

THE FLORIDA STATE UNIVERSITY
COLLEGE OF ARTS AND SCIENCES

FLORIDA STATE UNIVERSITY
COMPUTER SCIENCE
INTERNET TEACHING LAB

By

Raymond R. Curci

A Project submitted to the
Department of Computer Science
in partial fulfillment of the
requirements for the degree of
Master of Science
Computer Network and System Administration Track

FSU Computer Science Technical Report #TR-001201

FALL 2000

PROJECT COMMITTEE:
Dr. Lois Hawkes – Computer Science
Jeff Bauer – Office of Technology Integration
Dr. Xin Yuan – Computer Science
Dr. Steve Bellenot - Mathematics

CONTENTS

1	Introduction.....	4
2	Review of Existing Lab Instruction Resources.....	4
2.1	Textbooks.....	4
2.2	Software Simulations.....	5
2.3	CCIE Lab Bootcamps.....	5
3	Project Overview.....	6
3.1	FSU Computer Science ITL Network Lab.....	6
3.2	Framework for Naming and Addressing.....	7
3.2.1	Device Names.....	7
3.2.2	IP Addressing.....	7
3.2.3	Frame-Relay PVC DLCI Labels.....	9
3.3	Router and Switch Hardware.....	12
4	FSU Computer Science ITL Implementation.....	13
4.1	Out-of-band Communications.....	13
4.2	Firewall.....	13
4.3	Network Address Translation (NAT).....	14
4.4	Flexible Interconnections.....	14
4.4.1	Layer 2 Ethernet Switch VLANs.....	14
4.4.2	Physical Serial Cable Mesh.....	15
4.4.3	Frame-Relay WAN Emulation.....	15
4.4.4	GRE Tunnels.....	16
4.5	Physical Router Cabling.....	17
4.5.1	Serial Interfaces.....	17
4.5.2	FDDI Interfaces.....	18
4.5.3	Ethernet and Fast Ethernet Interfaces.....	18
4.6	Guidelines for Creating Labs.....	19
4.6.1	Loopback Interfaces.....	19
4.6.2	Team Challenges.....	20
4.6.3	Hints and Tools.....	20
4.6.4	Network Diagrams.....	21
4.6.5	Instructor Notes.....	23
4.7	Sample Lab Exercises.....	23
5	Conclusion.....	24
5.1	ITL as an Inexpensive Learning Tool.....	24
5.2	Future Directions.....	25
	Appendices.....	28
	Appendix A: Router Hardware Overview.....	28
	Cisco 7000 Core Router.....	28
	Cisco 4500 Mid-Size Router.....	35
	Cisco 2511 Access Server / Router.....	38
	Cisco 3548XL and 3524XL Ethernet Switches.....	39
	Appendix B: Router IOS Software.....	40
	Appendix C: IOS Software Documentation.....	41
	Appendix D: Cisco Router Password Recovery Procedure.....	43

Appendix E: Cisco 2511 Firewall Router Configuration	46
Appendix F: Baseline Router Configuration	49
Appendix G: Linux Scripts	53
Appendix H: Project CD-ROM	58
Appendix I: Acronyms	61

1 Introduction

With the increased importance of large computer networks including the Internet it is desirable to provide Computer Science students with exposure to practical hands-on computer networking. The Internet Teaching Lab (ITL) is a national project sponsored by the Cooperative Association for Internet Data Analysis (CAIDA) to implement hands-on teaching laboratories at 25 U.S. universities during the year 2000. The project aim is to improve curriculum resources as a step toward better preparing the next generation of network engineers and technology workers. The FSU Internet Teaching Lab combines computer networking equipment donated through CAIDA and the FSU Department of Computer Science to build a model instructional networking lab. This FSU Computer Science ITL project implementation includes designing a flexible network of inexpensive routers and switches along with sample lab exercises to augment existing Computer Science coursework. This paper includes many computer networking acronyms that are defined in Appendix I.

2 Review of Existing Lab Instruction Resources

2.1 Textbooks

There are many good books on computer networking such as Tannenbaum¹, but they tend to focus on theory and are lacking the practical information required for building real-world computer networks. As a response to this lack of practical computer network material, one of the major network equipment vendors, Cisco Systems, has created their own publishing company. Cisco Press has published several texts with extensive practical network examples on network architecture², TCP/IP protocol³ and routing protocols⁴ to fill this void. Additionally, they have published texts on router⁵ and switch⁶ configuration that include configuration details with examples in a manner easier to understand than the technical product manuals. There are a few texts focused on teaching practical networking with examples such as Caslow⁷ and Hutnik⁸, but these require the student have access to a large number of expensive routers to try out the examples. In general, textbooks tend to either ignore practical hands-on networking, or provide examples with exercises requiring expensive equipment out of reach for the average student.

¹ Andrew Tanenbaum. *Computer Networks, 3rd edition*. Prentice Hall. 1996.

² Bassam Halabi. *Internet Routing Architectures*. Cisco Press. 1997.

³ Jeff Doyle. *CCIE Professional Development: Routing TCP/IP Volume I*. Cisco Press. 1998.

⁴ Thomas M. Thomas II. *OSPF Network Design Solutions*. Cisco Press. 1998.

⁵ Laura Chappell. *Advanced Cisco Router Configuration*. Cisco Press. 1999.

⁶ Kennedy Clark and Kevin Hamilton. *CCIE Professional Development: Cisco LAN Switching*. Cisco Press. 1999.

⁷ Andrew Bruce Caslow and Valeriy Pavlichenko. *Cisco Certification: Bridges, Routers and Switches for CCIEs*. Prentice Hall. 1999.

⁸ Stephen Hutnik and Michael Satterlee. *All-In-One CCIE Lab Study Guide*. McGraw-Hill. 2000.

2.2 Software Simulations

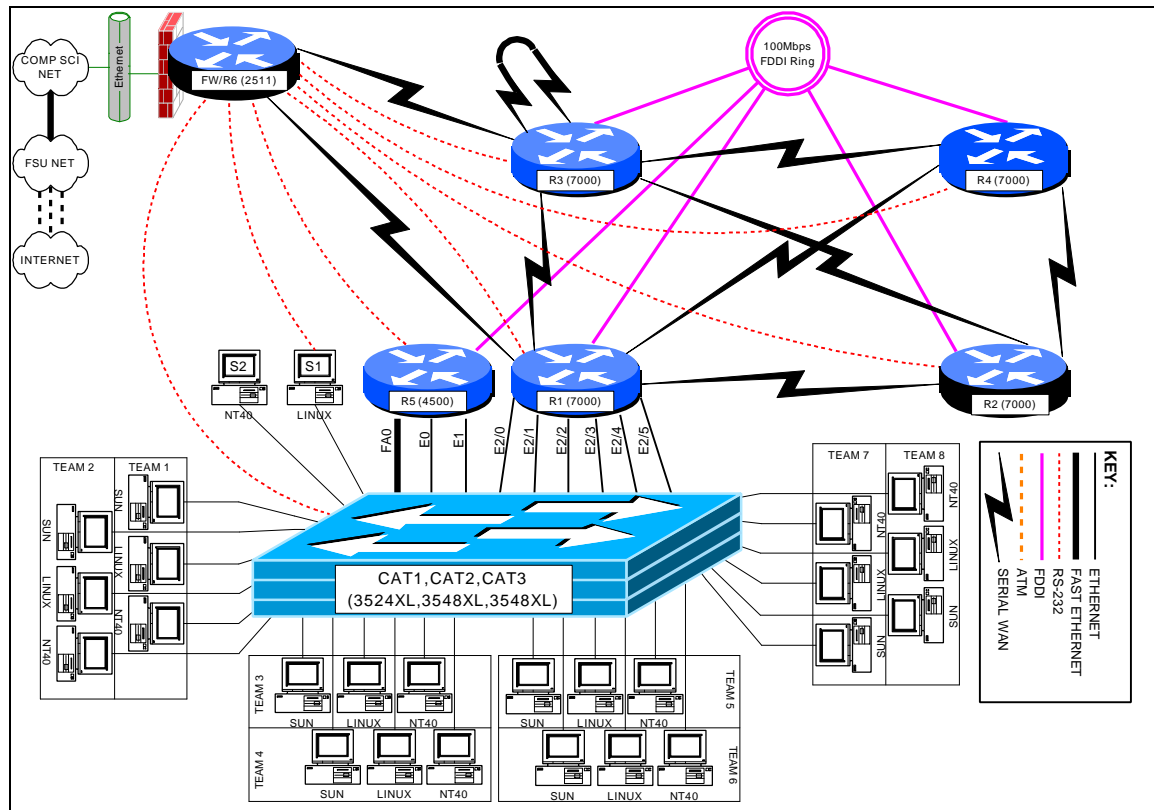
Cisco Systems has developed a series of PC-based software lab simulations to help train engineers without expensive hardware. These simulations are included in a product called Cisco Interactive Mentor (CIM). As of this writing, there are CIM modules on IP routing, ISDN, Voice over IP, Voice/Video, and LAN switching. These are helpful as training material but only simulate a small subset of router functions. Many tools that are helpful in a lab learning environment such as internal testing tools (PING, TRACEROUTE, TTCP), debug mode output, and the ability to simultaneously debug from two different devices on a real network are lacking.

2.3 CCIE Lab Bootcamps

Some vendors offer “bootcamp” classes, generally focused on preparing students for passing certification tests such as the CCIE (Cisco Certified Internetworking Expert) Lab practical exam. CCIE is a very marketable certification. Starting salaries for professionals holding the CCIE certification are typically in excess of \$100K per year. In these bootcamp classes, each student typically has an identical stack of 6-8 routers for building sample networks during the course of an accelerated one week class. Because of the complexity and volume of material to cover, these classes do not work nearly as well as when the training is delivered over a longer period of time. The cost for these bootcamp classes is also prohibitively expensive, typically \$3,000 in tuition for a single one-week course.

3 Project Overview

3.1 FSU Computer Science ITL Network Lab



The FSU Computer Science ITL network lab physically consists of a room with twenty student workspaces, each with three PC workstations. Each workspace houses a surface mount fixture with six RJ45 jacks wired to a central RJ45 patch panel on a telco relay rack compliant with the EIA568 building wiring standard. Each PC uses a patch cable to attach to the surface mount fixture. Each 8-position jack connects with a 4-pair 24 gauge category 5e unshielded twisted pair cable. This cable is suitable for not only 10baseT and 100baseTX ethernet, but also gigabit ethernet over copper, T1 circuits, 56K circuits, ISDN PRI circuits, ISDN BRI circuits, token ring over UTP, and POTS (Plain Old Telephone Service). Normally, patch cables at the relay rack will connect the active connections to 10/100 ethernet ports on a pair of Cisco 3548XL layer 2 switches. Since only 3 of the 6 cables to each workspace will normally be in use, there is flexibility to add additional devices at the workspace to connect back to the central relay rack or to another workspace. The two Cisco 3548XL switches use an IEEE 802.1Q 1000baseSX gigabit ethernet trunk to connect to each other, and to a Cisco 3524XL switch at a remote location over multimode 62.5 μ /125 μ fiber. The remote Cisco 3524XL switch connects to ethernet and fast ethernet ports on the lab routers. The VLAN capabilities of the layer-2

switches allow the student PC ethernet ports and router ethernet ports to be grouped into VLANs with software reconfiguration. The core routers also have serial and FDDI interconnections between each other. A Cisco 2511 router provides firewalled access to the departmental network, network address translation, and out-of-band communication to the EIA RS-232-C console ports on lab devices.

3.2 Framework for Naming and Addressing

Many different naming addressing schemes are possible for a network lab environment, however, adopting some conventions as outlined below help eliminate confusion. These conventions also help keep a focus on the interesting aspects of networking with less time spent on the mechanics.

3.2.1 Device Names

Each router is given a short name such as “r1”, “r2”, “r3”, etc. The router console ports attach the asynchronous lines of the r6 / firewall router “line1”, “line2”, “line3”, etc., respectively. The Cisco catalyst ethernet switches are named “cat1”, “cat2”, and “cat3”. Two test server PCs are labeled “s1” (Linux) and “s2” (NT 4.0 server).

Name	Model	r6/fw Line
r1	Cisco 7000	line1
r2	Cisco 7000	line2
r3	Cisco 7000	line3
r4	Cisco 7000	line4
r5	Cisco 4500	line5
r6/fw	Cisco 2511	n/a
cat1	Cisco 3524XL	line7
cat2	Cisco 3548XL	n/a
cat3	Cisco 3548XL	n/a
s1	Linux PC	line8
s2	WinNT PC	n/a

3.2.2 IP Addressing

Devices inside the FSU Computer Science ITL lab utilize RFC1918 private IP address space. Normally, the CIDR block of 256 class C networks, 192.168.0.0/16 is utilized. These class C networks are generally deployed using a classful 24-bit subnet mask (i.e. /24). (The shorthand /24 indicates a network mask of 255.255.255.0.) Classful masks avoid VLSM problems when making use of classful routing protocols such as RIP version 1 or IGRP. The FDDI backbone uses network 1. Networks for connections between routers are formed by concatenating the integer router identifiers with the smallest integer first. (i.e. a link between r3 and r6 is network 36). Since loopback

interfaces connect a router to itself, the router identifier is concatenated with itself to address the virtual loopback0 interface on each router. Ethernet and fast ethernet port networks are all divisible by 10 and derived by multiplying the team number times 10. The third octet of the IP address matches the network number as shown in the following table.

LINK	TYPE	NET	IP NETWORK
backbone	fddi	1	192.168.1.0/24
r1-r1	loopback	11	192.168.11.0/24
r1-r2	serial	12	192.168.12.0/24
r1-r3	serial	13	192.168.13.0/24
r1-r4	serial	14	192.168.14.0/24
r1-r6	serial	16	192.168.16.0/24
r2-r2	loopback	22	192.168.22.0/24
r2-r3	serial	23	192.168.23.0/24
r2-r4	serial	24	192.168.24.0/24
r3-r3	loopback	33	192.168.33.0/24
r3-r4	serial	34	192.168.34.0/24
r3-r6	serial	36	192.168.36.0/24
r4-r4	loopback	44	192.168.44.0/24
r5-r5	loopback	55	192.168.55.0/24
r6-r6	loopback	66	192.168.66.0/24

The last octet of the IP address indicates either the router identifier for networks between routers, or the number 1 for ethernet interfaces that connect routers to student PCs.

ROUTER	INTERFACE	ABBREVIATION	IP ADDRESS	DTE/DCE
R1	Loopback0	L0	192.168.11.1/24	
	Fddi0/0	FD0/0	192.168.1.1/24	
	Serial1/2	S1/2	192.168.12.1/24	DTE
	Serial1/3	S1/3	192.168.13.1/24	DTE
	Serial1/4	S1/4	192.168.14.1/24	DTE
	Serial1/6	S1/6	192.168.16.1/24	DTE
	Ethernet2/0	E2/0	192.168.10.1/24	
	Ethernet2/1	E2/1	192.168.20.1/24	
	Ethernet2/2	E2/2	192.168.30.1/24	
	Ethernet2/3	E2/3	192.168.40.1/24	
	Ethernet2/4	E2/4	192.168.50.1/24	
	Ethernet2/5	E2/5	192.168.60.1/24	
R2	Loopback0	L0	192.168.22.2/24	
	Fddi0/0	FD0/0	192.168.1.2/24	
	Serial1/1	S1/1	192.168.12.2/24	DCE
	Serial1/3	S1/3	192.168.23.2/24	DTE
	Serial1/4	S1/4	192.168.24.2/24	DTE
R3	Loopback0	L0	192.168.33.3/24	
	Fddi0/0	FD0/0	192.168.1.3/24	
	Serial1/1	S1/1	192.168.13.3/24	DCE
	Serial1/2	S1/2	192.168.23.3/24	DCE
	Serial1/4	S1/4	192.168.34.3/24	DTE
	Serial1/6	S1/6	192.168.36.3/24	DTE
R4	Loopback0	L0	192.168.44.4/24	
	Fddi0/0	FD0/0	192.168.1.4/24	
	Serial1/1	S1/1	192.168.14.4/24	DCE
	Serial1/2	S1/2	192.168.24.4/24	DCE
	Serial1/3	S1/3	192.168.34.4/24	DCE
R5	Loopback0	L0	192.168.55.5/24	
	Fddi0	FD0	192.168.1.5/24	
	FastEthernet0	FA0	192.168.70.1/24	
	Ethernet0	E0	192.168.80.1/24	
	Ethernet1	E1	192.168.90.1/24	
R6	Loopback0	L0	192.168.66.6/24	
	Ethernet0	E0	128.186.121.88/24	
	Serial0	S0	192.168.16.6/24	DCE
	Serial1	S1	192.168.36.6/24	DCE

3.2.3 Frame-Relay PVC DLCI Labels

Part of router r3 can be configured as a frame-relay switch. Since all routers with serial ports have a serial connection to r3, and since r3 has a serial cable looped back to itself, it is an ideal router to emulate a frame-relay switch. Frame-relay uses DLCI numbers to

identify PVCs. DLCIs can be different on both ends of a PVC and serve only to identify the PVCs. Since DLCI numbers are integers in the range from 16 through 1007 inclusive, a convenient convention is to label the DLCIs as a 3-digit integer of the form X0Y where X is the frame relay port number for the PVC and Y is the destination port number. Suppose we consider a PVC between frame-relay switch port 2 and port 4 which connect to router r2 and router r4 respectively. In that case, router r2 would use PVC 204 to reach router r4, while router r4 would use PVC 402 to reach router r2. The following table shows all DLCIs that would need to be defined to build a full mesh of PVCs between the five routers that have serial ports.

