

Lab 3 – Routing Information Protocol (RIPv1) on a Cisco Router Network

CMPE 150 Fall 2005

Introduction

Today you are going to be thrown into using Cisco's Internetwork Operating System (IOS) to configure three routers in a simple network topology. Unlike switches, routers are not very plug and play and require network planning to set up. Most of the planning has been done for you in this lab, and it's up to you to do the rest. To route packets among the networks, RIPv1 will be used. A short description of RIP follows.

RIP background

Due to time constraints you have today I will spare you long details on RIP, but I will include a small excerpt from the RIPv1 RFC 1058.

"RIP is one of a class of algorithms known as "distance vector algorithms". The earliest description of this class of algorithms known to the author is in Ford and Fulkerson [6]. Because of this, they are sometimes known as Ford-Fulkerson algorithms. The term Bellman-Ford is also used. It comes from the fact that the formulation is based on Bellman's equation, the basis of "dynamic programming"."

Getting Started with the Router and IOS (Internetwork OS)

Step 1 – Connecting to the Console port

Patch your computer's serial port to the router you are working on with a straight-through Cat5 cable (all the cables in the lab except the orange ones are straight-through). Refer to Appendix A-1 to figure out which computer will patch to which router, Appendix A-4 to figure out which port on the patch panels is connected to your computer's serial port and Appendix A-3 to figure out which port on the router connects to the serial port of your PC.

Step 2 – Running Minicom – a standard terminal emulator – to communicate with and configure the router

To start Minicom on your Linux machine, type `minicom` in a terminal window. The router's console port is connected to COM 1 of your computer and uses 9600 baud, 8 data bits, no parity, 1 stop bit and no flow control. If you don't manage to communicate with the router, make sure your Minicom settings are correct. To access the menu in Minicom, press Ctrl-A and then Z.

Step 3 - Bypassing Setup Mode

At this point, your host should have a console session with the Router. Since these routers may, or may not have a pre-existing configuration, what you see after pressing the Enter key may vary.

Possibility #1:

After several lines of information on the screen you should eventually see:

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

If you see such a message, the router is ready to enter Setup Mode. For now, if you are being prompted to answer a question, answer "n" and press return. If you accidentally press "y" and

enter Setup Mode, press and hold down the control key and press C (CTRL-C). Wait a few seconds, and then press Enter.

On some routers you may see the following message.

```
Would you like to terminate autoinstall? [yes/no]: y
Type "y" or yes to terminate autoinstall.
```

You should eventually see the router prompt:

```
Router>
Go to Step 4
```

Possibility #2

You may see a router prompt, similar to this example:

```
Router>
```

If you see a prompt ending in the ">" someone has saved a configuration on the router (startup-config). Because it is always best to begin our labs without a previous router configuration, you will need to erase the startup-config and reboot the router:

```
Router> enable
Router# erase startup-config
Turn off the router and then turn it back on or type reload.
Go to Possibility #1.
```

Step 4 - Basic Commands

You should now be at the User Mode with the prompt:

```
Router>
```

User mode will allow you to view the state of the router, but will not allow you to modify its configuration.

Using help

Use the "?" to view the list of available commands or command options

```
Router> ?
a) Press the Space Bar to scroll a "screen's worth" of more commands.
b) Press the Enter or Return key to scroll down just one line of the list.
c) Press any other key to halt the list output.
Router> show ?
Displays the next parameter or parameters that can be used with this command.
Router> show interface ?
Displays the next parameter or parameters that can be used with this command.
```

Try some others!

Privilege Mode

Now lets enter Privilege Mode (or enable mode) which does allow you to modify the router's configuration.

```
Router> enable
```

Note: On a production router (and in our labs) there would be a privilege password set where you would have to give the correct password before being able to enter privilege mode. Notice how the prompt changes!

Router# disable This will take you back to User Mode.
Router> enable This will take us back to Privilege Mode
Router> exit Exits the user out of User Mode (more later)
Router# ? Notice how many more commands are available to you in Privilege Mode.

Any commands entered from User Mode can also be entered from Privilege Mode. However, not all Privilege Mode commands can be entered in User Mode.

Using Help and Setting the Clock

Router# clock set ?
Use the clock set command along with the “?” help, to give you the commands the router IOS is expecting. Notice that it leaves the part of the command you already have included. Continue with this command until you have successfully changed the time and date.
Router# show clock
Verify the time and date.

