-end	Frame-relay end-to-end VC information
fragment	show frame relay fragmentation information
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 information
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 information
events	Important Frame Relay packet events
foresight	Frame Relay router 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 exchanges with 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
router#

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.

```
router#sh frame lmi

LMI Statistics for interface Serial0/1 (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 118           Num Status msgs Rcvd 118
  Num Update Status Rcvd 0           Num Status Timeouts 0
router#
```

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
router#
```

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:

```
router#sh frame pvc

PVC Statistics for interface Serial0/0 (Frame Relay DTE)
router#
```

Ouch! Obviously trouble here. We can see we are suppose to be a DTE connection. Let's start at layer 1 and work our way up:

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

router#sh controller s0/0
Interface Serial0/0
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 193.168.1.2/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load
1/255
  Encapsulation FRAME-RELAY, loopback not set, keepalive set (10
sec)
                        LMI enq sent  52, LMI stat recvd 53, LMI upd
recvd 0, DTE LMI up
```

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:

```
router#sh frame map
Serial0/0 (up): ip 192.168.1.1 dlci 16(0x10,0x400), dynamic,
          broadcast,, status defined, active
router#
```

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 Serial0/0
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, load 1/255
Encapsulation FRAME-RELAY, loopback not set, keepalive set (10 sec)
LMI enq sent 45, LMI stat recvd 47, LMI upd recvd 0, DTE LMI up
LMI enq recvd 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 Rcvd 54
Num Update Status Rcvd 0 Num Status Timeouts 0

```
router#sh frame map
Serial0/0 (up): ip 192.168.1.1 dlci 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, 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    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?

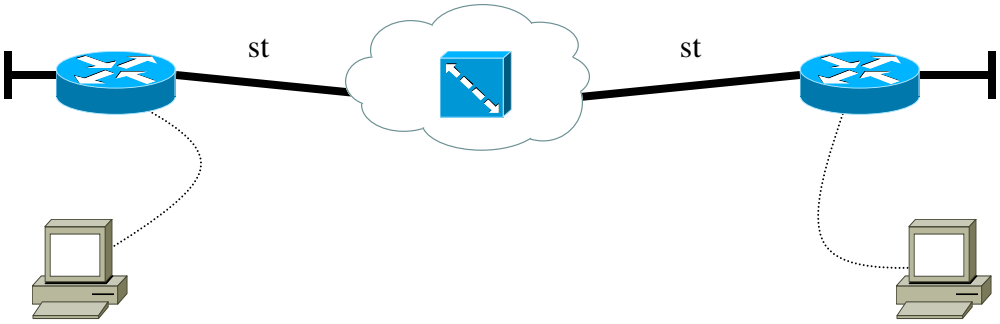
Nothing really glamorous just frame relay operation and troubleshooting. Again it is easy when you know how. Unlike the other troubleshooting labs this one relies more upon over-all troubleshooting commands (like show controllers and sh ip route) than upon specific frame-relay only troubleshooting commands. 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.

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:



Router	Smarts	Kissane
Loop 0	192.168.100.1/24	192.168.200.1/24
BRI0	192.168.1.1/24	192.168.1.2/24
Merge Port	A	B
Phone number	555-1000	555-2000

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).

```
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).

```
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:

```
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, calltype=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.

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

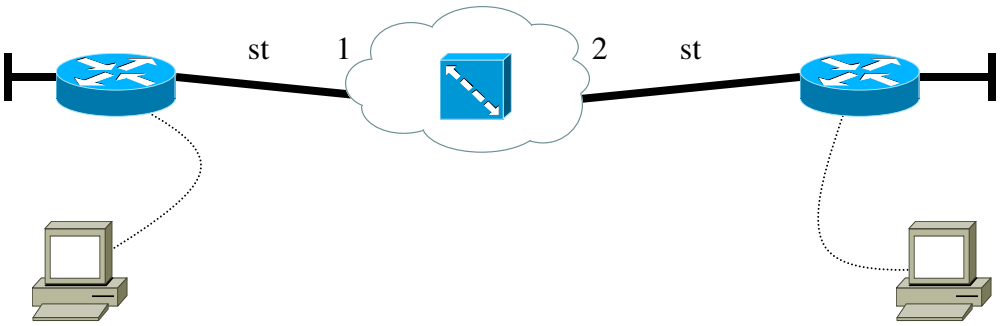
Timothy Kissane was a software developer for a company called System Management Arts Incorporated ("SMARTS"). When he was hired he signed a confidentiality agreement that he would never reveal any of the source code he developed for a program called "InCharge" (a network monitoring program). After he 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.