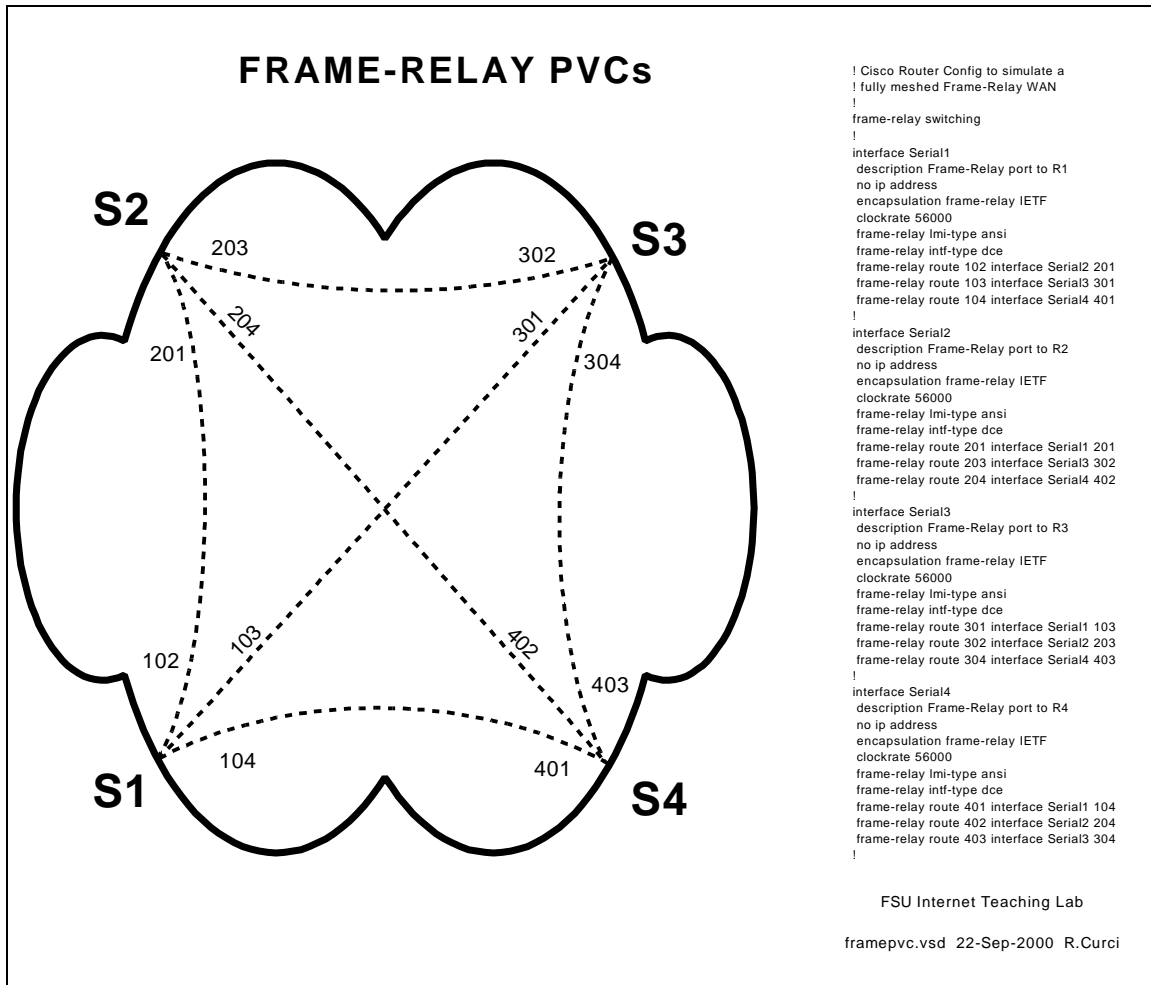
From:	To:	To:	To:	To:	To:
	Serial1/1	Serial1/2	Serial1/3	Serial1/4	Serial1/6
Serial1/1		102	103	104	106
Serial1/2	201		203	204	206
Serial1/3	301	302		304	306
Serial1/4	401	402	403		406
Serial1/6	601	602	603	604	

```

! Cisco Router Config to for R3 to simulate a fully meshed Frame-Relay WAN
! Connect ports S1/1, S1/2, S1/3, S1/4, S1/6 to router r1, r2, r3, r4, r6
respectively.
!
frame-relay switching
!
interface Serial1/1
description Frame-Relay port to R1
no ip address
encapsulation frame-relay IETF
clockrate 2000000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 102 interface Serial1/2 201
frame-relay route 103 interface Serial1/3 301
frame-relay route 104 interface Serial1/4 401
frame-relay route 106 interface Serial1/6 601
!
interface Serial1/2
description Frame-Relay port to R2
no ip address
encapsulation frame-relay IETF
clockrate 2000000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 201 interface Serial1/1 102
frame-relay route 203 interface Serial1/3 302
frame-relay route 204 interface Serial1/4 402
frame-relay route 206 interface Serial1/6 602
!
interface Serial1/3
description Frame-Relay port to R3
no ip address
encapsulation frame-relay IETF
clockrate 2000000
frame-relay lmi-type ansi
frame-relay intf-type dce

```

```
frame-relay route 301 interface Serial1/1 103
frame-relay route 302 interface Serial1/2 203
frame-relay route 304 interface Serial1/4 403
frame-relay route 306 interface Serial1/6 603
!
interface Serial1/4
description Frame-Relay port to R4
no ip address
encapsulation frame-relay IETF
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 401 interface Serial1/1 104
frame-relay route 402 interface Serial1/2 204
frame-relay route 403 interface Serial1/3 304
frame-relay route 406 interface Serial1/6 604
!
interface Serial1/6
description Frame-Relay port to R6
no ip address
encapsulation frame-relay IETF
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 601 interface Serial1/1 106
frame-relay route 602 interface Serial1/2 206
frame-relay route 603 interface Serial1/3 306
frame-relay route 604 interface Serial1/4 406
```



3.3 Router and Switch Hardware

- Cisco 7000 Core Router (r1,r2,r3,r4)
- Cisco 4500 Mid-Size Router (r5)
- Cisco 2511 Small Router / Access Server (r6)
- Cisco 3524XL Layer 2 Switch (cat1)
- Cisco 3548XL Layer 2 Switch (cat2,cats)

The Cisco 7000 routers are large systems once deployed on the MCI Internet backbone. They have both FDDI and serial interface cards. One additionally has a 6-port ethernet card. The Cisco 4500 has a FDDI port, two ethernet ports, and a fast ethernet port. The 7000 and 4500 routers are programmed by the students in these labs. The Cisco 2511 router provides two serial ports, an ethernet port, and 16 asynchronous ports. It provides both firewall functionality and out-of-band access to other lab devices through their console ports. The Cisco 3524XL and 3548XL switches provide connectivity between the router ethernet ports and student PC ethernet ports. They also tie together the router equipment with the network lab through a gigabit ethernet trunk. This allows for the

router equipment and student PCs to be located in different rooms to reduce the ambient noise level in the student network lab and provide a higher level of physical security for the router equipment. See the Appendix A for more detailed information.

4 FSU Computer Science ITL Implementation

4.1 Out-of-band Communications

It is important in a network lab environment to be able to configure the environment quickly. Because changes typically include modifying the addressing scheme, changing the routing protocols, or even erasing the configuration, it is not always possible to use the TCP/IP protocol to remotely access the router and switch devices directly. All router and switch devices in the ITL lab have RS232 console ports that can be used to configure the devices using a directly connected dumb terminal or terminal emulator. This approach solves the problem of configuring the network devices but requires physically moving the console cable from one device to the next for access. Moving cables is possible when the operator is near the equipment but inconvenient or impossible when distance separates the user from the router equipment. A router feature called “reverse telnet” on the Cisco 2511 router/access server solves this problem. A user can log into the firewall 2511 router and type an alias such as “r1”, “r2”, etc., to connect to the corresponding router console port. Since the 2511 router has 16 async RS232 ports, it is possible to leave one async port permanently attached to each router and switch console port. For example, when an instructor wants to reconfigure the setup on all five student routers, each router can be erased, rebooted, and reprogrammed in a matter of minutes. With the appropriate passwords, this reconfiguration can even be performed remotely.

4.2 Firewall

Router r6 doubles as a firewall. It has a permanent ethernet connection to the FSU Computer Science network and serves as the gateway between the ITL lab network and the outside. Since this is the only lab device connecting to the outside network, it provides a convenient single “choke point.” Access lists on this router’s ethernet port are used to help secure the lab by controlling what traffic is permitted to flow between the lab and outside networks. In general, the firewall limits access from outside into the lab network, but allows the lab network devices to access the outside. Since many assignments in the networking lab call for students to access the web to download files, this is very convenient. During times when more dangerous assignments are assigned, these access lists can be adapted to be more restrictive. For example, when security network probe tools like NMAP are explored, it may be prudent to prevent lab devices from accessing systems outside the Computer Science Department. The two serial ports on this router normally provide two 2Mbit/sec links to routers r1 and r3. See the appendix for a sample configuration of this router.

4.3 Network Address Translation (NAT)

Router r6 contains runs Cisco IOS v12.0 software which contains a Network Address Translation feature. The ethernet on router r6 is tagged as “outside” while all other interfaces are “inside.” When an IP packet is routed between an outside and inside interface, network address translation takes place. Normally, all devices inside the lab are configured with RFC1918 private IP address space. When a lab device attempts to reach a device outside the lab, the packet follows the default route to r6 where an unused port number is selected and the packet sent out the ethernet port. To devices outside the lab, router r6 appears as if it is a multiuser computer system. Response packets are translated in the opposite direction. Since lab devices only have private addresses, they are generally protected from the Internet, yet have access to the Internet. The command “show ip nat translation” can be used to see a snapshot of the current global and local address and port mappings. Normally, these mappings occur dynamically and overload the r6 ethernet port IP address by multiplexing using unused 16-bit port numbers. It is also possible to statically map an IP address. For example, in the course of this project, it has been handy to be able to access Linux server S1 and NT server S2. Inside the lab network, S1 and S2 have IP addresses 192.168.10.2/24 and 192.168.10.3/24 respectively. By statically mapping these local IP addresses to global addresses 128.186.121.89 and 128.186.121.90, and further defining the names itl2.cs.fsu.edu and itl3.cs.fsu.edu, these servers can be reached from outside using the fully qualified domain name.

4.4 Flexible Interconnections

Flexibility in how the lab network devices are interconnected improves the lab versatility. It is especially desirable to have the capability of reconfiguring the network connections without the need to physically move cables. Moving cables requires physical access and is inconvenient when the user is located remotely and is also prone to hardware problems such as bending cable connector pins or fouling fiber optic connectors. Flexibility in how routers are connected without the need for manual cable moves is achieved with three techniques:

1. Layer 2 Ethernet Switch VLANs
2. Physical Serial Cable Mesh
3. Frame-Relay WAN Emulation
4. GRE Tunnels

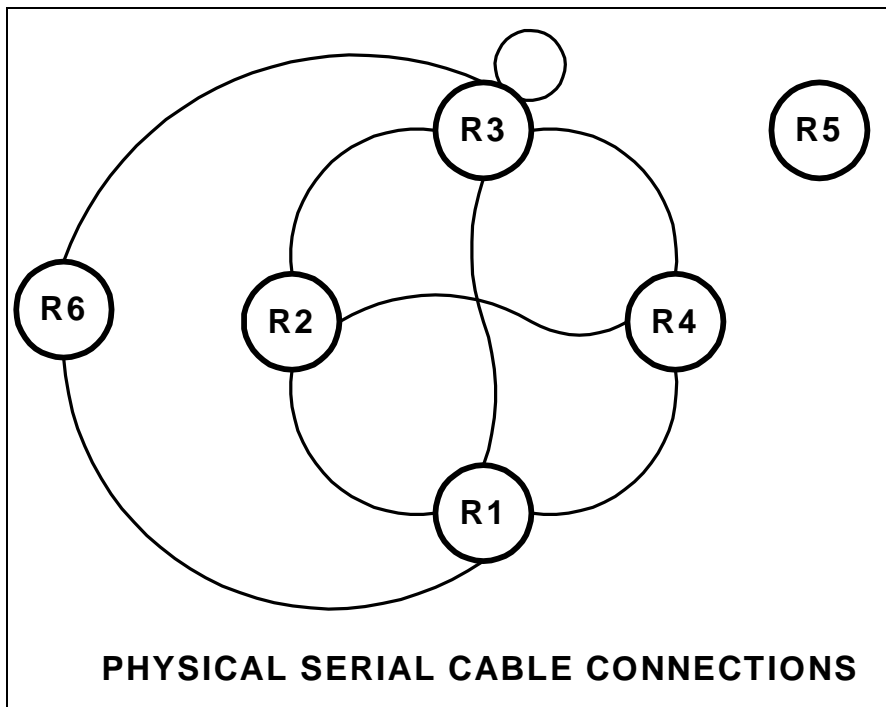
4.4.1 Layer 2 Ethernet Switch VLANs

Modern layer 2 ethernet switches such as the Cisco Catalyst 3524XL and 3548XL have the capability of implementing Virtual Local Area Networks (VLANs) and trunking. Most layer 2 ethernet switches default to logically acting as a multiport bridge where all ports are part of the same layer 2 network. VLANs allow the ports to be grouped, or colored, and segregated into different virtual LANs. Additionally, trunking protocols like IEEE 802.1Q and ISL (Inter Switch Link) allow single physical connections between

switches to carry multiple VLANs by prepending data link frames with a header indicating the VLAN. In effect, trunking allows a set of interconnected switches to logically act as a single large switch even when the switches are in different locations. For example, the student routers have a total of nine ethernet and fast ethernet ports that can each be assigned a different VLAN. The student lab PCs can then be logically connected to any router ethernet or fast ethernet port by assigning their ports to the appropriate matching VLAN. This technique allows the set of router ethernet ports and lab PC ethernet ports to be logically grouped in any combination of mutually exclusive subsets.

4.4.2 Physical Serial Cable Mesh

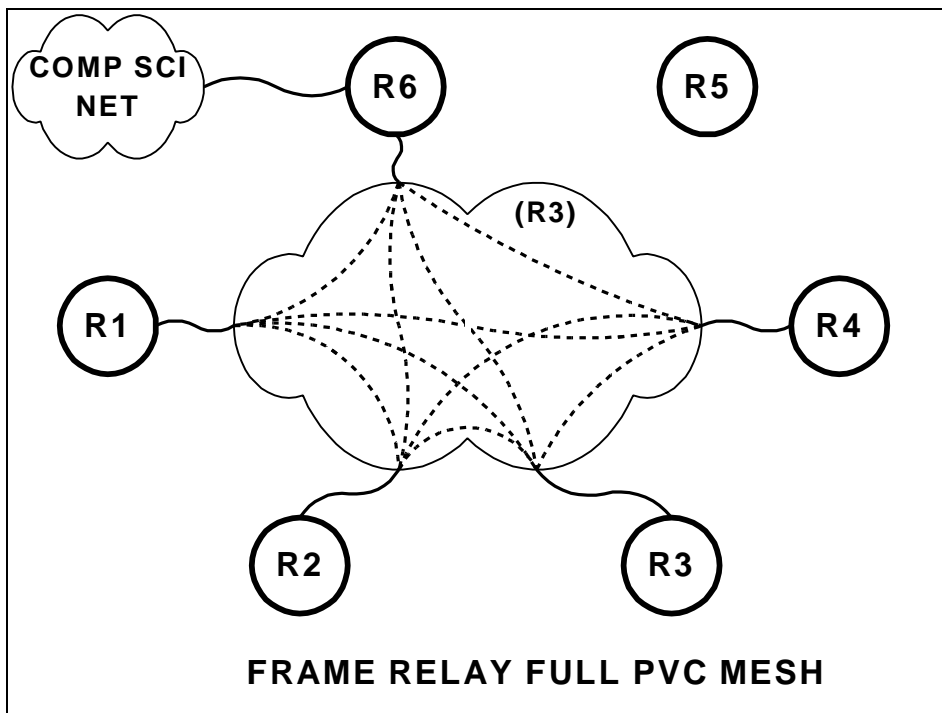
The four Cisco 7000 routers (r1,r2,r3,r4) each contain several serial ports. A set of $(N \times (N - 1))/2 = 6$ serial cables provide a full mesh among routers r1, r2, r3, and r4. Two of these routers also have serial connections to the r6/firewall router. Additionally, r3 also has a physical serial cable between two of its serial ports to facilitate the frame-relay configuration described below. The only other router, r5, has no serial ports and therefore no serial cables. By selectively configuring these serial ports to be either enabled or administratively disabled allows for many different combinations of serial connectivity without the need for physically moving any cable connections. The DCE clock rate can also be adjusted to simulate different speed WAN connections.



4.4.3 Frame-Relay WAN Emulation

Frame-Relay is a Wide Area Network (WAN) technology. Routers or frame-relay access devices (FRADs) physically connect to a redundant network of frame-relay switches. Permanent virtual circuits (PVCs) are created to build a logical partial or full mesh data

link network between the devices. Cisco routers contain a feature that allows a subset of router serial ports to emulate a frame-relay WAN network in software. This emulation supports the frame-relay link management interface (LMI) but not the forward explicit congestion notification (FECN) or backward explicit congestion notification (BECN) facility. Software configuration commands allow for PVCs to be created between any of the emulated frame-relay switch ports to create any mesh of connections. Data Link Connection Identifiers (DLCIs) identify logical PVCs on each port allowing multiple PVCs to terminate on a single physical port including multiple PVCs between the same two endpoints in parallel. The Cisco IOS software also allows the router frame-relay ports to be associated with a physical interface, point-to-point subinterface, or point-to-multipoint subinterface for a great amount of flexibility. Since all lab routers with serial interfaces have physical connections to router r3, it is an ideal choice to double as a frame-relay switch. A full mesh of PVC connections can be constructed between 5 routers using $N \times (N - 1)/2 = 10$ PVCs. Multiple PVCs between the same two routers can also be constructed to form parallel paths to explore load balancing techniques.