Abbreviated commands

The Cisco IOS will allow you to abbreviate any command or parameter as long as it uniquely identifies the command or parameter.

Router# sh inter Instead of “**show interface**”.

Viewing the configuration

To view the current configuration of the router (which resides in RAM) known as the running-config:

Router# show running-config
You will notice that this contains the actual commands that are either default commands or were used to configure the router. When you make changes to the router, those changes are stored in the running-config file.

Router# show startup-config
This will show the saved configuration in NVRAM. If the running-configuration has not been saved, the startup-config file will be empty.

Router# copy running-config startup-config
This will copy the current running-config file (in RAM) to the startup-config file in NVRAM. Now if the router is rebooted (or loses power and powered back up) the changes you made to the router have been saved. The startup-config is copied to the running-config during the reboot process.

Router# copy run start
This is an example of abbreviating a command. BE CAREFUL! Do NOT abbreviate the file names incorrectly, i.e. copy running start-up as that will have a different affect on the router, in this case overwrite the IOS in flash (later).

Router# show startup-config
Router# show running-config
Notice that the running-config and the startup-config are identical.

Erasing the saved startup-config

During labs it is always a good idea to save your running-config to the startup-config just in case something happens like a power outage. When you are done with every lab, before leaving the routers, you must erase the startup-config file so that the next student is working with an unconfigured router.

```
Router# erase startup-config
```

Normally, at this point you would turn-off the router, but for this lab, let us continue.

Rebooting the router

```
Router# reload
```

If you wanted to reset the router, turn it off and back on, you can do that with either the power switch or this reload command. Try it!

Setup Mode

Just like in Step 3, after several lines of information on the screen you should eventually see:

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

The reason the router went into Setup Mode is because there was no startup-config as it was erased. If there was a startup-config it would have been copied into the running-config and you would not have seen the above question. Instead you would have been taken to the router prompt:

```
Router>
```

At any time, you can enter Setup Mode, but most administrators do not configure the router this way as it is limiting in what it can do. You can enter Setup Mode by giving the following command:

```
Router# setup
```

Notice that you must be in privilege mode. Now press control-C to exit setup mode.

Step 5 - Viewing the output

Lets take a look at some of the show commands and discuss the output with your neighbor (or yourself).

```
Router# show interface
```

What does this command show you? Is there any information you recognize? How many interfaces does your router have and what are their names? How would you show a specific interfaces? (show interface serial 0)

```
Router# show ip interface brief
```

What kind of information does this command give you? What is the state of the interfaces? Do they have IP addresses configured yet? No is correct, but why not? That's right you have not configured them yet.

Match the interfaces displayed in the previous commands with the physical interfaces on the router (you can use the photo in Appendix A-3).

```
Router# show running-config
```

Does this command show you the name and types of interfaces also? (Yes!)