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:



Router	Smarts	Kissane
Loop 0	192.168.100.1/24	192.168.200.1/24
BRI0	192.168.1.1/24	192.168.1.2/24
Adtran Port	1	2
Phone number	555-1234	555-4000

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 BRI0/0. 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 BRI0/0 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.

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	PHYSICAL
2. ISDN call is made	DATA LINK
3. PPP handshaking	NETWORK

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 Prtocol). 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 — PHYSICAL
2. ISDN call is made — DATA LINK
3. PPP handshaking — NETWORK

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:

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:

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.

```
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
00:13:32: %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 Type	Calling Number	Called Number	Remote Name	Seconds Used	Seconds Left	Seconds Idle	Charges Units/Currency
Out		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'
01:33:236223242240: ISDN BR0/0: **RX** <- **CALL_PROC** pd = 8 callref = 0x87
01:33:236223201280: Channel ID i = 0x89
01:33:236223242240: ISDN BR0/0: **RX** <- **CONNECT** pd = 8 callref = 0x87
01:33:236223201280: 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**
01:33:55: ISDN BR0/0: TX -> **CONNECT_ACK** pd = 8 callref = 0x07
01:33:56: %LINEPROTO-5-UPDOWN: **Line protocol on Interface BRI0/0:1, changed state to up**
01:34:01: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 5551234 smarts

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
80108050402889018018 32C0735353531323334
01:52:90194354176: **ISDN BR0/0: RX <- RRr sapi = 0 tei = 64** nr = 23
01:52:92356225684: **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
01:52:90194354176: **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
01:52:90194313216: **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#
01:52:133144027136: **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 = (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 BRI0/0:1, changed state to up
8. Interface BRI0/0:1 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 BRI0/0:1, changed state to up
12. Interface BRI0/0:1 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 BRI0/0:1, changed state to up
02:04:20: %ISDN-6-CONNECT: Interface BRI0/0:1 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
02:04:20: BR0/0:1 **LCP: AuthProto CHAP** (0x0305C22305)
02:04:20: BR0/0:1 LCP: MagicNumber 0x0208BEB7 (0x05060208BEB7)
02:04:22: BR0/0:1 **LCP: I CONFREQ [REQsent]** id 7 len 15
02:04:22: BR0/0:1 **LCP: AuthProto CHAP** (0x0305C22305)
02:04:22: BR0/0:1 LCP: MagicNumber 0x045572BC (0x0506045572BC)
02:04:22: BR0/0:1 **LCP: O CONFACK [REQsent]** id 7 len 15
02:04:22: BR0/0:1 **LCP: AuthProto CHAP** (0x0305C22305)
02:04:22: BR0/0:1 LCP: MagicNumber 0x045572BC (0x0506045572BC)
02:04:22: BR0/0:1 **LCP: TIMEout: State ACKsent**
02:04:22: BR0/0:1 **LCP: O CONFREQ [ACKsent]** id 28 len 15
02:04:22: BR0/0:1 **LCP: AuthProto CHAP** (0x0305C22305)
02:04:22: BR0/0:1 LCP: MagicNumber 0x0208BEB7 (0x05060208BEB7)
02:04:22: BR0/0:1 **LCP: I CONFACK [ACKsent]** id 28 len 15
02:04:22: BR0/0:1 **LCP: AuthProto CHAP** (0x0305C22305)
02:04:22: BR0/0:1 LCP: MagicNumber 0x0208BEB7 (0x05060208BEB7)
02:04:22: BR0/0:1 **LCP: State is Open**
02:04:22: BR0/0:1 **PPP: Phase is AUTHENTICATING, by both**
02:04:22: BR0/0:1 **CHAP: O CHALLENGE** id 7 len 28 **from "kissane"**
02:04:22: BR0/0:1 **CHAP: I CHALLENGE** id 7 len 27 **from "smarts"**
02:04:22: BR0/0:1 **CHAP: I CHALLENGE** id 7 len 27 **from "smarts"**
02:04:22: BR0/0:1 **CHAP: I SUCCESS** id 7 len 4
02:04:22: BR0/0:1 **CHAP: I RESPONSE** id 7 len 27 **from "smarts"**
02:04:22: BR0/0:1 **CHAP: O SUCCESS** id 7 len 4
02:04:22: BR0/0:1 **PPP: Phase is UP**
02:04:22: BR0/0:1 **IPCP: O CONFREQ [Closed]** id 7 len 10