4.4.4 GRE Tunnels

Generic Route Encapsulation (GRE) tunnels are a flexible software device to build virtual point-to-point interfaces between routers. Tunnels encapsulate traffic between router endpoints. Probably the most common use of tunnels is to encapsulate non-IP traffic through an IP-only core network. It is also possible to tunnel RFC1918 private addresses through the public Internet with this device. In a situation where a point-to-point connection is needed between two routers where none exists, a tunnel can be implemented. For example, if we needed router r1 and router r5 to have a point-to-point

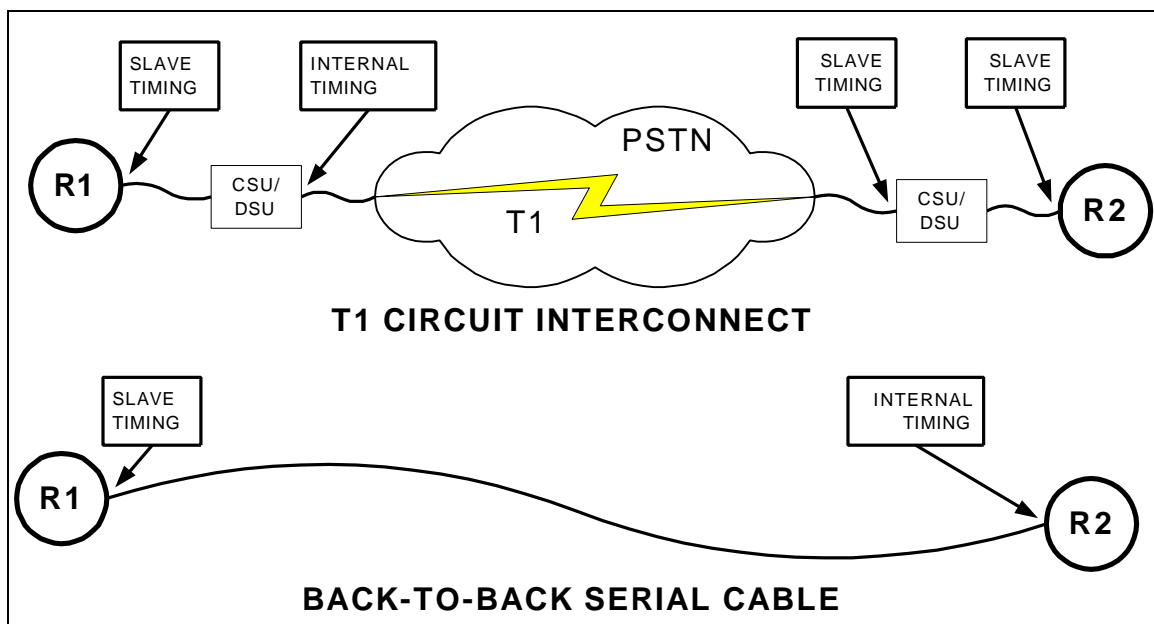
connection in order to do exterior BGP peering, a tunnel can be used. The lab exercise on EBGp protocol explores the use of tunnels.

4.5 Physical Router Cabling

4.5.1 Serial Interfaces

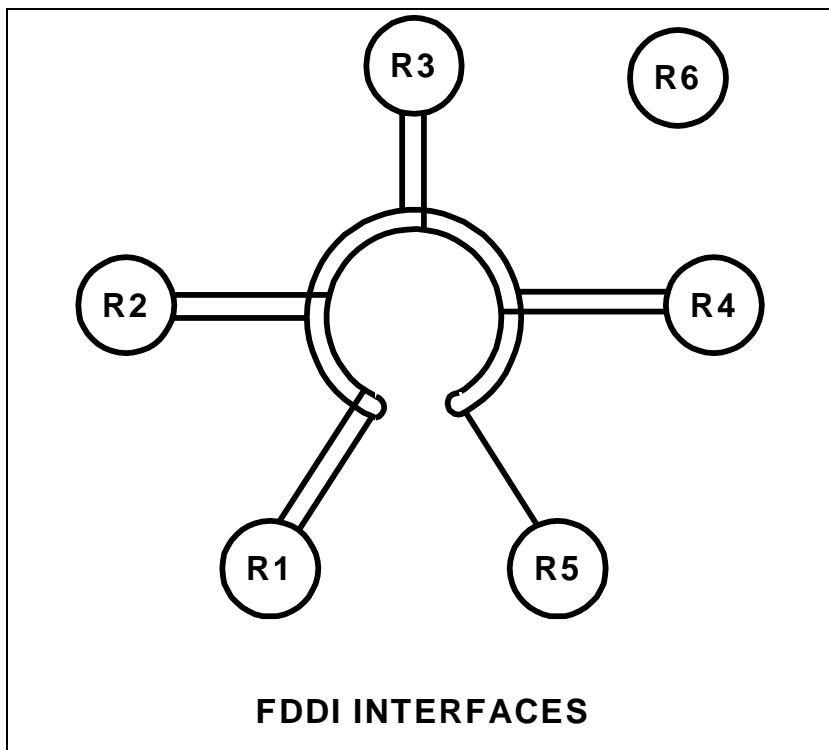
Serial connections in the ITL lab connect router serial ports without the use of any CSU/DSUs. Normally, serial connections between routers would use a phone company T1 or 56K DDS circuit where the router port is configured as data terminal equipment (DTE) and CSU/DSU configured as data communications equipment (DCE). In this situation, the CSU/DSU would provide clocking to the router which slaves its timing using the CSU/DSU clock source. With a direct serial connection between routers without CSU/DSUs, one end must be configured as DCE and provide clocking, while the other end must be configured as DTE and slave its timing off the clock source on the other end. Router serial interfaces acting as DCE must use the “clock rate xxxxxx” command to supply the clocking. The serial cables used in the ITL lab have one end clearly labeled “DTE” and the other end labeled “DCE”. In all cases where a serial cable connects two routers, the DCE side connects to the router with the higher integer identifier. For example, the cable between routers r2 and r4 is DTE on the r2 end and DCE on the r4 end.

Where possible, the serial cable interface name also corresponds to where the other end of the cable terminates. For example, router r3 has serial cables that connect it to r1, r2, r3 (itself), r4, and r6 that are on ports S1/1, S1/2, S1/3, S1/4, and S1/6 respectively.



4.5.2 FDDI Interfaces

Each of the routers r1, r2, r3, r4, and r5 has an FDDI port and form a backbone FDDI ring. No FDDI concentrator is used, so these devices are connected in sequence r1→r2, r2→r3, r3→r4, and r4→r5 but not r5→r1. Since r1, r2, r3, and r4 have DAS (dual-attach station) ports while r5 has a SAS (single attach station) port, the FDDI ring is always in a wrapped state and does not form a fully redundant dual ring. If all FDDI interfaces are up, routers r1 through r5 can communicate over the ring. If, however, one of the routers has its FDDI interface shut down or one router is powered off, it will break the FDDI network into multiple rings. When you want only a subset of routers r1 through r5 to participate on the FDDI ring, you should leave all FDDI interfaces enabled but simply remove any IP address from interfaces that should not participate. Another option is to shut down the FDDI interface on r3 which will make two separate physical FDDI rings – one ring with r1 and r2, and another ring with r4 and r5.

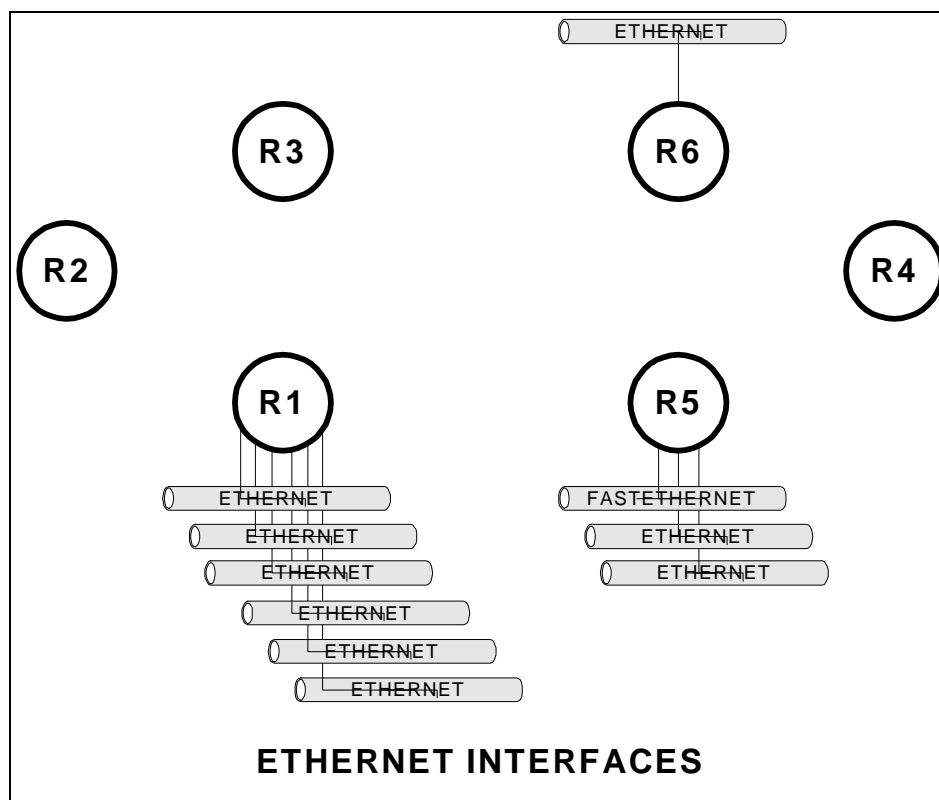


4.5.3 Ethernet and Fast Ethernet Interfaces

Router r1 has six ethernet interfaces while router r5 has one fast ethernet and two ethernet interfaces. These nine interfaces connect to the Cisco Catalyst 3524XL on ports FastEthernet0/1 through FastEthernet0/9 using standard RJ45 category 5 patch cables. Since the r1 ports use DB15 AUI connectors, Allied Telesyn 210TS transceivers adapt these ports to the 10baseT standard. R5 has both 10baseT and AUI ports on its ethernet interfaces, and 100baseTX and MII ports on its fast ethernet interface. Because r5 contains RJ45 connections, transceivers are unnecessary but care must be taken to active the correct physical connector with the interface “media-type” command. By default, the

switch ports are configured to auto sense the port speed and duplex settings. Normally, these nine ports are each placed in different VLANs as indicated in the table below.

Router	Router Interface	Cat3524XL	VLAN
r1	Ethernet2/0	FastEthernet0/1	10
	Ethernet2/1	FastEthernet0/2	20
	Ethernet2/2	FastEthernet0/3	30
	Ethernet2/3	FastEthernet0/4	40
	Ethernet2/4	FastEthernet0/5	50
	Ethernet2/5	FastEthernet0/6	60
r5	FastEthernet0	FastEthernet0/9	70
	Ethernet0	FastEthernet0/10	80
	Ethernet1	FastEthernet0/11	90



4.6 Guidelines for Creating Labs

4.6.1 Loopback Interfaces

Loopback interfaces are virtual router interfaces that can be created on demand which never fail. When a router is connected to a network through multiple physical connections, it is possible for a physical interface to go down while the router remains connected to the network. If a communication session such as a tunnel, ntp, telnet, bgp peering session, etc., is referencing the down interface, it will fail. For this reason, loopback interfaces are often created and an IP address assigned that is used to reference

the router which will remain up as long as the router has some connectivity and an appropriate routing protocol.

In the FSU Computer Science ITL lab environment, loopback interfaces are also useful. No matter what model Cisco IOS router and IOS software is available, many loopback interfaces can be created to make more complex and interesting lab exercises. For example, in the VLSM lab, each router has 4 loopback addresses named loopback0, loopback1, loopback2, and loopback3 each with different addresses and network mask. One aspect of this lab is the focus on using OSPF's ability to summarize a group of directly connected networks into a single aggregate routing advertisement. Another example is the RIP lab where some routers that have no physical ethernet ports use a loopback interface as a substitute. Although not functional like an ethernet interface, a loopback interface is treated almost the same in Cisco IOS and is ideal for experimenting with routing protocols.

4.6.2 Team Challenges

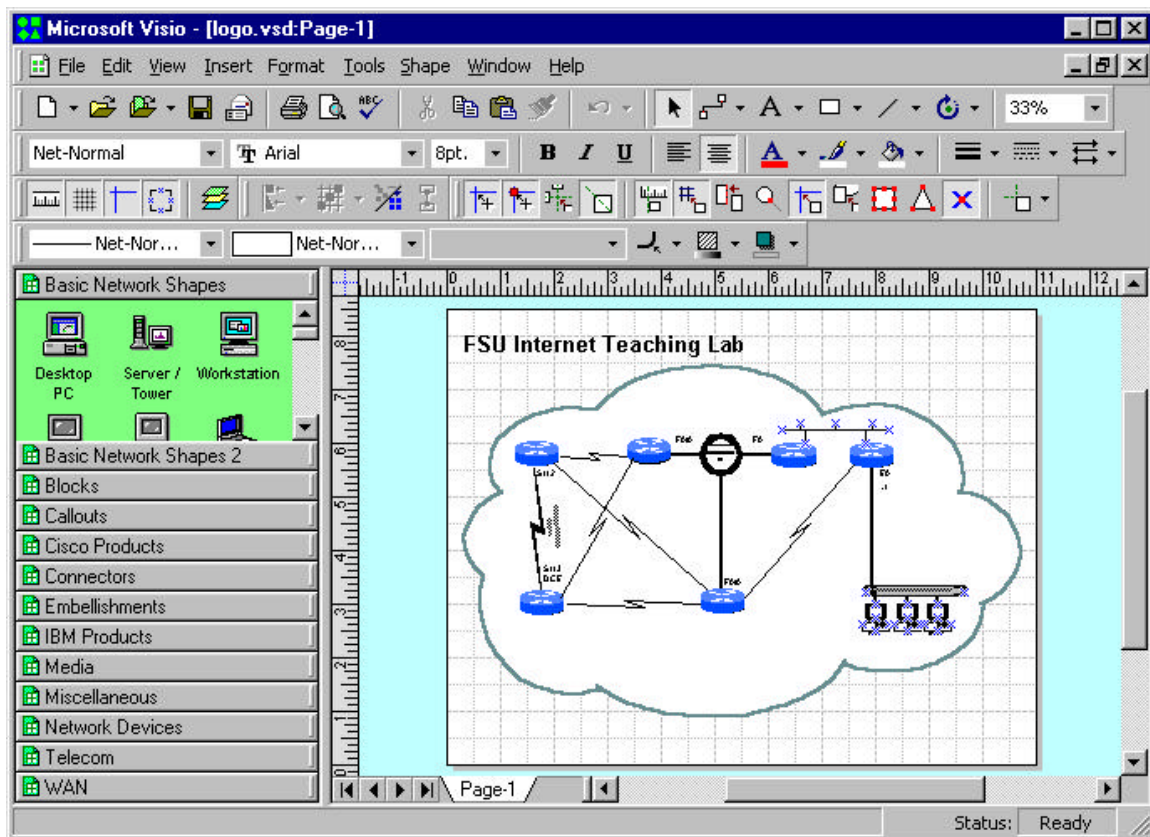
In practice, building and troubleshooting data networks requires a lot of teamwork. For example, if you are configuring a router for your organization, it will often need to communicate or connect to a router in a different organization where you are unlikely to have access. For this reason, it becomes important to clearly define the point of demarcation, IP addressing schemes, routing protocols, announcements of routes, OSPF area numbers, BGP autonomous system numbers, etc. Many of the sample labs include a detailed blueprint -- a detailed network diagram, information on the IP addressing scheme and routing protocols. If each team closely follows the instructions, the network will interoperate. It is also helpful to expose students to the process of working with the entire class of students to define the blueprint for the network. For example, the sample VLSM lab requires the entire class of students to first define a blueprint that defines the IP addressing and subnetting scheme before it can be implemented. This type of exposure is helpful to prepare students for team challenges they will face outside of school.

4.6.3 Hints and Tools

Many of the sample labs try to give students hints and tools rather than answers to questions. Helping students learn where to seek information will help with future challenges. Some hints suggest that the student read the manual section that describes a particular Cisco IOS configuration, show, or debug command. Understanding how to utilize tools and utilities such as IP PING, IP TRACEROUTE, IPX PING, Appletalk PING, and TFTP are helpful for debugging and isolating problems. Less frequently used options like extended IP PING or extended IP TRACEROUTE are also handy tools. With an understanding of how the various network protocols function, even a simple tool like TELNET can be used to connect to services such as WWW, SMTP, and POP3 for testing. When testing access lists, the /SOURCE-INTERFACE option inside the Cisco

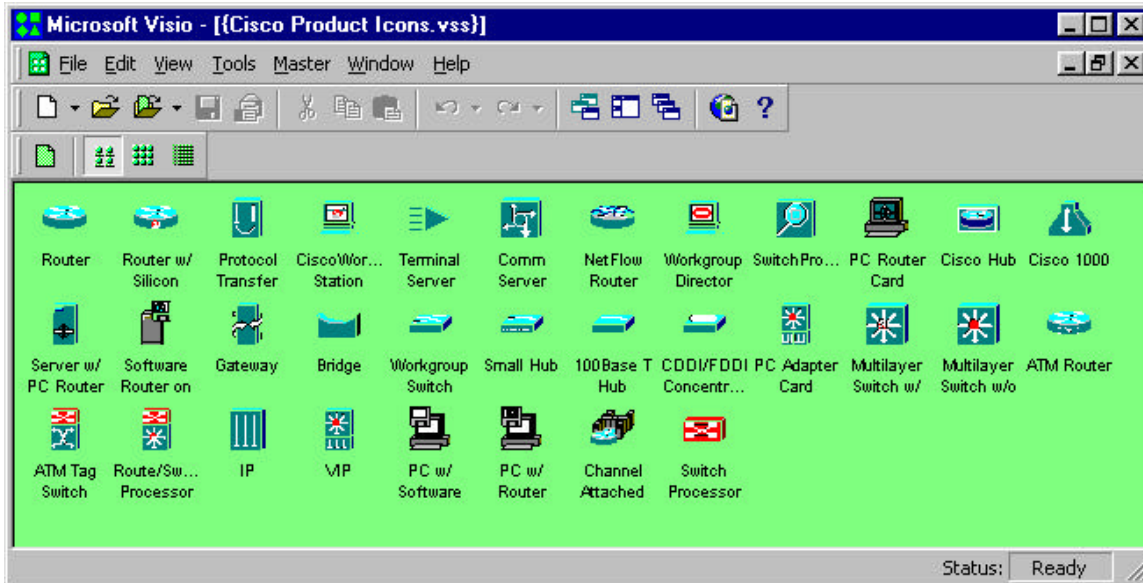
IOS TELNET can change the source IP address of the session which can be enormously helpful for debugging. The use of DEBUG mode and SYSLOG to send debug messages to a UNIX host where the many messages can be post-processed is also a powerful tool. Many of these tools are explored in the sample lab exercises.