Basic Commands Summary

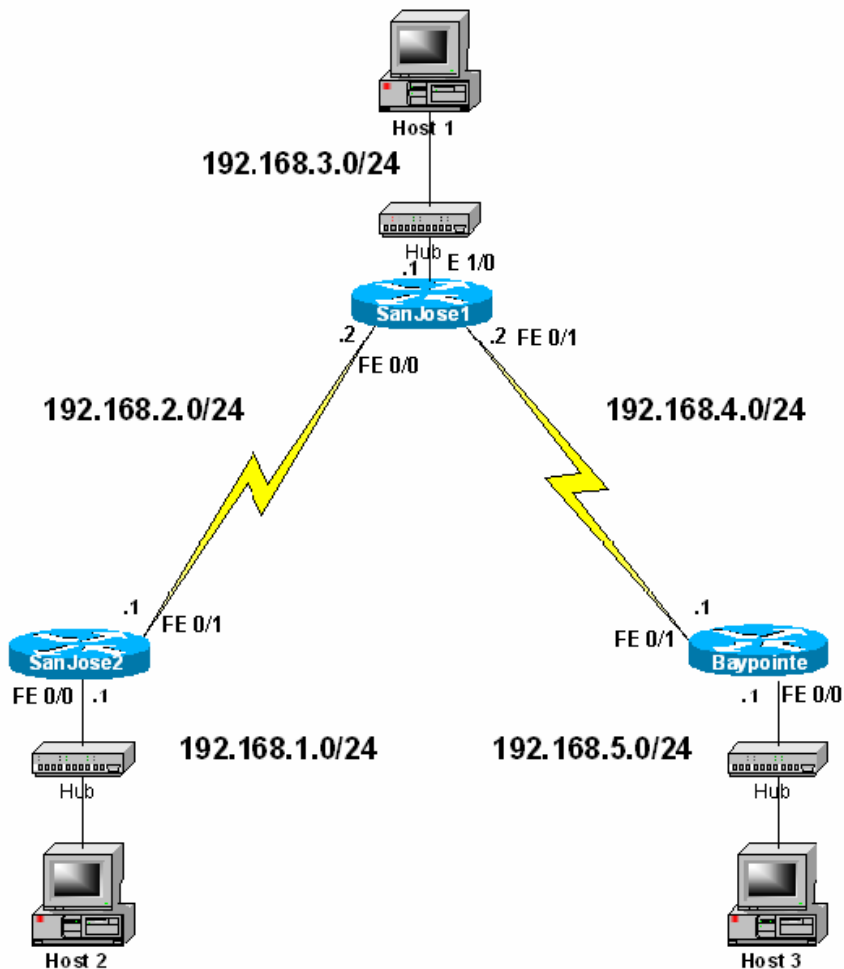
The best way to learn, understand and remember the commands will be using is by repetition (if you have time, there's lots more to do).

Here are some of the commands you learned so far in this lab:

```
Router> enable
Router# disable
Router> exit
Router> ?
Router# ?
Router# show interface
Router# clock set
Router# show running-config
Router# show startup-config
Router# erase startup-config
Router# copy running-config startup-config
Router# show ip interface brief
Router# reload
```

Setting up a Classful Network Scenario

Now on to a more interesting and challenging use of the routers. You will be setting up the network topology in the following figure. In doing this you will assign IP addresses to those interfaces of your router that you will be using and administratively turning on those interfaces.



Once you've set up the topology, you will configure the routers to run RIPv1 to establish routes between all the classful networks (see http://en.wikipedia.org/wiki/Classful_network for definition of classful network).

Setup

- Connect the three routers together with 2 crossover cables (short orange ones).
- Wait until you have RIP running to connect the computers to the routers. When the time comes to do that, you will patch them straight to the routers using crossover cables – no hubs or switches required. More detail on that will follow the router setup.
- Configure the routers to include hostnames and the proper interface commands including IP addresses, subnet masks, etc. Each router should be able to ping the interface of the adjacent (neighboring) router and the host on its LAN (Ethernet) interface. Test and troubleshoot as necessary.

Basic Configurations

Each router's console connection is connected to a separate computer. As there are three of you in each group, you can do this and the following step in parallel. The figure in Appendix A-1 indicates which router should be connected to which computer. If you connected them in any other way, make sure you switch the connections around so that they match those in the figure, or you may get very confused.

To set these configuration parameters, first, go to Privileged mode on the router (you will see the “#” on the prompt). Then enter the following command:

```
Router# configure terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

Next, each member of the group should enter the configuration parameters of their respective router.

Before starting, you will find it useful to put your console in synchronous mode which means that when the router prints out some information onto the screen, it doesn't interrupt the command you're typing.

```
Router(config)# line console 0
```

```
Router(config-line)# logging synchronous
```

SanJose2 – top router

```
hostname SanJose2
interface FastEthernet0/0
  ip add 192.168.1.1 255.255.255.0
  no shutdown
interface FastEthernet0/1
  ip add 192.168.2.1 255.255.255.0
  no shutdown
```

SanJose1 – middle router

```
hostname SanJose1
interface Ethernet1/0
  ip add 192.168.3.1 255.255.255.0
  no shutdown
interface FastEthernet0/0
  ip add 192.168.2.2 255.255.255.0
  no shutdown
interface FastEthernet0/1
  ip add 192.168.4.2 255.255.255.0
```

```
no shutdown
```

Baypointe – bottom router

```
hostname Baypointe
interface FastEthernet0/0
  ip add 192.168.5.1 255.255.255.0
  no shutdown
interface FastEthernet0/1
  ip add 192.168.4.1 255.255.255.0
  no shutdown
```

Starting debug ip rip

BEFORE we configure RIP, we want to turn on the debugging of RIP messages. This is not a normally something you would do on a production network unless you are troubleshooting the network. However, the whole purpose of this lab is to view the RIP updates messages sent and received by each router, so we will use the debug command to accomplish this task.

On each router, start the process debug ip rip, which will allow us to view RIP update messages being sent and received. You must be in privileged mode to run debug. Basically, each group member should type this command into their router prompt. Notice that by this time you should have exited configure terminal mode with CTRL-Z.