02:04:22: BR0/0:1 **IPCP: Address 192.168.1.2** (0x0306C0A80102)
02:04:22: BR0/0:1 **CDPCP: O CONFREQ** [Closed] id 7 len 4
02:04:22: BR0/0:1 **IPCP: I CONFREQ [REQsent]** id 7 len 10
02:04:22: BR0/0:1 **IPCP: Address 192.168.1.1** (0x0306C0A80101)
02:04:22: BR0/0:1 **IPCP: O CONFACK [REQsent]** id 7 len 10
02:04:22: BR0/0:1 **IPCP: Address 192.168.1.1** (0x0306C0A80101)
02:04:22: BR0/0:1 **CDPCP: I CONFREQ [REQsent]** id 7 len 4
02:04:22: BR0/0:1 **CDPCP: O CONFACK [REQsent]** id 7 len 4
02:04:22: BR0/0:1 **IPCP: I CONFACK [ACKsent]** id 7 len 10
02:04:22: BR0/0:1 **IPCP: Address 192.168.1.2** (0x0306C0A80102)
02:04:22: BR0/0:1 **IPCP: State is Open**

02:04:22: BR0/0:1 **CDPCP: I CONFACK [ACKsent]** id 7 len 4
02:04:22: BR0/0:1 **CDPCP: State is Open**
02:04:22: BR0/0 **IPCP: Install route to 192.168.1.1**
02:04:23: %LINEPROTO-5-UPDOWN: **Line protocol on Interface BRI0/0:1, changed state to up**
02:04:26: %ISDN-6-CONNECT: **Interface BRI0/0:1 is now connected to 5551234 smarts**

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 (from source)
3. LCP: AuthProto CHAP (from destination)
4. LCP: I CONFREQ [REQsent] (from source)
5. LCP: AuthProto CHAP (from destination)
6. LCP: O CONFACK [REQsent] (from source)
7. LCP: AuthProto CHAP (from destination)
8. LCP: TIMEout: State ACKsent (from source)
9. LCP: O CONFREQ [ACKsent] (from source)
10. LCP: AuthProto CHAP (from destination)
11. LCP: I CONFACK [ACKsent] (from source)
12. LCP: AuthProto CHAP (from destination)
13. LCP: State is Open (from source)
14. PPP: Phase is AUTHENTICATING, by both (from destination)
15. CHAP: O CHALLENGE from "kissane" (from source)
16. CHAP: I CHALLENGE from "smarts" (from destination)
17. CHAP: I CHALLENGE from "smarts" (from destination)
18. CHAP: I SUCCESS (from source)
19. CHAP: I RESPONSE from "smarts" (from destination)
20. CHAP: O SUCCESS (from source)
21. PPP: Phase is UP (from destination)
22. IPCP: O CONFREQ [Closed] (from source)
23. IPCP: Address 192.168.1.2 (from destination)
24. CDPCP: O CONFREQ (from source)
25. IPCP: I CONFREQ [REQsent] (from source)
26. IPCP: Address 192.168.1.1 (from destination)
27. IPCP: O CONFACK [REQsent] (from source)
28. IPCP: Address 192.168.1.1 (from destination)
29. CDPCP: I CONFREQ [REQsent] (from source)
30. CDPCP: O CONFACK [REQsent] (from source)
31. IPCP: I CONFACK [ACKsent] (from source)
32. IPCP: Address 192.168.1.2 (from destination)
33. IPCP: State is Open (from source)
34. CDPCP: I CONFACK [ACKsent] (from source)
35. CDPCP: State is Open (from destination)

36. IPCP: Install route to 192.168.1.1 (from source)
37. Line protocol on Interface BRI0/0:1, changed state to up
38. Interface BRI0/0:1 is now connected to 5551234 smarts

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):

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 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)**
03:02:24: BR0/0 DDR: ip (s=192.168.1.2, d=192.168.1.1), 100 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)**
kissane#
03:02:26: %ISDN-6-CONNECT**: Interface BRI0/0:1 is now connected to 5551234 smarts**

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
BRI0/0 is up, line protocol is up
  Internet address is 192.168.1.2/24
BRI0/0:1 is down, line protocol is down
BRI0/0:2 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 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
        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
Loopback0 is up, line protocol is up
  Internet address is 192.168.200.1/24

To fix this, go into your configuration under the BRI interfaces and type "no shut."

<u>No ISDN cable connected (layer 1)</u>
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
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

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: BRI0/0: **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 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 = 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
kissane#
```