4.6.4 Network Diagrams

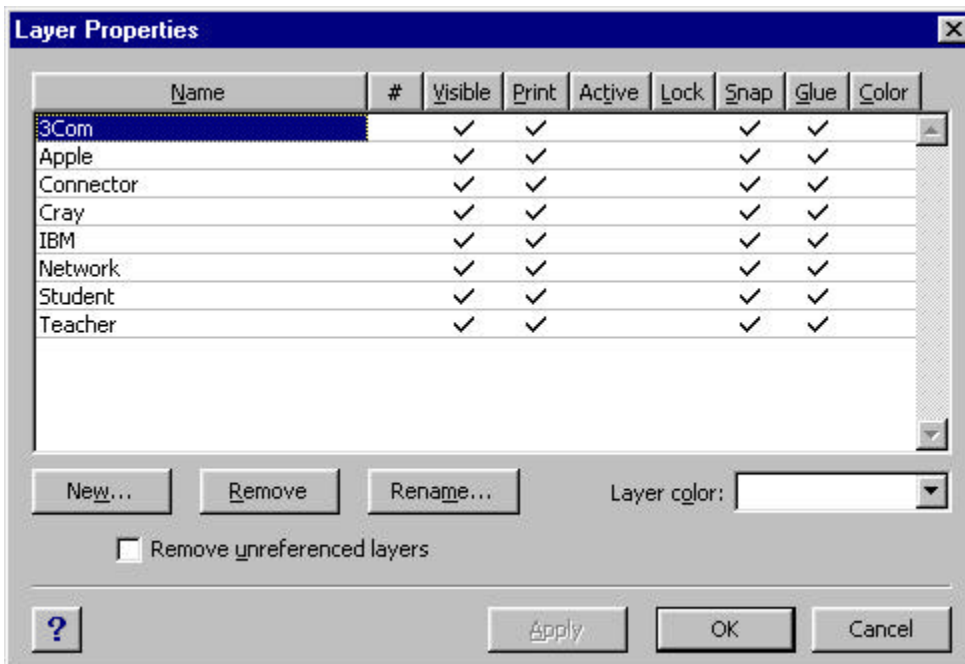


Good network diagrams are invaluable tools for communicating network designs. All of the sample labs contain a detailed network diagram. These diagrams were created with Microsoft Visio 2000 Professional, a Windows-based drawing tool. As of this writing, Visio is the defacto standard for drawing network diagrams and students can expect to receive many Visio e-mail attachments when working in the networking industry. With an FSU site license price of \$27 for software that normally costs \$500, every student should get a copy and become familiar with this utility. Visio uses “stencils”⁹ of graphic elements and connectors to speed the drawing process. Many graphics representing networking and computer components are included with the package. In preparing the sample labs, additional stencils downloaded from Cisco Systems were also used. Copies of these stencils are included in two ZIP archive files on the project CD-ROM.

⁹ These are similar to AutoCAD component libraries.



Another useful feature of Visio is its layering ability. Individual items can be assigned to different layers. Each layer can then be selected to be visible or printed. Some of the drawings for the sample labs have several common layers plus a “student” and “teacher” layer. For example, in the sample Topology Discovery Lab, the drawing can be printed with the student layer turned on and teacher layer turned off for the student, and printed with the settings reversed for the teacher. For many drawings, most of the work in their creation is in the components in the common layers. Maintaining a single drawing with two layers to be toggled on and off is much easier than maintaining separate drawings.



Another useful utility is Adobe Acrobat Writer. It can be installed on a Windows PC and appears to applications as a printer driver. Visio can then “print” a diagram to the driver to create an Adobe Acrobat PDF (Portable Document Format) file. Since the Adobe Acrobat Reader is a free utility that is widely deployed, a drawing in this format can be viewed, magnified, or laser printed without the expense of the Visio software. This format is especially convenient when storing drawings on a web server. Most of the sample labs include a Microsoft Word document that includes an embedded drawing. These embedded drawings are hyperlinked to PDF versions of the drawings that are easier to read and print.

4.6.5 Instructor Notes

The sample labs are written with the Microsoft Word word processor. Student lab exercises and instructor notes are maintained in a single document. A style sheet named “hidden” was created and applied to the sections intended only for the teacher. This style uses a monospaced font, hidden attribute, and a 4 ½ point red border on the right. This allows for printing both a student and teacher version of the lab by selecting whether to print the hidden text. When the hidden text enabled, the teacher notes appear interspersed and can be easily identified by the thick border on the right.

The instructor notes include answers, comments on common pitfalls, sample solutions, additional tables and diagrams, sample command output, etc.

It is much easier to maintain a single lab document file with both student and teacher components than separate documents.

4.7 Sample Lab Exercises

Several sample lab exercises have been written. There are three different types of sample labs.

1. Generic Labs
 - a. Cisco Router Basics (inverse telnet, modes, etc.)
 - b. Cisco Router Debugging (show commands, debug mode)

2. CIS5406 (Computer Network and System Admin) Labs
 - a. Topology Discovery Lab (RIP,SNMP,IPERF,TROUTE)
 - b. Start-From-Scratch Lab (RIP)
 - c. Multiprotocol Lab (IPX, Appletalk)
 - d. Routing Information Protocol Lab (RIP)
 - e. IGP Lab (RIP,OSPF,IGRP,EIGRP,ISIS)
 - f. ACL Lab (access lists, NTP, SYSLOG)
 - g. Frame-Relay Lab (Frame-Relay emulation, RIP, Split-Horizon)

- h. BGP Lab (Exterior BGP protocol, tunnels)
 - i. VLSM Lab (variable length subnetting, OSPF)
3. CEN5515 (Data and Computer Communications) Labs
- a. Spanning Tree Lab (802.1D)
 - b. Count-To-Infinity / Split Horizon Lab (RIP)

The generic labs include exercises to help students become familiar with the mechanics of the Cisco routers. This includes topics like how to log into a router, how to use reverse telnet to access a router console port, regular and enabled modes, configuration mode, etc. It also includes information on common “show” and “debug” commands for isolating and resolving network problems.

The CIS5406 labs are intended to be used as a hands-on lab component of this graduate Computer Network and Systems Administration class. They explore tools to measure network performance, routing management tools, routing protocols, subnetting, access lists, non-IP protocols, etc.

The CEN5515 labs are intended to explore data communications algorithms such as the 802.1D spanning tree protocol, distance vector routing protocols, and link state routing protocols.

Each lab contains a Microsoft Word writeup containing diagrams and exercises. The writeups also contain hidden text for instructors to point out common pitfalls, sample solutions, hints, and examples. By incorporating both components in each document, it can be printed in both a student and teacher version by disabling or enabling the hidden text. Each student version of the writeup is also available in hypertext format for easy web browser access. The hypertext version also has a hyperlink to a detailed network diagram in Adobe portable document format allowing easy printing of high resolution laser copies of the diagrams. All drawings were created in Microsoft Visio 2000 but also available in PDF format. The accompanying CD-ROM includes many other files related to the labs that include sample router configurations, captures of various show commands, routing tables, etc. Many also include additional information in Microsoft Excel spreadsheet format.

5 Conclusion

5.1 ITL as an Inexpensive Learning Tool

Computer networks and computer system administration have become increasingly important topics with the recent proliferation of computer networks, multiuser computer systems, and the Internet. Demand in the job market for professionals to build and maintain these systems continues to grow. Employers are seeking professionals with the right combination of theoretical background, problem solving skills, and practical

experience. Unfortunately, many Computer Science degree programs ignore the practical topics of the industry and focus solely on the theoretical aspects. This is a very similar paradigm to the situation 10 years ago when many students graduating with Computer Science degrees had experience with mainframe computers but little or no exposure to microcomputers. The Florida State University Department of Computer Science has been a leader in this area and has developed the Computer Networking and Systems Administration Masters Track to help prepare students for this important profession.

The FSU Internet Teaching Lab utilizes mostly older networking equipment that has been removed from production networks. Many of the donated items such as the Cisco 7000 routers once deployed on the MCI Internet backbone have been replaced with newer models. Although this equipment is unsupported and will not run the latest IOS software and somewhat obsolete, there is plenty of functionality to be useful as a learning tool. This gives universities like FSU equipment at little or no cost that can be used to help teach students. The students, in turn, will graduate with better practical networking experience and be more desirable as prospective employees of the high tech companies including those who have donated equipment.

Obviously, the goal of a program like the CNSA track should not be to train network technicians who only understand practical aspects with no theoretical background. A better approach is to educate professionals with a broad range of skills and knowledge and in both theoretical and practical areas of this industry who have the ability to learn, grow, and adapt as the computer networking industry changes. The ability to solve problems, grow, and adapt is critical in such a rapidly changing industry.

5.2 Future Directions

There are many topics that could not be explored in this Internet Teaching Lab due to a lack of equipment. Some topics could not be explored because many of the routers only support older IOS software and lack some of the newer features. Still other topics can be explored with the existing lab equipment but were not developed due to project time constraints.

- ISDN
With additional router ISDN PRI and/or BRI ports and an ISDN emulator, it would be possible to explore lab experiments that implement dial on demand routing (DDR). This feature allows for routers to establish backup dial connections upon detecting a failure in the network.
- VoIP / Telephony
Voice over IP is a hot topic. IP telephones and programs like Microsoft NetMeeting can be used to establish voice calls over an IP data network. ISDN PRI, ISDN BRI, FXO, and FXS interfaces are available on routers to experiment with these protocols. With the proper hardware and software, for example, a

router can be connected to a telephone or ISDN line and configured as an H.323 gateway and accessed from a remote IP telephone or PC running NetMeeting. These experiments would require additional hardware and some phone lines, ISDN lines, or simulator.

- ATM
Asynchronous Transfer Mode is an important topic in wide area networks. Lab experimenting would require some rather expensive router ATM interface cards and an ATM switch. Many topics could be covered such as PVCs, SVCs, classical IP over ATM, and LANE.
- QoS
Quality of service is an important topic in modern networks. Use of the IP type of service (TOS) bits to classify traffic and implement different queuing strategies could be explored. These issues are becoming more important with the high cost of Internet bandwidth and the mixing of voice, video, and data traffic that place different demands on a network. For example, video and audio are very sensitive to jitter and insensitive to some packet loss, while data is usually unaffected by jitter but packet loss is intolerable. Newer router hardware that supports newer IOS software has many QoS features.
- IPSEC / VPNs / MPLS
IPSEC tunneling, virtual private networks, and multiprotocol label switching can also be explored. These techniques are used to build virtual private networks across public Internetworks and are very important topics. Access to these features again is restricted to newer routers that can run the latest IOS software.
- IP Multicast
IP Multicast is an important feature to distribute datastreams to multiple recipients. Protocols such as PIM, DVRMP, IGMP, and CGMP are supported in recent IOS software images. These topics can be explored without additional lab equipment.
- HSRP
The hot standby routing protocol enables two routers on a LAN segment to work together to provide a reliable virtual router. This is handy for hosts that do not understand routing protocols configured with a static default route pointing at the highly available virtual router. No additional lab hardware is required to experiment with this protocol.
- ISL Trunking
An important topic is to be able to use router in a “one armed router” or “router on a stick” configuration. A single router port capable of ISL trunking such as the fast ethernet port on r5 is connected to a switch and programmed to trunk. Many subinterfaces on the router can be created which can expand the number of logical ethernet ports on the router. Each physical port on the switch can be logically

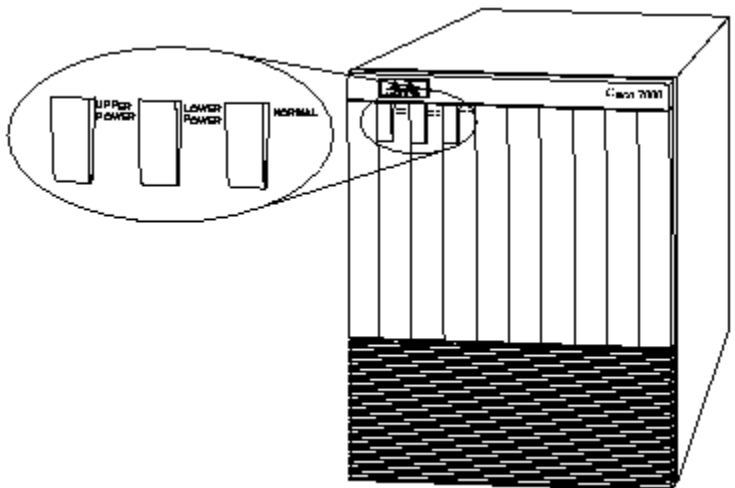
configured as a separate router interface on a different network. No additional lab equipment is required to experiment with this protocol, although only the single fast ethernet interface is capable of this feature.

Appendices

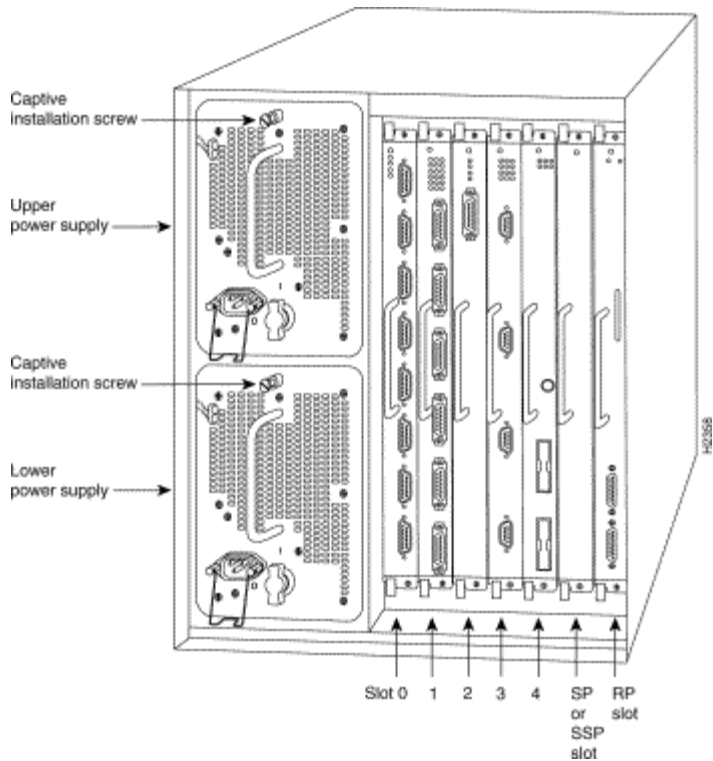
Appendix A: Router Hardware Overview

Cisco 7000 Core Router

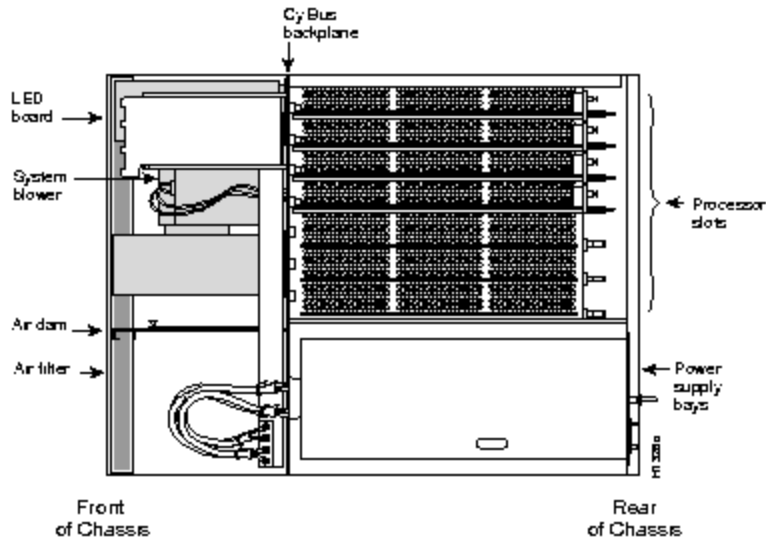
The Cisco 7000 is a core router designed for network backbone applications. It has dual power supplies for redundancy and 7 card slots for route processor, silicon switch processor and interface cards. The system backplane uses a “CX” bus. The route processor or “RP” and silicon switch processor or “SP” are required and contain the CPU, flash memory, DRAM memory, RS-232 console port, and switching hardware. This leaves 5 slots to accommodate “Interface Processor” cards. At the rear of the chassis from left to right, the slots are labeled “slot0”, “slot1”, “slot2”, “slot3”, “slot4”, “SP”, and “RP”. In our lab environment, the four 7000s have a FDDI card in slot 1, serial card in slot 2, and on R1 only an ethernet card in slot 2. The interface names in IOS depend on the slot containing the card. For example, an 8-port serial card in slot1 corresponds to interface names “serial1/0”, “serial1/1”, ... “serial1/7”. The same card in slot 4 would be labeled “serial4/0”, “serial4/1”, ... “serial4/7”. The 7000 chassis weighs 145 pounds when fully populated.