```
SanJose1# debug ip rip
RIP protocol debugging is on
```

```
SanJose2# debug ip rip
RIP protocol debugging is on
```

```
Baypointe# debug ip rip
RIP protocol debugging is on
```

Starting RIP

Now we can start RIP on each router.

From global configuration (after having typed configure terminal) you will enter the command:

```
Router(config)# router rip
```

Once you are in the Router RIP configuration sub-mode, all you need to do is enter the classful network address for each directly connected network, using the network command.

```
Router(config-router)#network directly-connected-classful-
network-address
```

For the best viewing results, I would suggest doing these commands in this order (so don't all do this step in parallel, wait each other out).

Here are the commands for each router:

San Jose 2 – top router

```
SanJose2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SanJose2(config)# router rip
SanJose2(config-router)# network 192.168.1.0
SanJose2(config-router)# network 192.168.2.0
```

Baypointe – bottom router

```
Baypointe# configure terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.
Baypointe(config)# router rip
Baypointe(config-router)# network 192.168.4.0
Baypointe(config-router)# network 192.168.5.0
```

SanJose1 – middle router

```
SanJose1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SanJose1(config)# router rip
SanJose1(config-router)# network 192.168.2.0
SanJose1(config-router)# network 192.168.3.0
SanJose1(config-router)# network 192.168.4.0
```

You will immediately begin to see time stamped debug RIP messages. Does this remind you of any of the homework problems you did on Chapter 5 in Tanenbaum?

Understanding the debug ip rip output

Because we have RIP protocol debugging on (debug ip rip) we will begin to see network discovery, as each router:

SENDING RIP MESSAGES

Each router will begin to send RIP update message out each interface belonging to one of the network statements.

```
SanJose2(config)#router rip
SanJose2(config-router)#network 192.168.1.0
SanJose2(config-router)#network 192.168.2.0
```

- For example, SanJose2 will send out RIP update messages on FastEthernet0/0 because that interface has an IP address that belongs to the network 192.168.1.0, and on FastEthernet0/1 because that interface has an IP address that belongs to the network 192.168.2.0.
- Just because a router has a directly connected network does not mean it will automatically include that network in its routing updates to neighboring routers. The network command also tells the RIP to include these networks in its updates to adjacent neighbors.

LISTENING FOR RIP MESSAGES

Routers will also listen for RIP messages on each interface belonging to one of the network statements.

- For example, SanJose2 will listen for RIP update messages on Ethernet 0 because that interface has an IP address that belongs to the network 192.168.1.0, and also listen for RIP update messages on Serial 0 because that interface has an IP address that belongs to the network 192.168.2.0.
- As RIP messages are received, the router will add those networks in the messages to their routing tables:
 - If the RIP message contains a network not currently in the routing table.
 - If the RIP message contains a network with a better metric (fewer hops) than an entry currently in the routing table.

Before going any further, look at the routing table at each router by using the command:

```
Router# show ip route
```

Connecting the hosts

Up until now, your network consisted of nothing more than three routers. Now it's time to make it a little more realistic and add the end-hosts. Using crossover cables, patch your computers'

Ethernet ports to the correct interface of the router (you can figure this out from the topology figure a few pages earlier. The patch panel ports are clearly outlined in Appendix A-4.

What do you notice about the RIP update messages now?

Look at the routing tables at the router now. Any changes from before?

Now that your computers are connected to the topology let's use `ping` and `tracert` to investigate the network. You can't do that just yet though. To see why, open a new terminal window (a terminal to your local computer, not the router) and type

```
> ifconfig
```

What do you notice about your ip address? Does it belong on any of the networks in the topology you've just set up?

To give your computer an address that is on the right network you will use `ifconfig`, but you have to have root access in order to do this. To gain root access, type the command:

```
> sudo bash
```

This basically opens a new shell with root permissions. Now you can assign your computer an IP address. You need to set one more parameter on top of the ip address – you need to tell your computer the ip address of the interface on the router through which you connect to the network topology. This is your “gateway” to all the other networks. The following table lists suggested IP addresses to use for the host computers, and the corresponding default gateways.

Router host is connected to	Suggested IP address	Gateway IP address
SanJose2	192.168.1.10	192.168.1.1
SanJose1	192.168.3.10	192.168.3.1
Baypointe	192.168.5.10	192.168.5.1

The command to set the IP address is:

```
> ifconfig eth0 address
```

The command to set the gateway is

```
> route add default gw gateway_address
```

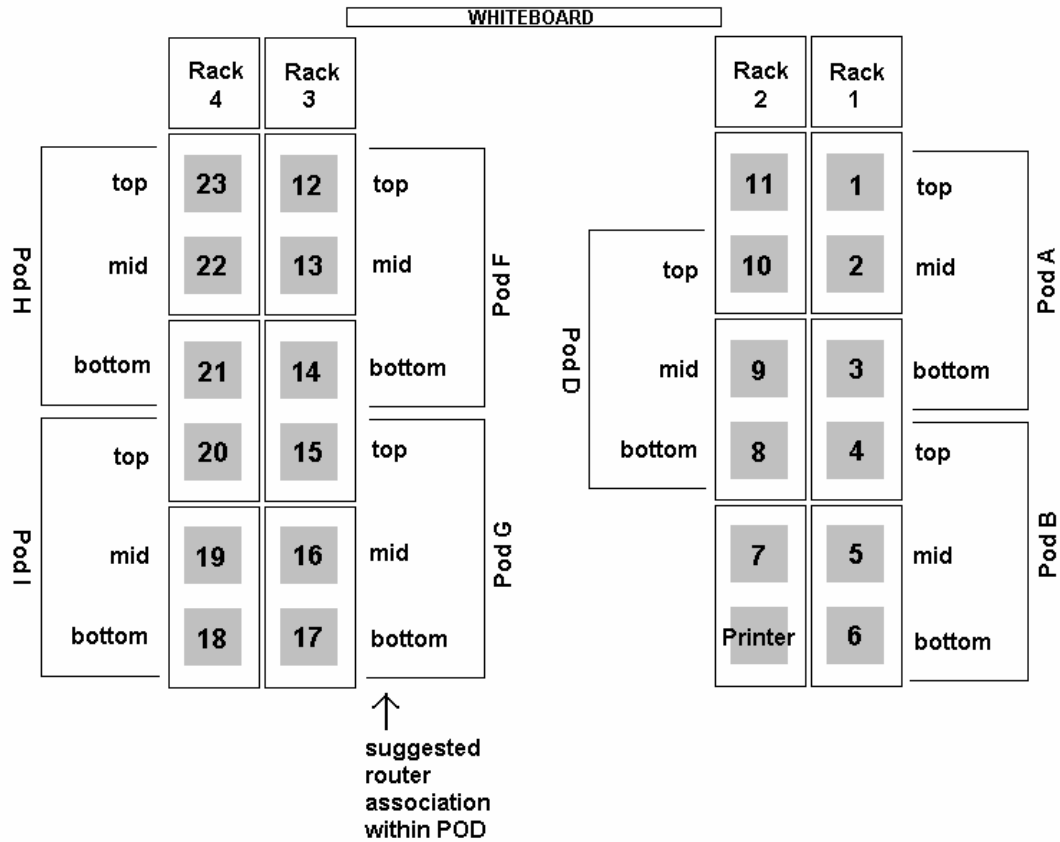
Each of you should do this for your particular station. When you're done setting your addresses, try pinging each other's computers. Are they reachable? Now try `tracert`. What IP addresses does `tracert` list on the route it finds?

Congratulations, you're finished.

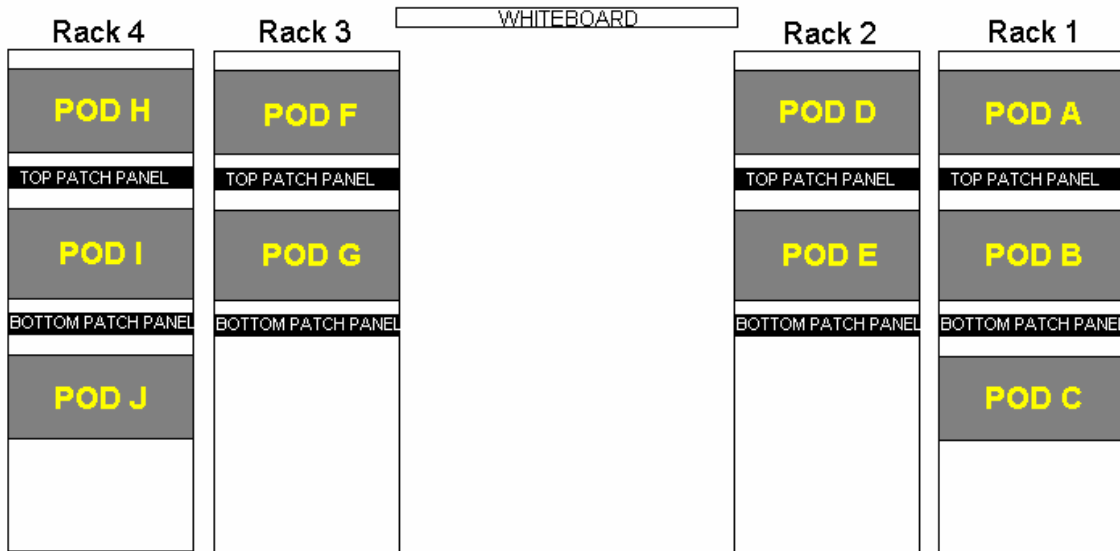
Acknowledgements

Credit for the content in this lab goes to Rick Graziani, an instructor at Cabrillo College that knows Cisco routers very very well and