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
kissane#
```

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.

```
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.

```
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 BR0/0: TX -> SETUP pd = 8 callref = 0x35
06:19:44: Bearer Capability i = 0x8890
06:19:44: Channel ID i = 0x83
06:19:44: Keypad Facility i = '5551234'
06:19:188978601984: ISDN BR0/0: RX <- CALL_PROC pd = 8 callref = 0xB5
06:19:188978561024: Channel ID i = 0x89
06:19:188978601984: ISDN BR0/0: RX <- CONNECT pd = 8 callref = 0xB5
06:19:188978561024: Channel ID i = 0x89
06:19:44: %LINK-3-UPDOWN: **Interface BRI0/0:1, changed state to up**
06:19:44: %ISDN-6-CONNECT**: Interface BRI0/0:1 is now connected to 5551234**
06:19:44: ISDN BR0/0: TX -> CONNECT_ACK pd = 8 callref = 0x35
06:19:45: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0/0:1, changed state to up
06:19:50: %ISDN-6-CONNECT: **Interface BRI0/0:1 is now connected to 5551234 smarts**
kissane# 06:19:249108103851: %ISDN-6-DISCONNECT: **Interface BRI0/0:1 disconnected from 5551234 smarts**, call lasted 13 seconds
06:19:58: ISDN BR0/0: TX -> DISCONNECT pd = 8 callref = 0x35
06:19:58: Cause i = 0x8090 - Normal call clearing
06:19:251270015772: ISDN BR0/0: RX <- RELEASE pd = 8 callref = 0xB5
06:19:58: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to down
06:19:58: ISDN BR0/0: TX -> RELEASE_COMP pd = 8 callref = 0x35
06:19:59: %LINEPROTO-5-UPDOWN: **Line protocol on Interface BRI0/0:1, changed state to down**
kissane#

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#
06:23:81604378624: ISDN BR0/0: RX <- RRp sapi = 0 tei = 64 nr = 13
06:23:19: ISDN BR0/0: TX -> RRf sapi = 0 tei = 64 nr = 10
06:23:19: ISDN BR0/0: TX -> INFOc sapi = 0 tei = 64 ns = 13 nr = 10 i = 0x0
80136050402889018018 32C0735353531323334
06:23:81604419584: ISDN BR0/0: RX <- RRr sapi = 0 tei = 64 nr = 14

06:23:83766291092: ISDN BR0/0: RX <- INFOc sapi = 0 tei = 64 ns = 10 nr = 14 i = 0x0801B602180189
06:23:19: ISDN BR0/0: TX -> RRr sapi = 0 tei = 64 nr = 11
06:23:81604419584: ISDN BR0/0: RX <- INFOc sapi = 0 tei = 64 ns = 11 nr = 14 i = 0x0801B607180189
06:23:19: ISDN BR0/0: TX -> RRr sapi = 0 tei = 64 nr = 12
06:23:19: %LINK-3-UPDOWN: **Interface BRI0/0:1, changed state to up**
06:23:19: %ISDN-6-CONNECT: **Interface BRI0/0:1 is now connected to 5551234**
06:23:19: ISDN BR0/0: TX -> INFOc sapi = 0 tei = 64 ns = 14 nr = 12 i = 0x0801360F
06:23:85899345920: ISDN BR0/0: RX <- RRr sapi = 0 tei = 64 nr = 15
06:23:20: %LINEPROTO-5-UPDOWN: **Line protocol on Interface BRI0/0:1, changed state to up**
06:23:107374223360: ISDN BR0/0: RX <- RRp sapi = 0 tei = 65 nr = 0
06:23:25: ISDN BR0/0: TX -> RRf sapi = 0 tei = 65 nr = 0
06:23:25: %ISDN-6-CONNECT: **Interface BRI0/0:1 is now connected to 5551234 smarts**
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 BRI0/0:1 disconnected from 5551234 smarts, call lasted 19 seconds**