[Cisco 7000 Router, Front View]

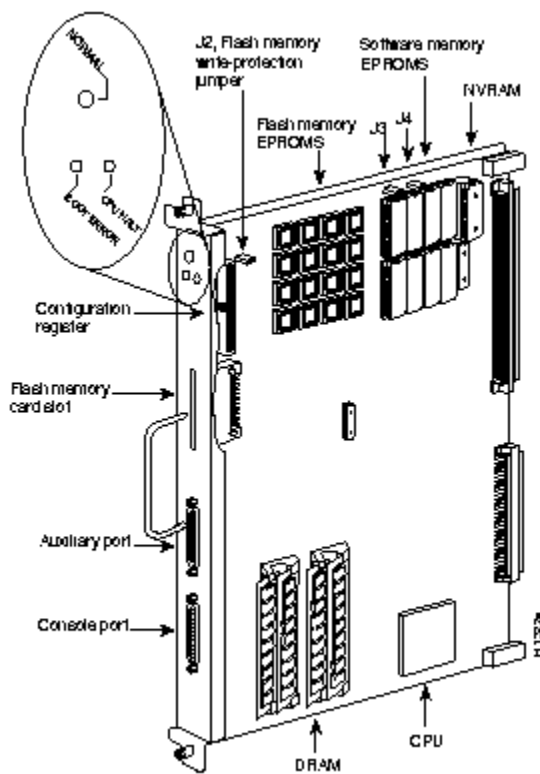


[Cisco 7000 Router, Rear View]



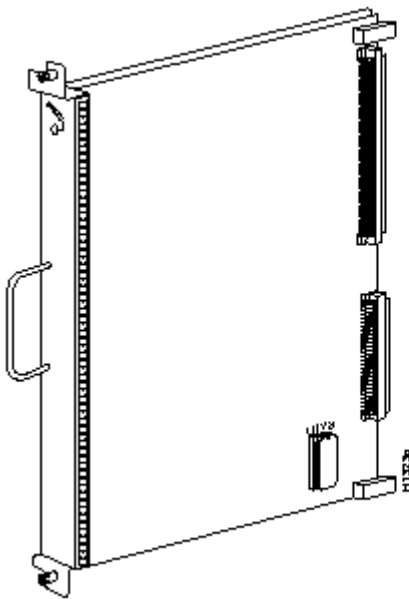
[Cisco 7000 Router, Top View w/cover removed]

The route processor is the brains of the router and contains the CPU, configuration register, boot ROMs, FLASH, DRAM, console port, auxiliary port, etc. The CPU is a Motorola 68040. Our systems are configured with 64M DRAM and 4M flash. There is also a special NVRAM memory device (Non-Volatile RAM) used to hold the configuration file. When the system boots, it executes code in the boot ROM similar to a PC BIOS. The system checks the configuration register to determine whether to boot into the ROM monitor, load an image from flash, boot from the network ,etc. Normally, the system loads an IOS (Internetwork Operating System) image from FLASH memory into DRAM and begins execution. Executing from DRAM requires additional memory but has a performance advantage since DRAM access times are faster than FLASH memory access times. The routers can also accommodate additional FLASH memory in the form of a PCMCIA FLASH card that can be used for storing IOS images or configuration files.



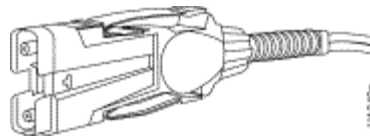
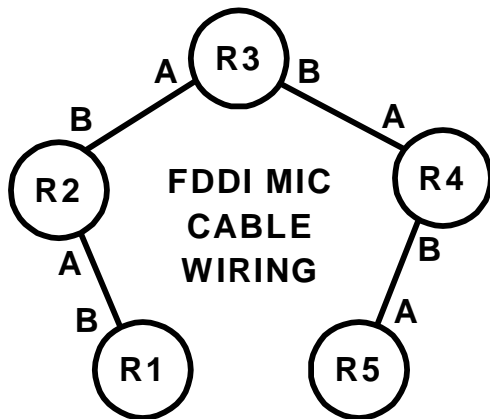
[RP Route Processor Card]

The silicon switch processor contains the switching hardware. The router has many switching modes or switching paths through the system. The most common are “processor switching,” “fast switching,” “CEF – Cisco Express Forwarding,” and “SSE – Silicon Switching Engine.” Processor switching uses the CPU to make forwarding decisions by looking at the routing table. Fast-switching and CEF use special forwarding tables when there is a software cache hit in interrupt mode to speed the switching of packets. SSE switching uses the SP card with dedicated switching hardware which is the fastest switching path. Initial packets require processor switching but subsequent packets can often use the SSE except under certain circumstances such as when access lists are applied to an interface. Use the IOS command “ip route-cache SSE” to enable the silicon switching path.



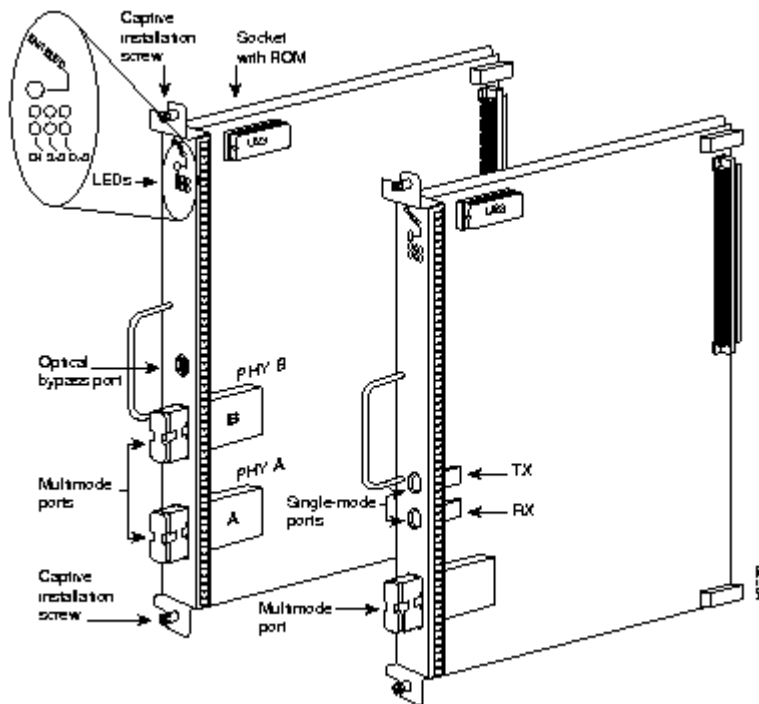
[SP Silicon Processor Card, also called Silicon Switching Engine (SSE)]

The CX-FIP or “CS-Bus FDDI Interface Processor” is a single port FDDI interface card. Our lab uses the type shown on the left with two multimode “MIC” FDDI connectors. It is a DAS (Dual-Attached Station) card with physical “A” and “B” ports. DAS devices are normally physically wired in a ring with a cable from the “A” port of router X to the “B” port of router X+1, where the last router’s “A” port connects to the first router’s “B” port. Each MIC connector has two singlemode 62.5/125µ fibers, to form two counter-routing rings. FDDI is a reliable 100Mbps backbone token-ring technology that can survive a break by going into a “WRAP” state. All four lab 7000 routers have DAS ports and the lab 4500 router has a SAS port. Because they are not all DAS, our lab network is normally in a “WRAP” state and does not make a complete ring and therefore will not sustain a cut.



[FDDI MIC Connector]

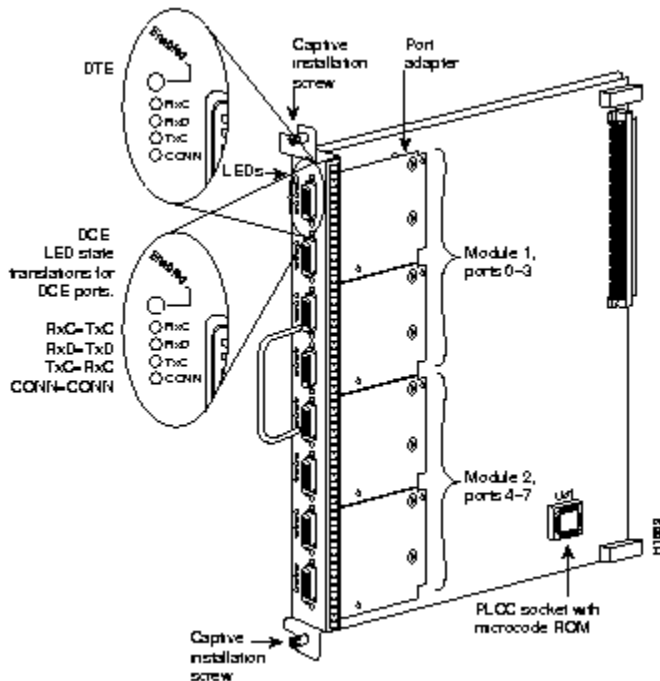
[Physical FDDI Wiring Diagram]



[CX-FIP 1-Port FDDI Mult multimode DAS card with MIC connectors (left)]

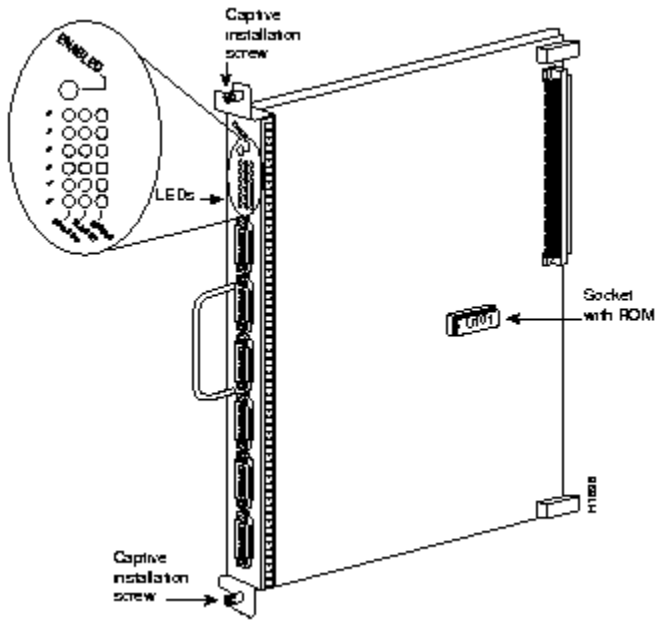
The CX-FSIP card or “CX-Bus Fast Serial Interface Processor” card contains eight serial connections on female DB60 connectors. Normally, the maximum speed of is 2Mbits/second but can be as high as 8Mbits/second under special circumstances. Normally, these ports are used to connect to T1 and E1 CSU/DSUs for connecting two routers through a telco circuit where the CSU/DSU provides the clocking for the port. In our lab environment, we are using special “back-to-back” serial cables to directly connect two router ports without any CSU/DSUs. Although both connectors are identical, one end of the cable is “DTE” or Data Terminal Equipment, while the other is “DCE” or Data Communications Equipment. The key difference is that the router port where the DCE end plus in must provide clocking which requires the use of the “clock rate” command. If you have a serial connection on your router and are unsure of whether the cable is DCE or DTE, you can use the command “show controller cbus” on 7000 routers (or “show controller serial” on 2500 routers) to identify the presence of the cable and cable type.

```
R2# show controller cbus
...
Interface 9 - Serial 1/1, electrical interface is V.35 DCE
...
Interface 10 - Serial 1/2, electrical int is Universal (cable unattached)
...
Interface 11 - Serial 1/3, electrical interface is V.35 DTE
...
```



[CX-FSIP 8-port Serial Card]

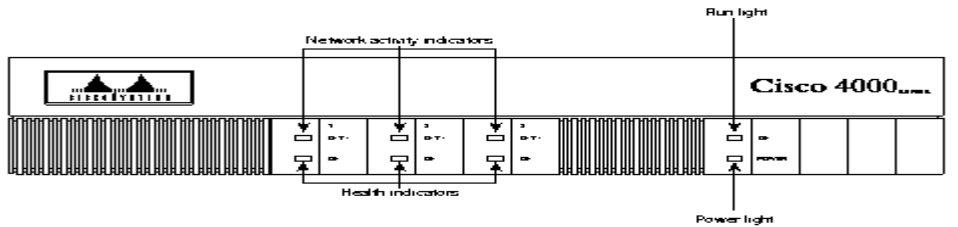
The CX-EIP card or “CX-Bus Ethernet Interface Processor” provides six 10Mbps ethernet ports using DB15F AUI connectors. These are the old style half-duplex ethernet and require an ethernet traneiver to adapt the port to the proper cabling scheme such as 10baseT or 10base2 “thinnet”. Our lab is using Allied Telesyn model AT210TS traneivers which adapt the ports to use 10baseT with RJ45 connectors. The traneivers also have handy status LEDs including a LINK LED that can be used to ascertain 10baseT LINK status.



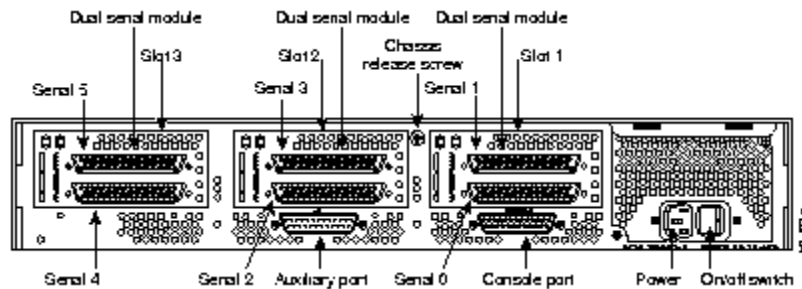
[CX-EIP 6-port 10Mbps Ethernet Card]

Cisco 4500 Mid-Size Router

The Cisco 4500 is a mid-size router with room to accommodate up to three interface “NP” modules. It utilizes a MIPS R4000 CPU (RISC) with internal NVRAM, BOOTFLASH, FLASH, shared DRAM and normal DRAM memory. It also has a console and auxiliary RS232 ports for out-of-band communication.

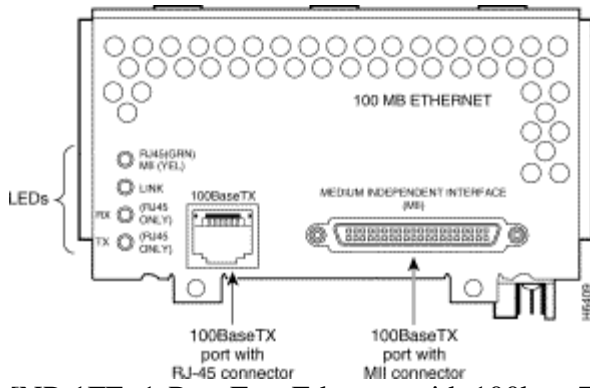


[Cisco 4500 Router, Front View]



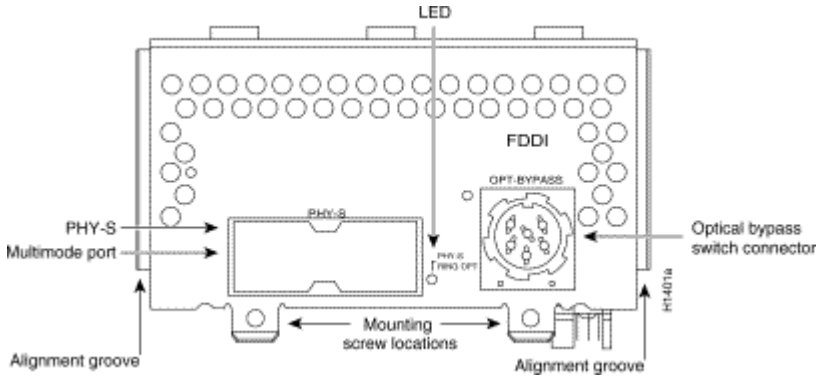
[Cisco 4500 Router, Rear View (Our 4500 has different interface modules than shown)]

The NP-1FE adapter provides a single 100Mbit/second Fast Ethernet interface labeled in the IOS software as “FastEthernet0”. There is both an RJ45 (100baseTX) connector and an MII (Media Independent Interface) connector. We will use the 100baseTX standard. The MII port accepts an adapter to allow connecting to other types of media such as multimode fiber to support the 100baseFX standard. A common configuration problem is to forget the program the IOS software to use the appropriate connector. The IOS command “media-type 100baseX” selects the R45 port, while “media-type MII” selects the MII port. This adapter can support full duplex operation. This adapter also supports the ISL (Inter Switch Link) trunking protocol to create VLAN subinterfaces.



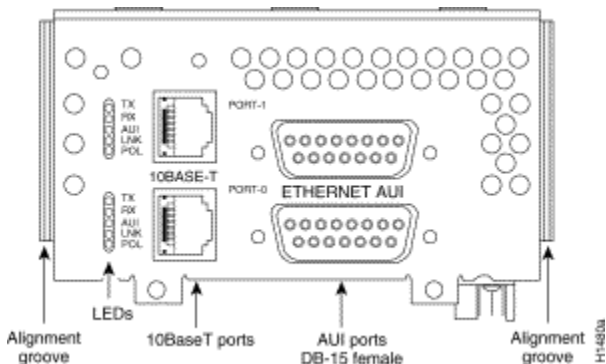
[NP-1FE 1-Port Fast Ethernet with 100baseTX and MII Ports]

The NP-1F-S-M adapter provides a single attached FDDI MIC interface using multimode fiber. Use a FDDI MIC/MIC cable to connect this device's physical "S" port to one of the 7000 DAS "B" ports.



[NP-1F-S-M 1-Port FDDI Multimode SAS with MIC connector]

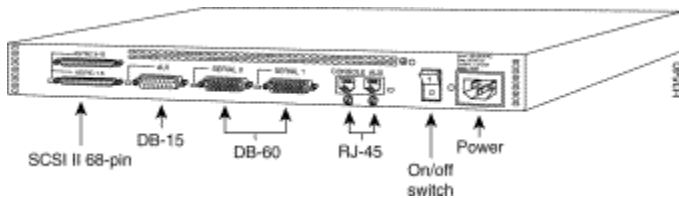
The NP-2E adapter provides two legacy 10Mbps/second half-duplex ethernet ports. It provides both an RJ45 (10baseT) and AUI interfaces. Under IOS, these interfaces are labeled as "Ethernet0" (bottom connector) and "Ethernet1" (top connector). The AUI interface is used with a transceiver to adapt to different ethernet media types such as 10base2. In our lab, we will be using the RJ45 10baseT port. A common configuration problem is to forget to specify which connector you are using under IOS. Use the command "media-type 10baseT" to select the RJ45 connector, or "media-type AUI" to select the AUI port.