designed the labs this lab was derived from. Many thanks also to Kumar Viswanath and Vladi Petkov who was instrumental in getting the lab setup in BE168 to a usable stage and modifying the lab content to fit the equipment setup here.

Appendix A-1: BE 168 Computer and Rack Layout



Appendix A-2: Rack Layout



Appendix A-3: Pod Layout

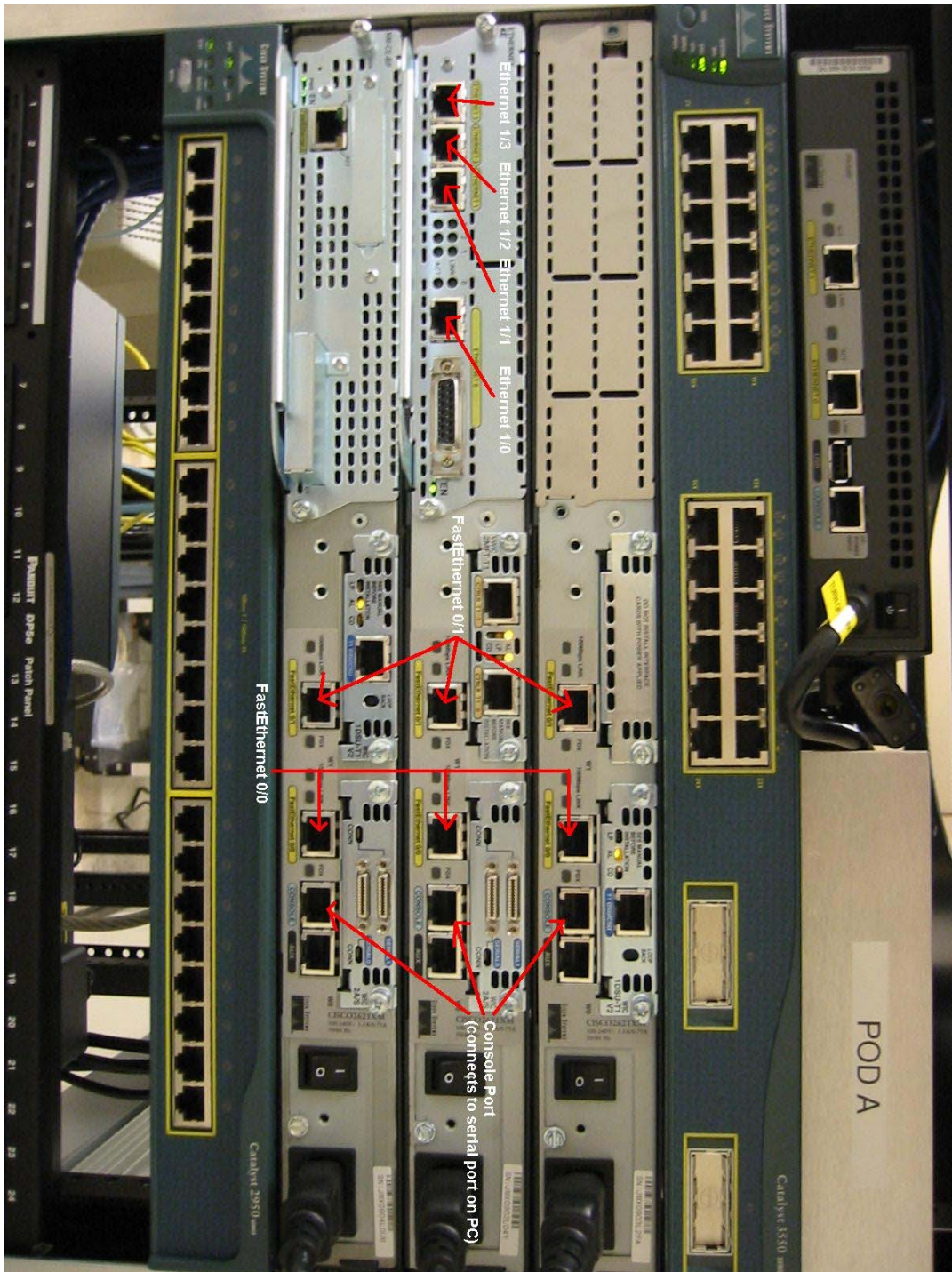


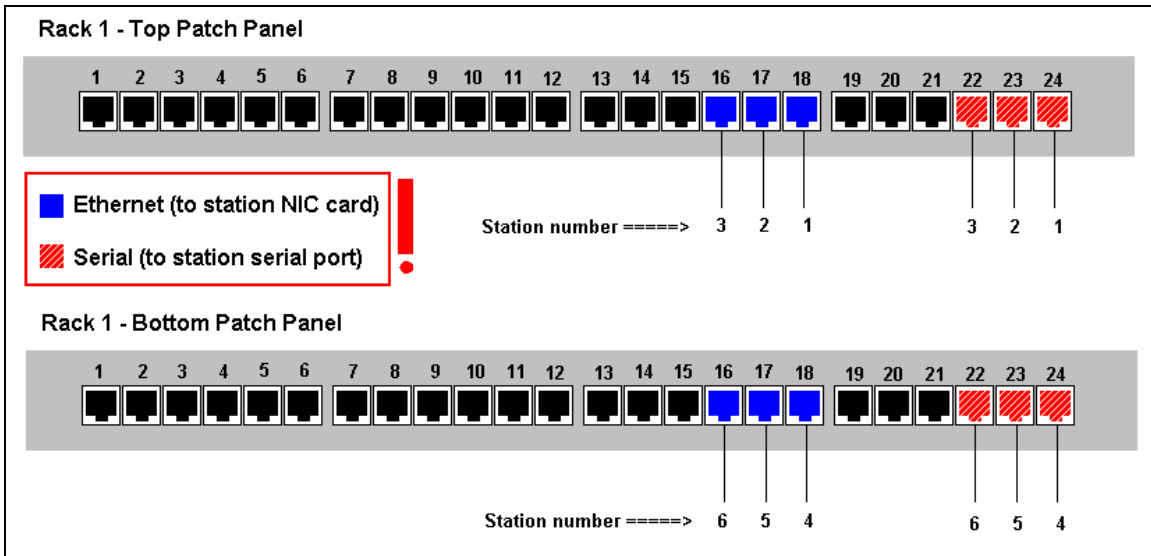
Photo of a Pod A on Rack 1. Each pod has 3 identical routers (Cisco 2600 series), which have different modules installed in them. There are many ways of interconnecting the routers, but for this lab, all network connections will be made