06:23:39: ISDN BR0/0: TX -> INFOc sapi = 0 tei = 64 ns = 15 nr = 12 i = 0x0801364508028090
06:23:169652894924: ISDN BR0/0: RX <- RRr sapi = 0 tei = 64 nr = 16
06:23:169665637012: ISDN BR0/0: RX <- INFOc sapi = 0 tei = 64 ns = 12 nr = 16 i = 0x0801B64D
06:23:39: ISDN BR0/0: TX -> RRr sapi = 0 tei = 64 nr = 13
06:23:39: %LINK-3-UPDOWN: **Interface BRI0/0:1, changed state to down**
06:23:39: ISDN BR0/0: TX -> INFOc sapi = 0 tei = 64 ns = 16 nr = 13 i = 0x0801365A
06:23:167503724544: ISDN BR0/0: RX <- RRr sapi = 0 tei = 64 nr = 17
06:23:40: %LINEPROTO-5-UPDOWN: **Line protocol on Interface BRI0/0:1, 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 BRI0/0:1, **changed state to up**
06:26:48: %ISDN-6-CONNECT: Interface **BRI0/0:1 is now connected to 5551234**
06:26:49: %LINEPROTO-5-UPDOWN: **Line protocol on Interface BRI0/0:1, changed state to up**
06:26:54: %ISDN-6-CONNECT: **Interface BRI0/0:1 is now connected to 5551234 smarts**
06:26:253403070464: **%ISDN-6-DISCONNECT: Interface BRI0/0:1 disconnected from 5551234 smarts, call lasted 10 seconds**
06:26:59: %LINK-3-UPDOWN: **Interface BRI0/0:1, changed state to down**
06:27:00: %LINEPROTO-5-UPDOWN: **Line protocol on Interface BRI0/0:1, changed state to down**
kissane#

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 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 HDLC, loopback not set

Got it! We need to change our encapsulation to PPP to get this thing to work.

<u>Incorrect username/password (layer 2/3):</u>
For this I changed the username but left the password "as-is."

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 = 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
kissane#

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 CONFACK [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 CONFACK [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
06:41:13: BR0/0:1 PPP: **Phase is AUTHENTICATING, by the peer**
06:41:13: BR0/0:1 **CHAP: I CHALLENGE id 32 len 27 from "smarts**"
06:41:13: BR0/0:1 **CHAP: Username smarts not found**
06:41:13: BR0/0:1 **CHAP: Unable to authenticate for peer**
06:41:13: BR0/0:1 **PPP: Phase is TERMINATING**
06:41:13: BR0/0:1 LCP: O TERMREQ [Open] id 31 len 4
06:41:13: BR0/0:1 LCP: I TERMACK [TERMsent] id 31 len 4
06:41:13: BR0/0:1 LCP: **State is Closed**
06:41:13: BR0/0:1 PPP: **Phase is DOWN**
06:41:13: BR0/0:1 PPP: Phase is ESTABLISHING, Passive Open
06:41:13: BR0/0:1 LCP: State is Listen
06:41:13: %LINK-3-UPDOWN: **Interface BRI0/0:1, changed state to down**
06:41:13: BR0/0:1 LCP: State is Closed
kissane#
06:41:13: BR0/0:1 PPP: Phase is DOWN

A problem with our username (CHAP: Username smarts not found). We just go back and fix it.

Missing username/password (layer 2/3):

```
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:
        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