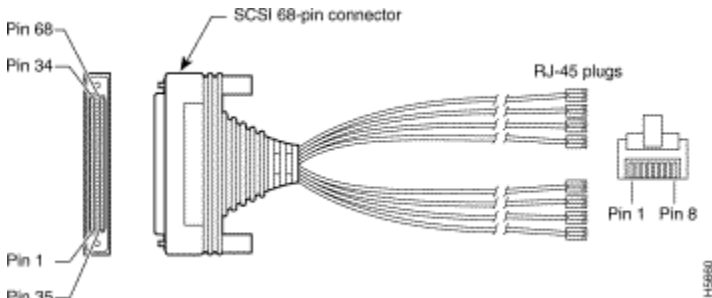
[NP-2E 2-Port Ethernet with both 10baseT (RJ45) and AUI (DB15S) Connectors]

Cisco 2511 Access Server / Router

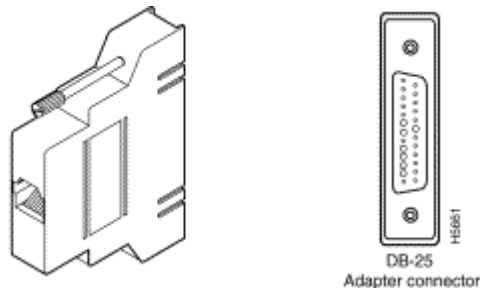
The Cisco 2511 is a small, non-expandable router. It utilizes a Motorola 68030 CPU with internal NVRAM, ROM, FLASH, and DRAM memory. It has two high speed serial ports suitable for speeds up to 2Mbits/second, a 10Mbit/second ethernet AUI port, and 16 asynchronous RS-232 ports. Two 68-pin SCSI style connectors provide 8 asynchronous ports each and use an octopus breakout cable (p/n CAB-OCTAL-ASYNC) to break into individual ports. CAB-25AS-MMOD adapters adapt the octal cable to DB25M connectors which attach to the 7000 and 4500 DB25F console ports. This router was designed to provide a small platform to support up to 16 analog dialup modems, but in our lab environment we will be using a feature called “inverse telnet.” This feature allows us to connect to the router with a TELNET session and establish an RS232 terminal session with one of the async lines. These async lines are programmed for 9600 baud and connect to individual router console ports. This provides out-of-band access to program the lab routers, even when they are not in a working configuration.



[Cisco 2511 Access Server / Router]



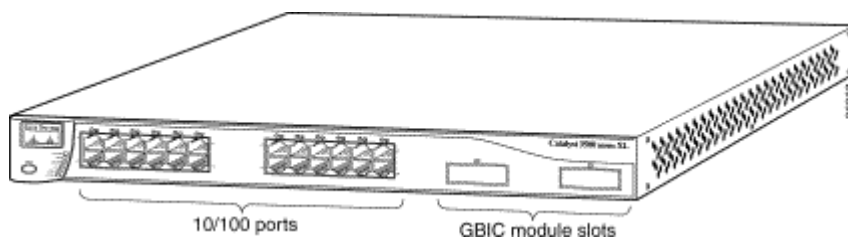
[Octopus Cable CAB-OCTAL-ASYNC]



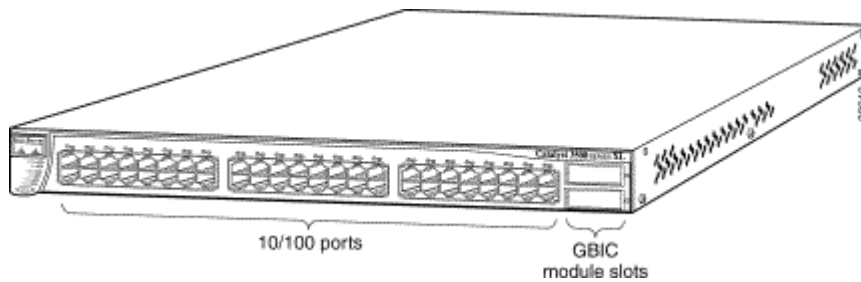
[RJ45S-DB25M Adapter CAB-25AS-MMOD]

Cisco 3548XL and 3524XL Ethernet Switches

A Cisco 3524XL ethernet switch is used to connect the lab router ethernet and fast ethernet ports. This switch uses a 1000baseSX GBIC adapter across multimode fiber cable to physically connect to two 3548XL ethernet switches that provide 96 10/100 ethernet ports to connect to student PCs in the networking lab. The switches use the ISL trunking protocols to implement VLANs that span the three switching allowing the switches to be configured to group the router ethernet ports and student computer ports in any desired configuration. This provides a lot of flexibility for building different labs. For ease of programming, the console port of the switch connects to an async line on the 2511 to provide out-of-band access for configuration.



[Cisco 3524XL Ethernet Switch with 24 10/100 ethernet ports + 2 GigE ports]



[Cisco 3548XL Ethernet Switch with 48 10/100 ethernet ports + 2 GigE ports]

Appendix B: Router IOS Software

The ITL lab Cisco routers use the Internetwork Operating System (IOS) software. The IOS software is typically stored as a compressed binary image in flash. Lower end platforms (25xx) execute code directly from flash while higher end platforms (45xx,7xxx) copy the code from FLASH to DRAM to take advantage of faster DRAM memory access times. Most of IOS is written in the C Programming Language and cross-compiled with the GNU C compiler for each router architecture. Software is distributed as binary images, usually through downloading from a password protected area on the Cisco web site. There are many different versions indicated by a major version number, minor version number, release level, and optionally “train”. Within a given version, there are “feature sets” which generally determine which protocols are supported. Sometimes features available only in the enterprise feature set are incorporated into the baseline IP feature set in subsequent versions such as network address translation (NAT). There are about twenty different feature sets, but the most important are as follows:

- IP (IP Protocol and Bridging Only)
- DESKTOP (adds support for IPX, Appletalk, and DECnet)
- ENTERPRISE (adds support for Apollo,Banyan,ISO CLNS,XNS,etc.)

Software versions have minimum DRAM and FLASH memory requirements. On the Cisco ITL routers, we have chosen to use the most stable version of IOS software with the largest feature sets that will fit in available memory to maximize flexibility as follows:

	MODEL	DRAM	FLASH	VER	FEATURE	IMAGE
R1	7000	64M	4M	11.1(24)	ENTERPRISE	gs7-j-mz.111-24.bin
R2	7000	64M	4M	11.1(24)	ENTERPRISE	gs7-j-mz.111-24.bin
R3	7000	64M	4M	11.1(24)	ENTERPRISE	gs7-j-mz.111-24.bin
R4	7000	64M	4M	11.1(24)	ENTERPRISE	gs7-j-mz.111-24.bin
R5	4500	48M	8M	12.0(13)	ENTERPRISE + IPSEC	c4500-js56i-mz.120-13.bin
R6	2511	4M	8M	12.0(13)	DESKTOP	c2500-d-l.120-13.bin

As of this writing, the most current IOS software version is 12.1(5) which is supported on the 4500 and 2511 platforms but would require additional memory. The Cisco 7000 has reached its end of life and the most recent software version supported is 11.2(24). Running the enterprise 11.2(24) software is possible on the lab 7000s but would require a 8M PCMCIA flash card and BIOS BOOT ROM upgrade as the current v10.0(7) BOOT ROMs do not understand PCMCIA flash cards and the image will not fit in the internal 4M flash memory.

In summary, the most important protocols like IP, IPX, and Appletalk are present on all routers. Network Address Translation (NAT) which was not incorporated until version 11.3 is only present on routers R5 and R6.

Appendix C: IOS Software Documentation

The Cisco IOS documentation is available in three forms – (1) world-wide-web, (2) CD-ROM, and (3) hardcopy manuals.

- **WORLD-WIDE-WEB**
The documentation on the Cisco web page does not require any special accounts or passwords. The URL is <http://www.cisco.com>. From the home page, go to Technical Documents → Documentation Home Page → Cisco IOS Software Configuration. From this point, choose the appropriate software version. The documents are available in both hypertext (for viewing) and PDF (for printing).
- **CD-ROM**
The same documentation is available on a single CD-ROM which is distributed with new router equipment. It requires a Microsoft Windows 95/98/NT/2000 PC and contains the manuals in hypertext format and includes a search engine. This is a handy form when your network is broken or you do not have access to the Internet.
- **HARDCOPY**
The manuals are also available as a set of hard copy volumes. Two small volumes have an index of the command reference volumes and configuration guide volumes. The volumes are 8.5”x11”. As of version 12.1, the full set requires approximately 5 linear feet of shelf space.

Since IOS v11.1, many new features have been added and the number of manual pages has increased around five-fold as of version 12.1. Most of the commands in the earlier versions will still work with newer software although occasionally some of the default behaviors have changed. When studying the core IP routing protocols, the v11.1 manuals are probably the best source of information as much of the extraneous new features are not present. The IOS v11.1 manuals are organized as follows:

- Configuration Fundamentals
 - o User Interface
 - o Configuration Files
- Access Services
 - o Terminal Lines
 - o PPP/SLIP
 - o Telnet
- Wide Area Networks
 - o ATM
 - o Frame-Relay
 - o ISDN
 - o X.25
- Network Protocols, Part 1

- Appletalk
- IP
- IPX
- IP Routing Protocols
 - RIP
 - OSPF
 - IGRP
 - EIGRP
 - BGP
 - IS-IS
- Network Protocols, Part 2
 - Apollo Domain
 - Banyan Vines
 - DECnet
 - ISO CLNS
 - Xerox XNS
- Bridging and IBM Networking
 - Transparent Bridging
 - Source-Route Bridging
 - DLSW

For each topic, there are configuration guides and command references. The configuration guides address groups of related commands, explain more of the theory, and have more complex examples. The command references are generally alphabetized listings of configuration commands that detail the command syntax

Appendix D: Cisco Router Password Recovery Procedure

On occasion, the router password may be forgotten and need to be recovered. The following procedure may be used to recover from a situation where the password is lost provided you have physical access to the router.

Cisco routers use a 16 bit configuration register to control how the system will boot and are normally set to the value 0x2102. Bit 6 of this register controls whether the router will load the startup configuration upon booting (bit 6 is clear), or simply start with an empty configuration (bit 6 is set). The basic idea is to power cycle the router with a dumb terminal or emulator attached to the console port. Within the first 60 seconds of booting, send a BREAK signal to the router to make it stop the boot process. You then change the configuration register from the default value of 0x2102 to 0x2142, and reboot the system. You will often get a configuration dialog when the system reboots where you simply press control-C to abort the dialog. You will now be at the command prompt "Router>" where you type the "enable" command where the prompt will change to "Router#" without prompting for a password. At this point, you copy the startup configuration into the running configuration, however, all interfaces are shut down and must be manually enabled. Now you can change the console, vty, and enable passwords. You must then copy the new running configuration to the startup configuration, change the configuration register back to 0x2102, then reboot. The procedure is slightly different among the different router platforms and is detailed below.

1. Physically connect a dumb terminal or PC with a terminal emulator such as HyperTerm to the router console port.
2. Configure your terminal or emulator for the following settings:
 - a. 9600 baud rate
 - b. no parity
 - c. 8 data bits
 - d. 1 stop bit
 - e. no flow control
3. Power cycle the router to make it reboot and send a BREAK signal from the terminal within the first 60 seconds to stop the boot process and enter the ROM monitor. (Control-BREAK in HyperTerm).
4. Change the configuration register from the default value 0x2102 to 0x2142 so the router will ignore the stored configuration with unknown passwords upon reboot. Afterwards, reboot the router.

Cisco 7000 Routers:

```
System Bootstrap, Version 5.0(7), RELEASE SOFTWARE  
Copyright (c) 1986-1994 by cisco Systems  
RPl processor with 65536 Kbytes of main memory
```

```
F3: 3559856+114640+289292 at 0x1000
Abort at 0x2C8AC (PC)
>o/r 0x2142
>i
```

Cisco 4500 Routers:

```
System Bootstrap, Version 5.1(1) [daveu1], RELEASE SOFTWARE
Copyright (c) 1994 by cisco Systems, Inc.
monitor: command "cisco2-C4500" aborted due to user interrupt
rommon 1 > confreg 0x2142
You must reset or power cycle for new config to take effect
rommon 2 > reset
```

Cisco 2500 Routers:

```
System Bootstrap, Version 5.2(8a), RELEASE SOFTWARE
Copyright (c) 1986-1995 by cisco Systems
2500 processor with 2048 Kbytes of main memory
Abort at 0x10EA87C (PC)
>o/r 0x2142
>i
```

5. Upon rebooting, the router will normally start a dialog asking if you want it to help you configure the system. Abort this dialog by typing control-C (^C).

```
Would you like to enter the initial config dialog? [yes]: ^C
```

6. Wait a moment and hit return a few times. You should get the non-enabled router prompt "Router>". Type "enable" <cr> and the prompt should change to "Router#".

```
Router> enable
```

7. Copy the startup configuration to the running configuration as your starting point.

```
Router# copy startup-config running-config
```

8. Note which interfaces are present as they will be administratively shut down and will need to be manually enabled.

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

9. Go into configuration mode and enable each interface as appropriate.

```
Router# config term
Router(config)#int e0
Router(config-if)#no shutdown
Router(config)#int fddi0
Router(config-if)#no shutdown
```

10. Set the enable, console, and vty password. If you are using the "enable secret" mechanism, set it as well. Here we will use the password "cisco".

```
Router(config-if)#enable password cisco {weak encryption}
Router(config-if)#enable secret cisco {strong encryption}
Router(config)#line con 0
Router(config-line)#password cisco
```

```
Router(config-line)#line vty 0 4  
Router(config-line)#password cisco  
. . .
```

11. Change the configuration register back to the default value 0x2102 to take effect upon the next reboot.

```
Router(config-line)# config-reg 0x2102
```

12. Get out of the configuration mode by pressing control-Z (^Z) and save your changes by copying them to the startup configuration.

```
Router(config)# ^Z  
Router# copy running-config startup-config
```

13. Reboot the router

```
Router# reload  
Proceed with reload? [confirm] y
```

14. After rebooting, the router will accept the new password.

Appendix E: Cisco 2511 Firewall Router Configuration

```
version 12.0
service timestamps debug datetime localtime show-timezone
service timestamps log datetime localtime show-timezone
service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname fw/r6
!
enable secret level 2 5 $1$wLUZ$$SjI2kZwadyltWehijBOvj0
enable secret 5 $1$I9LE$c5KYutJeq/gmRcYNb4z3B0
!
ip subnet-zero
ip host r1 2001 128.186.121.88
ip host r2 2002 128.186.121.88
ip host r3 2003 128.186.121.88
ip host r4 2004 128.186.121.88
ip host r5 2005 128.186.121.88
ip host cat1 2007 128.186.121.88
ip host s1 2008 128.186.121.88
ip name-server 128.186.121.10
clock timezone EST -5
clock summer-time EDT recurring
!
!
interface Loopback0
 ip address 192.168.66.6 255.255.255.0
 no ip directed-broadcast
 ip nat inside
!
interface Ethernet0
 ip address 128.186.121.88 255.255.255.0
 no ip directed-broadcast
 ip nat outside
!
interface Serial0
 description Link to R1 S1/6
 ip address 192.168.16.6 255.255.255.0
 no ip directed-broadcast
 ip nat inside
 clockrate 2000000
!
interface Serial1
 description Link to R3 S1/6
 bandwidth 2000
 ip address 192.168.36.6 255.255.255.0
 no ip directed-broadcast
 ip nat inside
 clockrate 2000000
!
router eigrp 100
 redistribute static metric 10000 100 255 128 1500
 network 192.168.16.0
 network 192.168.36.0
 network 192.168.66.0
!
router ospf 100
 network 192.168.16.0 0.0.0.255 area 0
 network 192.168.36.0 0.0.0.255 area 0
```

```

    default-information originate always
!
router rip
  network 192.168.16.0
  network 192.168.36.0
  network 192.168.66.0
  default-metric 5
!
router igrp 100
  network 192.168.16.0
  network 192.168.36.0
  network 192.168.66.0
!
ip nat inside source list rfc1918 interface Ethernet0 overload
ip nat inside source static 192.168.10.2 128.186.121.90
ip nat inside source static 192.168.10.3 128.186.121.89
ip http server
ip classless
ip route 0.0.0.0 0.0.0.0 128.186.121.1
!
ip access-list standard rfc1918
  permit 192.168.0.0 0.0.255.255
  permit 172.16.0.0 0.15.255.255
  permit 10.0.0.0 0.255.255.255
  deny any
!
ip access-list extended fw-in
  permit tcp any any established
  permit ip host 128.186.121.41 any
  permit ip host 128.186.121.10 any
  permit ip host 128.186.2.206 any
  deny tcp any any eq telnet
  deny tcp any host 128.186.121.88 range 2001 2017
  permit udp any any eq ntp
  permit udp any any eq snmp
  permit udp any any eq 22
  permit udp any eq ntp any
  permit udp any eq snmp any
  permit udp any eq 22 any
  deny tcp any any
  deny udp any any
  permit icmp any any
snmp-server community public RO
privilege router level 15 network
privilege interface level 15 ip address
privilege interface level 15 ip
privilege configure level 15 interface
privilege exec level 2 more
privilege exec level 2 traceroute
privilege exec level 2 ping
privilege exec level 2 show running-config
privilege exec level 2 show configuration
privilege exec level 2 show
privilege exec level 2 clear line
privilege exec level 2 clear counters
privilege exec level 2 clear
!
line con 0
  password 7 030752180500
  login
  transport input none
line 1 16
  no exec

```

```
modem InOut
transport input all
transport output none
line aux 0
password 7 09495E0410091201
line vty 0 4
exec-timeout 60 0
password 7 104D000A0618
login
!
ntp server 128.186.121.10
end
```


Appendix F: Baseline Router Configuration

In order to quickly get the student ITL routers r1, r2, r3, r4, and r5 working with a rudimentary RIP setup, baseline configurations have been written. Since routers r1 through r5 each had a small amount of free space on their internal flash memory used to hold the Cisco IOS software, a small baseline configuration file has been saved. This may be a bit more convenient than using a terminal emulator copy and paste capabilities. On each router r1 through r5, two steps are required:

1. Copy the appropriate baseline configuration file in flash to the startup-configuration non-volatile memory using the command (Replace X with the integer router identifier):

```
RouterX# copy flash:base-rX.cfg startup-config
```

2. Reboot the router using the “reload” command:

```
RouterX# reload
```

```
r5#dir flash:
Directory of flash:/

   1  -rw-        6558840          <no date>  c4500-js56i-mz.120-13.bin
   2  -rw-         1148          <no date>  base-r5.cfg

8388608 bytes total (1828492 bytes free)
r5#copy flash:base-r5.cfg startup-config
Destination filename [startup-config]?
[OK]
1148 bytes copied in 0.156 secs
r5#reload
Proceed with reload? [confirm]y
00:02:11: %SYS-5-RELOAD: Reload requested
System Bootstrap, Version 5.1(1) [daveu 1], RELEASE SOFTWARE (fc1)
Copyright (c) 1994 by cisco Systems, Inc.
C4500 processor with 32768 Kbytes of main memory

Self decompressing the image : ####[OK]

Cisco Internetwork Operating System Software
IOS (tm) 4500 Software (C4500-JS56I-M), Version 12.0(13), RELEASE
SOFTWARE (fc1)
...
cisco 4500 (R4K) processor (revision 0x00) with 32768K/16384K bytes of
memory.
...
Press RETURN to get started!
r5>enable
Password:
r5#
```

Alternatively, you can erase the startup configuration, reboot the router, skip the initial configuration dialog, enter privileged mode, enter configuration mode, and paste the appropriate configuration commands.