using Ethernet cables. Each router has 2 built in Fast-Ethernet ports (FastEthernet 0/0 and 0/1) and the middle router has a module that provides 4 extra Fast-Ethernet ports (Ethernet 1/0 – 1/3). The

middle router is thus ideally suited to be the center router in the network topology in this lab.

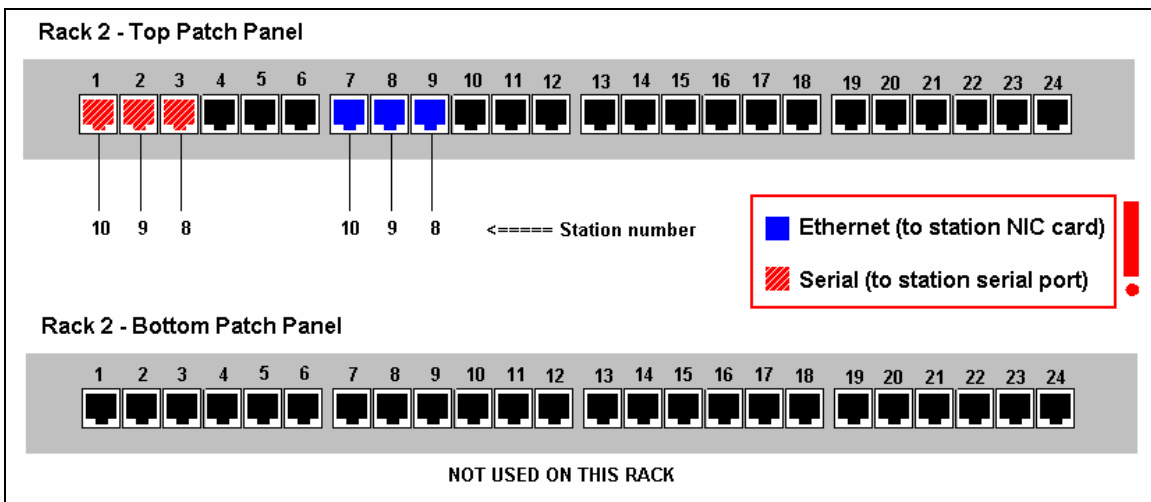
Apart from the routers, each pod has two Cisco switches (not same model, but serve same function). The device on top of the top switch is a firewall. Unfortunately we won't get to play with it in this lab.

Appendix A-4: Patch Panel Layout

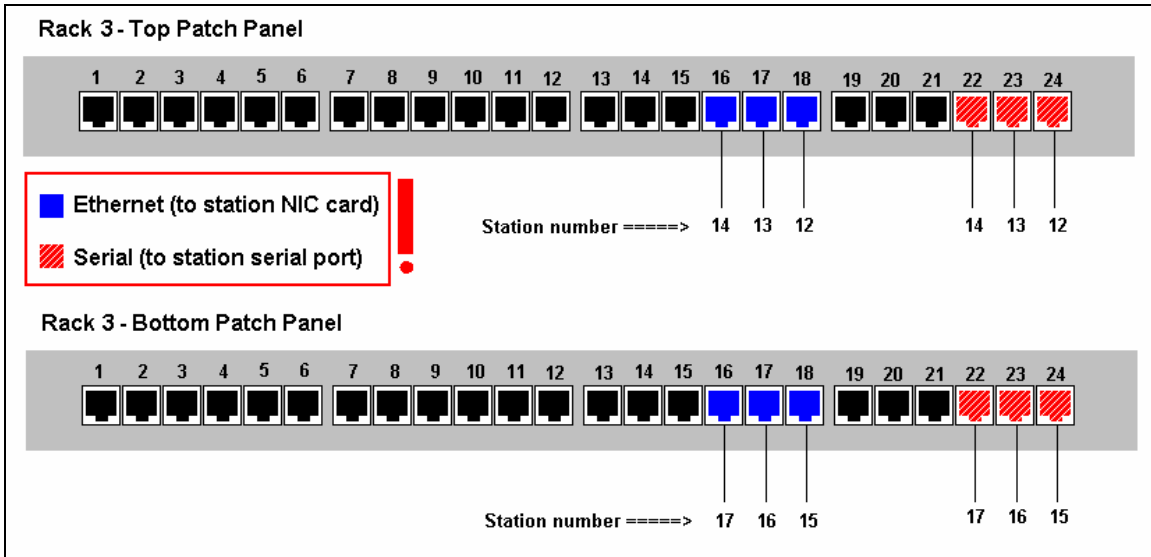
Rack 1



Rack 2



Rack 3



Rack 4