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#
06:46:52: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0/0, TEI 64 changed
to up
06:46:52: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0/0, TEI 65 changed
to up
06:46:52: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
06:46:52: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to
5551234

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)
```

06:47:25: BR0/0:1 PPP: Treating connection as a callout
06:47:25: BR0/0:1 PPP: Phase is ESTABLISHING, Active Open
06:47:25: BR0/0:1 LCP: O CONFREQ [Closed] id 51 len 10
06:47:25: BR0/0:1 LCP: MagicNumber 0x030BEA68 (0x0506030BEA68)
06:47:25: BR0/0:1 LCP: I CONFREQ [REQsent] id 77 len 15
06:47:25: BR0/0:1 LCP: AuthProto CHAP (0x0305C22305)
06:47:25: BR0/0:1 LCP: MagicNumber 0x055899FC (0x0506055899FC)
06:47:25: BR0/0:1 LCP: O CONFACK [REQsent] id 77 len 15
06:47:25: BR0/0:1 LCP: AuthProto CHAP (0x0305C22305)
06:47:25: BR0/0:1 LCP: MagicNumber 0x055899FC (0x0506055899FC)
06:47:25: BR0/0:1 LCP: I CONFACK [ACKsent] id 51 len 10
06:47:25: BR0/0:1 LCP: MagicNumber 0x030BEA68 (0x0506030BEA68)
06:47:25: BR0/0:1 LCP: State is Open
06:47:25: BR0/0:1 PPP: Phase is AUTHENTICATING, by the peer
06:47:25: BR0/0:1 **CHAP: I CHALLENGE id 41 len 27 from "smarts"**
06:47:25: BR0/0:1 **CHAP: Username smarts not found**
06:47:29: BR0/0:1 **CHAP: Unable to authenticate for peer**
06:47:29: BR0/0:1 PPP: Phase is TERMINATING
06:47:29: BR0/0:1 LCP: O TERMREQ [Open] id 56 len 4
06:47:29: BR0/0:1 LCP: I TERMACK [TERMsent] id 56 len 4
06:47:29: BR0/0:1 LCP: State is Closed
06:47:29: BR0/0:1 PPP: Phase is DOWN

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).

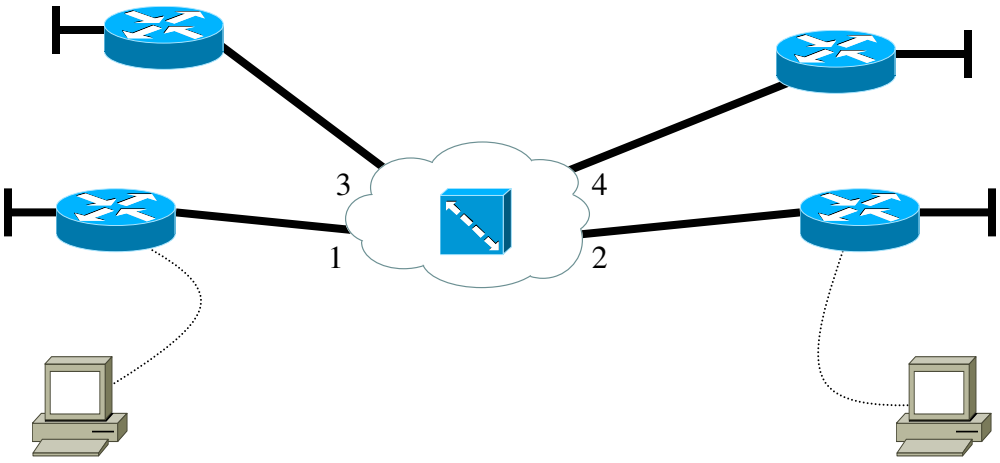
So what have I learned here?
So is your mind fried yet? In this lab we learned how the theory of how ISDN operates and how to troubleshoot it. Like our other troubleshooting labs we just follow the OSI model. Please keep in mind that ISDN problems may not always be ISDN-related. In one of the next labs you will be setting up ISDN between four routers. At first when I could not get it to work I ran through my IDSN 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:
(2) routers
(2) straight-through cables
(1) workstation
(1) ADTRAN Atlas 550
(2) PC/workstations
(2) console cables

Lab Design:



Router	Geoffrey	Osowski
Loop 0	192.168.110.1/24	192.168.120.1/24
BRI0	192.168.1.1/24	192.168.1.2/24
Adtran Port	1	2
Phone number	555-1234	555-4000
SPID1	51055512340001	51055540000001
SPID2	51055512350001	51055540010001

Router	Wilson	Tang
Loop 0	192.168.130.1/24	192.168.140.1/24
BRI0	192.168.1.3/24	192.168.1.4/24
Adtran Port	3	4
Phone number	555-7000	555-8000
SPID1	51055570000001	51055580000001
SPID2	51055570010001	51055580010001

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 "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).