1. Erase the startup configuration using the “write erase” command.
2. Reboot the router using the “reload” command.
3. After reboot, when prompted by the initial configuration dialog, exit with control-C.
4. Enter the enable mode with the command “enable”.
5. Enter the configuration mode with the command “config term”
6. Copy the appropriate text configuration into the clipboard. (In notepad, wordpad, or Microsoft Word, highlight the configuration text and select Edit→Copy.)
7. Inside your terminal emulator, paste the clipboard contents to the router. (In the MS-Windows TELNET program, use Edit→Paste.)
8. Exit the configuration mode with control-Z.
9. Issue the “write” command to save your changes to non-volatile memory.

```
r5#write erase
Erasing nvram filesystem will remove all files! Continue? [confirm]y[OK]
Erase of nvram: complete
r5#
r5#reload
Proceed with reload? [confirm]y
00:17:06: %SYS-5-RELOAD: Reload requested
System Bootstrap, Version 5.1(1) [daveu 1], RELEASE SOFTWARE (fc1)

      --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]^C
...
Press RETURN to get started!
...
Router>
Router>enable
Router#config term
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
(Paste the configuration text here)
Router(config)#^Z
Router#
00:02:40: %SYS-5-CONFIG_I: Configured from console by console
Router#write
Building configuration...
[OK]
Router#
```

The following is the baseline router configuration. See also the “baseline” subdirectory on the accompanying project CD-ROM.

COMMON:

```
service udp-small-servers
service tcp-small-servers
enable password cisco
no ip domain-lookup
no ip classless
logging buffered
snmp-server community public RO
line con 0
  exec-timeout 0 0
line aux 0
line vty 0 4
  password cisco
  login
```

R1:

```
hostname r1
interface Loopback0
  ip address 192.168.11.1 255.255.255.0
  no shutdown
interface Fddi0/0
  ip address 192.168.1.1 255.255.255.0
  no shutdown
interface Serial1/2
  description Link to R2 S1/1
  ip address 192.168.12.1 255.255.255.0
  bandwidth 2000
  no shutdown
interface Serial1/3
  description Link to R3 S1/1
  ip address 192.168.13.1 255.255.255.0
  bandwidth 2000
  no shutdown
interface Serial1/4
  description Link to R4 S1/1
  ip address 192.168.14.1 255.255.255.0
  bandwidth 2000
  no shutdown
interface Serial1/6
  description Link to R6 S0
  ip address 192.168.16.1 255.255.255.0
  bandwidth 2000
  no shutdown
interface E2/0
  description Vlan 10 to cat1 FA0/1
  ip address 192.168.10.1 255.255.255.0
  no shutdown
interface E2/1
  description Vlan 20 to cat1 FA0/2
  ip address 192.168.20.1 255.255.255.0
  no shutdown
interface E2/2
  description Vlan 30 to cat1 FA0/3
  ip address 192.168.30.1 255.255.255.0
  no shutdown
interface E2/3
  description Vlan 40 to cat1 FA0/4
  ip address 192.168.40.1 255.255.255.0
  no shutdown
interface E2/4
  description Vlan 50 to cat1 FA0/5
  ip address 192.168.50.1 255.255.255.0
  no shutdown
interface E2/5
```

```
  description Vlan 60 to cat1 FA0/6
  ip address 192.168.60.1 255.255.255.0
  no shutdown
router rip
  network 192.168.11.0
  network 192.168.12.0
  network 192.168.13.0
  network 192.168.14.0
  network 192.168.16.0
  network 192.168.1.0
  network 192.168.10.0
  network 192.168.20.0
  network 192.168.30.0
  network 192.168.40.0
  network 192.168.50.0
  network 192.168.60.0
```

R2:

```
hostname r2
interface Loopback0
  ip address 192.168.22.2 255.255.255.0
  no shutdown
interface Fddi0/0
  ip address 192.168.1.2 255.255.255.0
  no shutdown
interface Serial1/1
  description Link to R1 S1/2
  ip address 192.168.12.2 255.255.255.0
  bandwidth 2000
  clockrate 2000000
  no shutdown
interface Serial1/3
  description Link to R3 S1/2
  ip address 192.168.23.2 255.255.255.0
  bandwidth 2000
  no shutdown
interface Serial1/4
  description Link to R4 S1/2
  ip address 192.168.24.2 255.255.255.0
  bandwidth 2000
  no shutdown
router rip
  network 192.168.12.0
  network 192.168.22.0
  network 192.168.23.0
  network 192.168.24.0
  network 192.168.1.0
```

R3:

```
hostname r3
interface Loopback0
  ip address 192.168.33.3 255.255.255.0
  no shutdown
interface Fddi0/0
  ip address 192.168.1.3 255.255.255.0
  no shutdown
interface Serial1/0
  description Link to self
  no ip address
  bandwidth 2000
  no shutdown
interface Serial1/1
  description Link to R1 S1/3
```

```

ip address 192.168.13.3 255.255.255.0
bandwidth 2000
clockrate 2000000
no shutdown
interface Serial1/2
description Link to R2 S1/3
ip address 192.168.23.3 255.255.255.0
bandwidth 2000
clockrate 2000000
no shutdown
interface Serial1/3
description Link to self
no ip address
bandwidth 2000
clockrate 2000000
no shutdown
interface Serial1/4
description Link to R4 S1/3
ip address 192.168.34.3 255.255.255.0
bandwidth 2000
no shutdown
interface Serial1/6
description Link to R6 S1
ip address 192.168.36.3 255.255.255.0
bandwidth 2000
no shutdown
router rip
network 192.168.33.0
network 192.168.13.0
network 192.168.23.0
network 192.168.34.0
network 192.168.36.0
network 192.168.1.0

```

R4:

```

hostname r4
interface Loopback0
ip address 192.168.44.4 255.255.255.0
no shutdown
interface Fddi0/0
description Link to R5 FDDI0
ip address 192.168.1.4 255.255.255.0
no shutdown
interface Serial1/1
description Link to R1 S1/4
ip address 192.168.14.4 255.255.255.0
bandwidth 2000
clockrate 2000000
no shutdown
interface Serial1/2

```

```

description Link to R2 S1/4
ip address 192.168.24.4 255.255.255.0
bandwidth 2000
clockrate 2000000
no shutdown
interface Serial1/3
description Link to R3 S1/4
ip address 192.168.34.4 255.255.255.0
bandwidth 2000
clockrate 2000000
no shutdown
router rip
network 192.168.44.0
network 192.168.14.0
network 192.168.24.0
network 192.168.34.0
network 192.168.1.0

```

R5:

```

hostname r5
interface loopback0
ip address 192.168.55.5 255.255.255.0
no shutdown
interface FastEthernet0
description Vlan70 to cat1 FA0/7
ip address 192.168.70.1 255.255.255.0
media-type 100BaseX
no shutdown
interface Ethernet0
description Vlan80 to cat1 FA0/8
ip address 192.168.80.1 255.255.255.0
media-type 10BaseT
no shutdown
interface Ethernet1
description Vlan90 to cat1 FA0/9
ip address 192.168.90.1 255.255.255.0
media-type 10BaseT
no shutdown
interface Fddi0
description Link to R4 FDDI0/0
ip address 192.168.1.5 255.255.255.0
no keepalive
no shutdown
router rip
network 192.168.55.0
network 192.168.70.0
network 192.168.80.0
network 192.168.90.0
network 192.168.1.0

```

Appendix G: Linux Scripts

There an *expect* script available to execute commands on routers, *cisexec*, and another more dangerous script, *ciscfg*, that can automatically modify a router's configuration. These two scripts were downloaded from the Chesapeake Computer Consultants web site <http://www.ccci.com> and were modified slightly to work under Linux. A small PERL script, *cis2inv.pl*, was also written to take output from one of the *expect* scripts to produce a comma delimited file suitable for Microsoft Excel import with IOS and memory information on the router to produce tables like that shown below.

ROUTER	CPU	IOS VER	IOS IMAGE	BOOT ROM	FLASH	DRAM
192.168.11.1	RP1 (68040)	11.1(24)	gs7-j-mz.111-24.bin	5.0(7)	4096	65536
192.168.22.2	RP1 (68040)	11.1(24)	gs7-j-mz.111-24.bin	5.0(7)	4096	65536
192.168.33.3	RP1 (68040)	11.1(24)	gs7-j-mz.111-24.bin	5.0(5)	4096	65536
192.168.44.4	RP1 (68040)	11.1(24)	gs7-j-mz.111-24.bin	5.2(9)	4096	65536
192.168.55.5	4500 (R4K)	12.0(13)	flash:c4500-js56i-mz.120-13.bin	5.1(1)	4096	49152
192.168.66.6	2511 (68030)	12.0(13)	flash:c2500-d-l.120-13.bin	5.2(8a)	8192	4096

To use these scripts, you must create a text file with the router passwords. For example, if the login password is "cisco" and enable password also "cisco", a text file like this is required. The first column specifies the router name which can include the wildcard "*".

```
[curci@sl cisinfo]$ cat key.dat
*      cisco  cisco
```

Before using the *expect* scripts, you must set an environment variable named "CISCOPASSWORDS" to point to the file with the passwords.

```
[Borne Shell Example]
Linux$ CISCOPASSWORDS=key.dat
Linux$ export CISCOPASSWORDS

[C-Shell Example]
Linux$ setenv CISCOPASSWORDS key.dat
```

It is also convenient to have a text file with a list of router IP addresses that can be given as an argument.

```
[curci@sl cisinfo]$ cat routers
192.168.11.1
192.168.22.2
192.168.33.3
192.168.44.4
192.168.55.5
```

The *cisexec* script simply uses TELNET to log into each router specified and executes a list of commands given as arguments. Here is an example where I use *cisexec* to display the clock on routers r1 through r5. The first try is unsuccessful, because I forget to define the environment variable CISCOPASSWORDS, while the second try works.

```
[curci@s1 cisinfo]$ ./cisexec "show clock" ./routers
Environment variable CISCOPASSWORDS not set, exiting...
```

```
[curci@s1 cisinfo]$ CISCOPASSWORDS=key.dat
[curci@s1 cisinfo]$ export CISCOPASSWORDS
[curci@s1 cisinfo]$ ./cisexec "show clock" ./routers
```

```
spawn telnet 192.168.11.1
Trying 192.168.11.1...
Connected to 192.168.11.1.
User Access Verification
Password:
r1>terminal length 0
r1> show clock
12:52:56.077 EST Fri Nov 24 2000
r1>
```

```
spawn telnet 192.168.22.2
Trying 192.168.22.2...
Connected to 192.168.22.2.
User Access Verification
Password:
r2>terminal length 0
r2> show clock
*17:52:53.474 UTC Fri Nov 24 2000
r2>
```

```
spawn telnet 192.168.33.3
Trying 192.168.33.3...
Connected to 192.168.33.3.
User Access Verification
Password:
r3>terminal length 0
r3> show clock
*17:53:01.310 UTC Fri Nov 24 2000
r3>
```

```
spawn telnet 192.168.44.4
Trying 192.168.44.4...
Connected to 192.168.44.4.
User Access Verification
Password:
r4>terminal length 0
r4> show clock
*17:52:44.474 UTC Fri Nov 24 2000
r4>
```

```
spawn telnet 192.168.55.5
Trying 192.168.55.5...
Connected to 192.168.55.5.
User Access Verification
Password:
r5>terminal length 0
r5> show clock
*17:53:02.346 UTC Fri Nov 24 2000
r5>
```

To execute a privileged command, you must use the '-p' option which forces the script to first change the router to enabled mode before executing the commands. It is also

possible to execute multiple commands separated with a semi-colon. Instead of using a filename containing IP addresses of the routers, you can list the routers individually.

```
Linux$ ./cisexec -p "sh run;sh ver;sh ip ro" 192.168.11.1 192.168.22.2
```

With the *cisexec* utility, several extraneous lines, blank lines, and carriage return characters get logged in the capture. I have also written a small wrapper script to remove these.

```
Linux$ cat go
#! /bin/sh
#
# 20-Nov-2000 R.Curci
# Borne shell script to execute ./cisexec but remove extraneous
# parts of the captured session, consecutive blank lines,
# and carriage return '\r' (ASCII 13 decimal) characters.
#
CISCPASSWORDS=key.dat
export CISCPASSWORDS
#
for x in 1 2 3 4 5
do
    h=192.168.$x$x.$x
    ./cisexec -p "$1" $h
done |
egrep -v '(^spawn|^Trying|^Connect|^User Acc|>enable)' |
egrep -v '(terminal length 0|^Password:|^Escape character)' |
tr -d '\r' | uniq
```

Here is a sample execution where we execute the “show cdp neighbor” command to see which devices are adjacent using the data link Cisco Discovery Protocol (CDP).

```
[curci@s1 cisinfo]$ ./go "show cdp neighbor"
r1# show 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
r2                  Ser 1/2          122        R           RP1       Ser 1/1
r2                  Fddi0/0          122        R           RP1       Fddi0/0
r3                  Ser 1/3          135        R           RP1       Ser 1/1
r3                  Fddi0/0          135        R           RP1       Fddi0/0
r4                  Ser 1/4          134        R           RP1       Ser 1/1
r4                  Fddi0/0          134        R           RP1       Fddi0/0
cat1                Eth 2/5          162        T S         WS-C3524-XFas 0/6
cat1                Eth 2/4          162        T S         WS-C3524-XFas 0/5
cat1                Eth 2/3          162        T S         WS-C3524-XFas 0/4
cat1                Eth 2/2          162        T S         WS-C3524-XFas 0/3
cat1                Eth 2/1          162        T S         WS-C3524-XFas 0/2
cat1                Eth 2/0          162        T S         WS-C3524-XFas 0/1
r5                  Fddi0/0          121        R           4500      Fddi0
fw/r6               Ser 1/6          144        R           2511      Ser 0
r1#

r2# show 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
r3             Ser 1/3       131      R           RP1       Ser 1/2
r3             Fddi0/0      131      R           RP1       Fddi0/0
r1             Fddi0/0      139      R           RP1       Fddi0/0
r1             Ser 1/1       139      R           RP1       Ser 1/2
r4             Ser 1/4       130      R           RP1       Ser 1/2
r4             Fddi0/0      130      R           RP1       Fddi0/0
r5             Fddi0/0      177      R           4500     Fddi0
r2#

```

```
r3# show 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
r2             Ser 1/2       174      R           RP1       Ser 1/3
r2             Fddi0/0      173      R           RP1       Fddi0/0
r3             Ser 1/3       127      R           RP1       Ser 1/0
r3             Ser 1/0       127      R           RP1       Ser 1/3
r1             Fddi0/0      135      R           RP1       Fddi0/0
r1             Ser 1/1       135      R           RP1       Ser 1/3
r4             Ser 1/4       126      R           RP1       Ser 1/3
r4             Fddi0/0      125      R           RP1       Fddi0/0
r5             Fddi0/0      173      R           4500     Fddi0
fw/r6         Ser 1/6       136      R           2511     Ser 1
r3#

```

```
r4# show 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
r2             Fddi0/0      169      R           RP1       Fddi0/0
r2             Ser 1/2       169      R           RP1       Ser 1/4
r3             Fddi0/0      123      R           RP1       Fddi0/0
r3             Ser 1/3       123      R           RP1       Ser 1/4
r1             Fddi0/0      131      R           RP1       Fddi0/0
r1             Ser 1/1       131      R           RP1       Ser 1/4
r5             Fddi0/0      169      R           4500     Fddi0
r4#

```

```
r5# show cdp neighbors
```

```

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

```

```

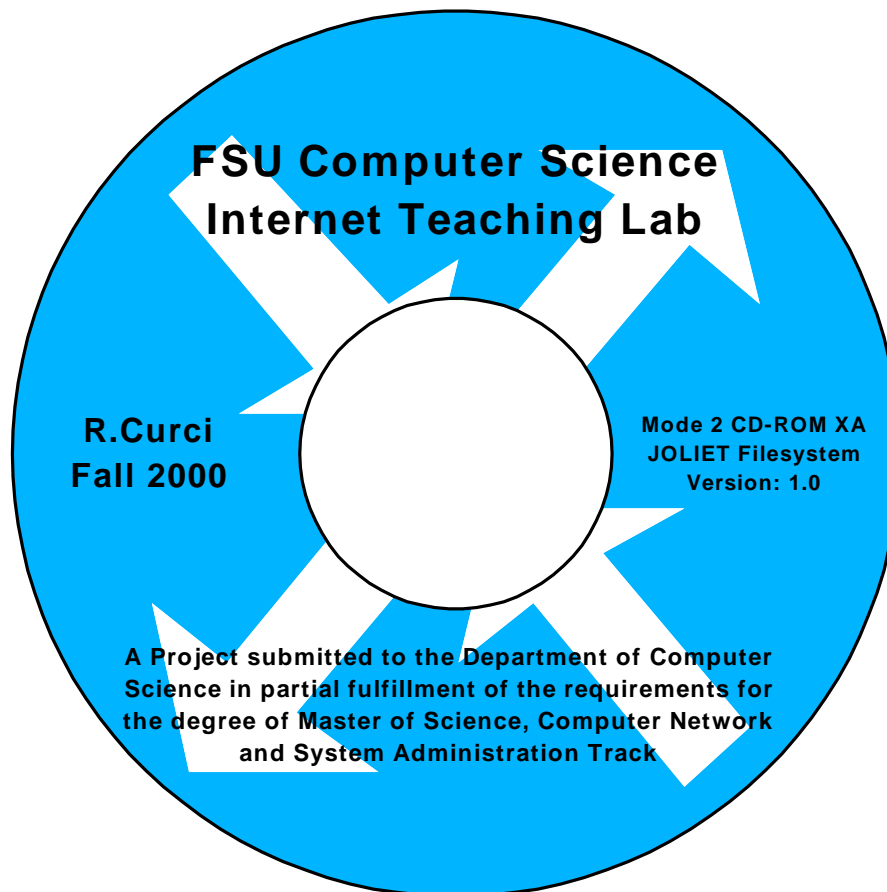
Device ID      Local Intrfce  Holdtme  Capability  Platform  Port ID
r2             Fddi0         166      R           RP1       Fddi0/0
r3             Fddi0         179      R           RP1       Fddi0/0
r1             Fddi0         127      R           RP1       Fddi0/0
r4             Fddi0         178      R           RP1       Fddi0/0
cat1           Fas 0         146      T S         WS-C3524-XFas 0/7
cat1           Eth 1         146      T S         WS-C3524-XFas 0/9
cat1           Eth 0         146      T S         WS-C3524-XFas 0/8
r5#

```

These scripts are of limited usefulness. They are only useful when the routers are properly configured to accept TELNET sessions from the Linux system. I have found them helpful mostly to collect the current configuration and routing tables to quickly give me a snapshot of the state of the system after finishing lab exercises.