```
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 192.168.1.1 255.255.255.0
osowski(config-if)#no shut
osowski(config-if)#dialer-group 1
osowski(config-if)#dialer map ip 192.168.1.2 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 192.168.1.2
```

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.”

```
osowski(config)#ip route 192.168.110.0 255.255.255.0 192.168.1.1
```

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
wilson(config)#dialer-list 1 protocol ip permit

tang(config)# isdn switch-type basic-ni
tang(config)#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
wilson(config-if)#ip address 192.168.1.3 255.255.255.0
wilson(config-if)#no shut
wilson(config-if)#dialer-group 1
wilson(config-if)#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
wilson(config-if)#ppp authentication chap
wilson(config)#username tang password cisco
wilson(config)#ip host tang 192.168.1.4

tang(config)#int bri0/0
tang(config-if)#ip address 192.168.1.4 255.255.255.0
tang(config-if)#no shut
tang(config-if)#dialer-group 1
tang(config-if)#dialer map ip 192.168.1.3 name wilson 5557000
tang(config-if)#isdn spid1 51055580000001 5558000
tang(config-if)#isdn spid2 51055580010001 5558001
tang(config-if)#enc ppp
tang(config-if)#ppp authentication chap
tang(config)#username wilson password cisco
tang(config)#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.

    geoffrey(config-if)#dialer map ip 192.168.1.3 name wilson 5557000
    geoffrey(config-if)#dialer map ip 192.168.1.4 name tang 5558000
    geoffrey(config-if)#ppp multilink
    geoffrey(config)#username wilson password cisco
    geoffrey(config)#username tang password cisco
    geoffrey(config)#ip host wilson 192.168.1.3
    geoffrey(config)#ip host tang 192.168.1.4
    geoffrey(config)#ip route 192.168.300.0 255.255.255.0 192.168.1.3
    geoffrey(config)#ip route 192.168.400.0 255.255.255.0 192.168.1.4

    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 192.168.300.0 255.255.255.0 192.168.1.3
    osowski(config)#ip route 192.168.400.0 255.255.255.0 192.168.1.4

    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 192.168.200.0 255.255.255.0 192.168.1.2
wilson(config)# ip route 192.168.100.0 255.255.255.0 192.168.1.1

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 192.168.200.0 255.255.255.0 192.168.1.2
tang(config)# ip route 192.168.100.0 255.255.255.0 192.168.1.1
```

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

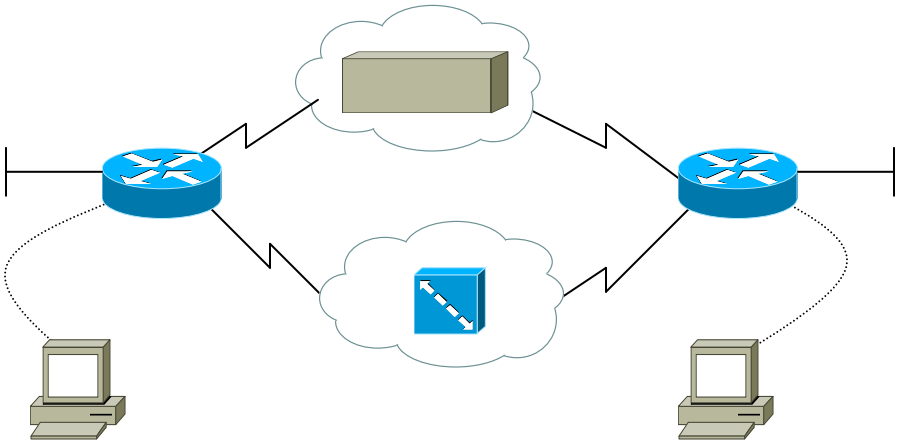
Geoffrey Osowski and Wilson Tang were accountants who worked for CISCO (yes…them!). They were charged with computer-related crimes when, being accountants, they illegally issued almost $8,000,000 in CISCO stock to themselves. They each received 34 months in prison, 36 months of probation, and restitution of almost $8,000,000. The moral of the story is don't try this at home!

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:



Router	Patrice	Williams
S0/0	192.168.1.1/24	192.168.1.2/24
BRI0/0	10.0.0.1/8	10.0.0.2/8
Loop0	1.1.1.1/8	2.2.2.2/8

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.

```
patrice(config)#int bri0/0
patrice(config-if)#dialer idle-timeout 20
patrice(config-if)#dialer wait-for-carrier-time 10
patrice(config)#ip route 2.0.0.0 255.0.0.0 10.0.0.2 150

williams(config)#int bri0/0
williams(config-if)#dialer idle-timeout 20
williams(config)#ip route 1.0.0.0 255.0.0.0 10.0.0.1 150
```

11. 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.

Part 5 Command Review

Objective:
To list all commands utilized in Part 5 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.

Prompt	Command	Shortcut	Description

Prompt	Command	Shortcut	Description

Whole Enchilada/Crazy Insano Lab #1 (WECIL)

This whole book was designed to get you to use critical thinking skills and apply the theories with hands-on applications. In this last WECIL I want you to design a network that will encompass some or all of the material in this book. Just some topics to jumble around:

Different address classes (public/private)
Different routing protocols: RIP, RIPv2, IGRP (maybe a bit of BGP or EIGRP)
Using IP or IPX or both
Using static or dynamic routing
Using VLAN's
Using STP
Using DUN
Using Frame Relay with ISDN backup
Using PPP
Using ACL's

Coming In the Next Version of the Textbook in Summer 2003 (sorry….I had lofty ambitions and the deadline for publication came up quickly):

Part 1:
Writing a resume for computer-related occupations
Writing a cover letter for computer-related occupations
Using CISCO Design Software Tutorial
Using CISCO Design Software: John's Brewhouse
UNIX operating system basics
Paper Lab: Token Ring Packet Structure
Small Networking Lab: Token Ring Networks
Small Networking Lab: More on Microsoft Windows
Small Networking Lab: Windows 2000
Small Networking Lab: Novell Networks
Small Networking Lab: Unix Networks
Small Networking Lab: Macintosh Networks
DNS servers
SNMP Lab

Part 2:
Larger RIP networks
More subnetting examples
Subnetting examples with CISCO Works 2000
Passwords and Recovery
Loading an IOS
Using a web browser with your router
Boot Sequence and Confreg (hacking lab)
IOS 10 vs. 11 vs. 12 commands

Part 3
More subnetting examples
Subnetting examples with CISCO Design Software
Using the command line interface with 1900/2900/4000/5000 switches
Bridges

Part 4
More redistribution labs
More subnetting examples
Subnetting examples with CISCO Design Software
Dynamic ACL basics
CBAC basics
AAA basics
More ACL labs

Part 5
More on DDR

More redistribution labs
More subnetting examples
Programming ADTRAN's
Programming MERGE boxes
Setting up DSU/CSU's
Using T-1's
ISDN PRI's

Plus I am going to fix any edits or problems within the labs. I have taken every lab and used them in 2-4 classes and had all of them proof-read several times but things still can slip by. Sorry for any inconvenience. If you send me an email I promise I will change any errors in the next edition. BashamM@spjc.edu

Did you like this book?

There's more!

Coming Fall 2002

More books in the
"Learning by Doing Series"

Computer Security Fundamentals

General Networking Concepts

Each book will continue the tradition of "Learning by Doing" by providing real-world, hands-on labs to understand various concepts. These two books will contain both theory and extensive labs that you can use right in your own school or home. Ask your teacher or textbook publisher about your copy today!