These utilities can be found on the accompanying project CD-ROM in the “cisinfo” directory. The files are present individually and as a UNIX TAR archive. Note that PERL and EXPECT are prerequisite and it is sometimes necessary to edit the first line of the scripts to provide the proper fully qualified filename for the PERL and EXPECT executables.

Appendix H: Project CD-ROM



The project CD-ROM contains many files created or downloaded during the course of completing this project. It is a CD-R recordable mode 2 CD-ROM XA disk. It uses a Joliet filesystem and should be readable under Microsoft Windows 95/98/NT/2000. Most original documents were created with Microsoft Word 2000, Excel 2000, PowerPoint 2000 and Visio 2000 Professional which use file extension .DOC, .XLS, .PPT and .VSD respectively. Several Adobe Acrobat portable document format files (.PDF) and hypertext (.HTM) files were derived from the source files. There are also many router configuration files and captured output from text-based router commands stored in files with extension .TXT. The following is a brief description of the CD-ROM directories.

- **acl**
Access Control List sample lab.
- **baseline**
Rudimentary baseline RIP router configuration mentioned in Appendix F.
- **basic**

- Cisco router basics lab that covers inverse telnet, router modes, etc.
- **bgp**
Border Gateway Protocol lab that covers exterior BGP and GRE tunnel interfaces.
 - **cisinfo**
Linux *expect* and *perl* scripts mentioned in Appendix G.
 - **countinf**
Count-To-Infinity lab that explores this problem with distance vector routing protocols like RIP.
 - **debug**
Cisco Router Debugging lab that explores router show and debug commands.
 - **frame**
Frame-Relay Lab that explores both Frame-Relay emulation and the Split-Horizon problem.
 - **hardware**
Source documents from the paper section on router and switch hardware including many drawings and images of the routers, switches, cables, and other components.
 - **igp**
Interior Gateway Protocol lab that explores RIP, OSPF, IGRP, EIGRP, and IS-IS routing protocols.
 - **ios**
Cisco Internetwork Operating System (IOS) images for use with the lab router hardware plus a Windows-based TFTP server implementation that may be used to download new software to the router flash memory.
 - **misc**
Assorted diagrams and notes that did not fit elsewhere.
 - **multi**
Multiprotocol lab that explores the IPX and Appletalk protocols.
 - **paper**
Source documents for this project paper.
 - **photos**
Several photographs of the network lab room, routers, switches, cables, connectors, etc., in .JPG format.
 - **purchase**
Files related to the purchase of the used 2511 router and misc cables and connectors for the lab.
 - **pw-recover**
Files relating to the router password recovery procedure in appendix D.
 - **rip**
RIP protocol lab that explores wide area networking at the Center for Entertainment Studies.
 - **scratch**

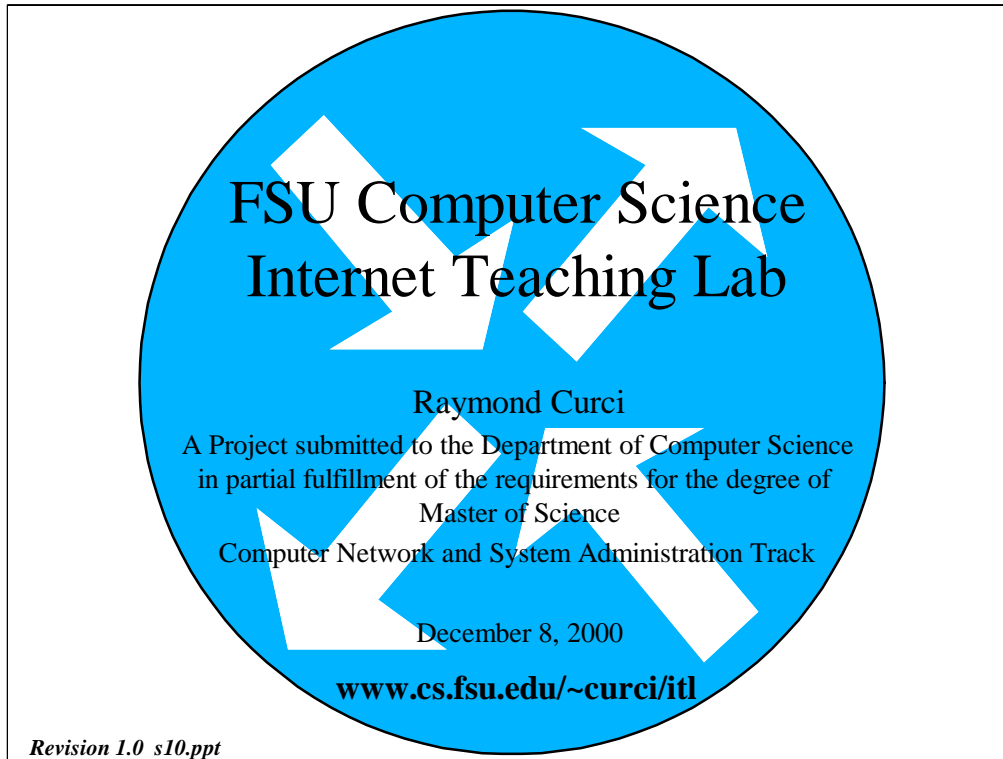
Start-From-Scratch lab that has students rebuild the router configurations after they have been mysteriously erased by the instructor.

- **slides**
Powerpoint slide show presentation given at the project defense, December 8, 2000.
- **spantree**
Spanning Tree Protocol (802.1D) lab.
- **sysadm**
An assortment of many public domain programs utilized in CIS5406 (Computer Network and System Administration) used for testing the network. Many are used in the topology discovery lab and are copied here as a convenience.
- **top**
Topology Discovery lab where students use many PC utilities to learn more information about the lab network without password access to the lab routers and switches.
- **vintl**
The Virginia Internet Teaching Lab sample lab exercises at the University of Virginia.
- **visio**
Computer network stencils for use with Microsoft Visio 2000 Professional packed in two ".ZIP" files. The Windows program "WinZIP v8.0" program is also included to unpack the archive files.
- **vls**
Variable Length Subnet Masking lab which explores VLSM, OSPF, and route summarization.
- **www**
Files related to the World-Wide-Web including source for the initial FSU Computer Science Internet Teaching Lab home page.

Appendix I: Acronyms

ACRONYM	DEFINITION
1000baseLX	Gigabit Ethernet over singlemode fiber standard
1000baseSX	Gigabit Ethernet over multimode fiber standard
100baseFX	Fast Ethernet over fiber standard
100baseTX	Fast Ethernet over UTP standard
802.10	IEEE FDDI trunking protocol (aka SDE)
802.1D	IEEE Spanning Tree standard for bridges
802.1Q	IEEE standard ethernet trunking protocol
ACL	Access Control List
ARP	Address Resolution Protocol
AS	Autonomous System
ASN	Autonomous System Number
AUI	Attachment Unit Interface
AWG	American Wire Gauge
BECN	Backward Explicit Congestion Notification
BGP	Border Gateway Protocol (routing protocol)
BRI	Basic Rate ISDN
CAIDA	Cooperative Associate for Internet Data Analysis
CCIE	Cisco Certified Internetworking Expert
CDP	Cisco Discovery Protocol
CGMP	Cisco Group Management Protocol
CIDR	Classless Internet Domain Routing
CIM	Cisco Interactive Mentor
CLLI	Common Language Location Identifier
CNSA	Computer Network and System Administration
CSU/DSU	Channel Service Unit/Data Service Unit
DAS	Dual Attach Station
DDS	Digital Data Service
DLCI	Data Link Channel Identifier
DSL	Digital Subscriber Line
DVMRP	Distance Vector Multicast Routing Protocol
EBGP	Exterior Border Gateway Protocol
EIA	Electronic Industries Alliance
EIA568	Commercial Building Telecom Cabling Standard
EIGRP	Enhanced Interior Gateway Routing Protocol
FDDI	Fiber Distributed Data Interface
FECN	Forward Explicit Congestion Notification
FRAD	Frame-Relay Access Device
FXO	Foreign Exchange Office
FXS	Foreign Exchange Station
H.323	A video and audio videoconferencing standard
HSRP	Hot Standby Routing Protocol
IAB	Internet Activities Board
IBGP	Interior Border Gateway Protocol

IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IGRP	Interior Gateway Routing Protocol
IMUX	Inverse Multiplexer
IOS	Internetwork Operating System
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISIS	Intermediate System-Intermediate System (routing protocol)
ISL	Inter-Switch Link (trunking standard)
ITL	Internet Teaching Lab
LAN	Local Area Network
LANE	LAN Emulation (ATM)
LMI	Link Management Interface
MAN	Metropolitan Area Network
Mbps	Megabits per second
MII	Media Independent Interface
MM	MultiMode (fiber optic cable)
OCTET	An 8-bit Byte
OSPF	Open Shortest Path First
PDF	Portable Document Format
PIM	Protocol Independent Multicast
POTS	Plain Old Telephone Service
PRI	Primary Rate ISDN
PSTN	Public Switched Telephone Network
PVC	Permanent Virtual Circuit
QoS	Quality of Service
RFC	Request For Comments
RIP	Routing Information Protocol (routing protocol)
RJ45S	8-Position Modular Jack
RS232C	Asynchronous Serial Communication Standard
SAS	Single Attach Station
SDE	Secure Data Exchange (aka 802.10)
SM	SingleMode (fiber optic cable)
SNMP	Simple Network Management Protocol
SPF	Shortest Path First
SVC	Switched Virtual Circuit (ATM)
TCP/IP	Transmission Control Protocol/Internet Protocol
TIA	Telecommunications Industry Association
UTP	Unshielded Twisted Pair
VLAN	Virtual Local Area Network
VLSM	Variable Length Subnet Masking
VoIP	Voice over IP
VPN	Virtual Private Network
WAN	Wide Area Network

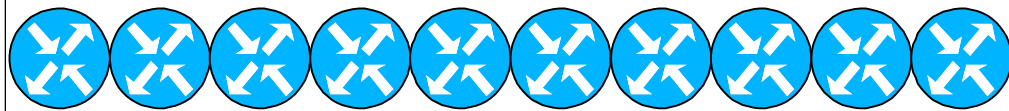


[Revision 1.0, file “s10.ppt”, 05-Dec-2000 R.Curci]

Good afternoon. My name is Ray Curci. I am a computer science student in the computer network and system administration masters track. My project is entitled “FSU Computer Science Internet Teaching Lab” About 6 months ago, my good friend Mike Sloderbeck mentioned that he had read on the Computer Science web page that Computer Science and Information Studies will share a new Internet Teaching Laboratory, with routers provided by the Cooperative Association for Internet Data Analysis or CAIDA. I called the Computer Science chairman, Dr. Ted Baker, to find out more. Dr. Baker explained that some Cisco router equipment was donated and that renovations were underway in the Love building to build a new computer network lab. He encouraged me to talk with Jeff Bauer and Damon Snyder. Jeff Bauer suggested creating network assignments to supplement the assignments in his system administration class. The idea was to give students some hands-on exposure to network router equipment. I worked closely with him to develop labs to dovetail with his system administration class and we turned this effort into a masters degree project.

OBJECTIVE

- Design, implement and document a highly flexible hands-on Internetwork Teaching Lab to augment systems administration and data communications courses including the creation of several sample lab exercises.



The objective of this project is to design, implement and document a highly flexible hands-on Internetwork Teaching Lab to augment systems administration and data communications courses including the creation of several sample lab exercises.



DESIGN GOALS

- Low cost
- Secure design
- Extensible design
- Ease of soft reconfiguration
- Remote access
- Distributed design

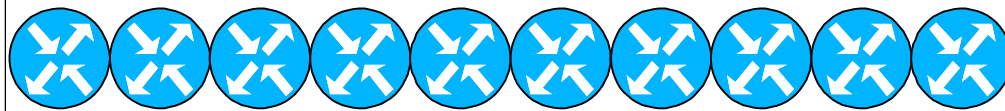
The design goals are as follows:

Low cost, secure design, extensible design, ease of soft reconfiguration, remote access, and distributed design.

I will revisit these design goals in more detail near the end of this presentation.

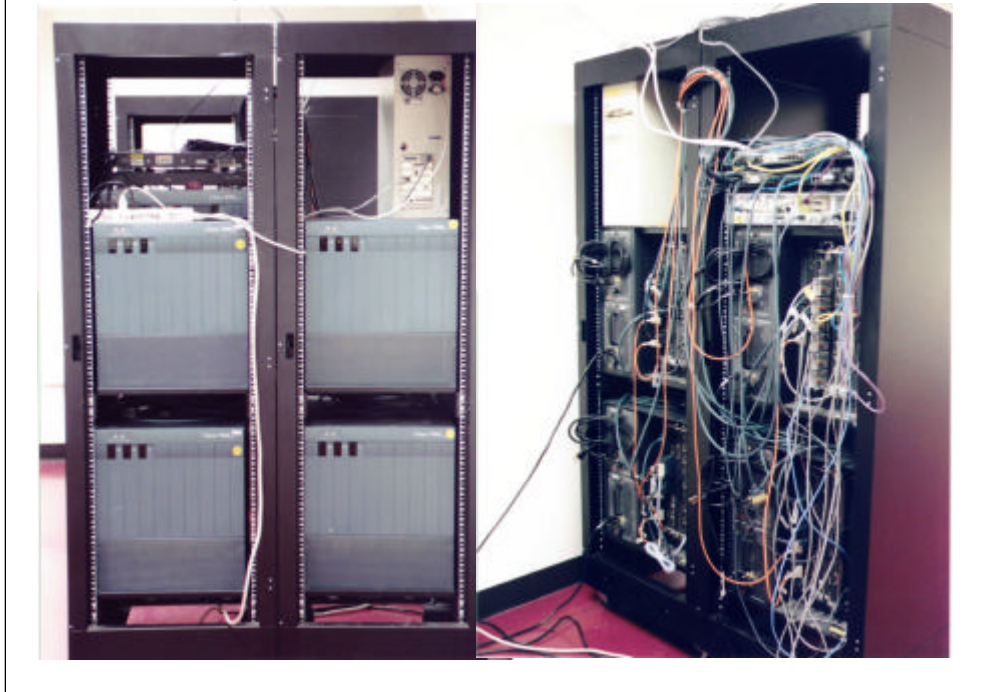
FSU Computer Science Internet Teaching Lab Hardware

- Cisco 7000 routers (R1, R2, R3, R4)
- Cisco 4500 router (R5)
- Cisco 2511 router/firewall (R6)
- Cisco 35xxXL switches (CAT1, CAT2, CAT3)
- UNIX and NT computers



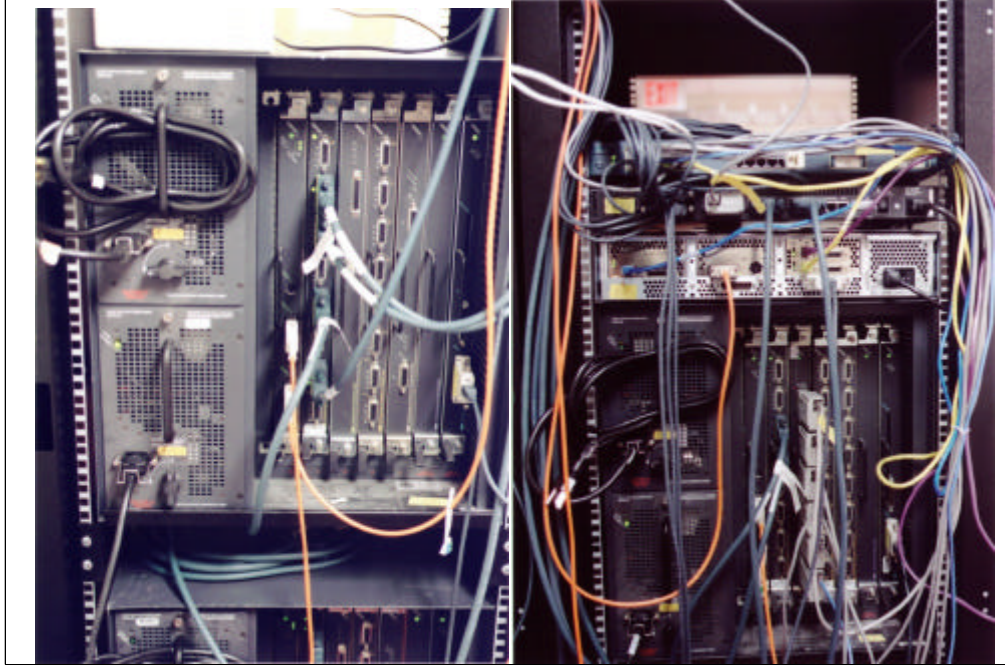
The lab hardware consists of four Cisco 7000 core routers labeled r1,r2,r3 and r4; one cisco 4500 router labeled r5, a cisco 2511 access router labeled r6, three Cisco catalyst 3500XL series layer 2 ethernet switches labeled cat1, cat2, and cat3; and several Linux, Sun, and Windows NT PCs.

EQUIPMENT CABINETS




Here is what the equipment looks like at the front and rear. As you can see, it nearly fills two equipment cabinets. The four large 145 pound 7 000s are the large routers on the bottom. As you can see, there are many cables. I tried to locate most equipment with connections to each other in a single cabinet to minimize the number of cables that span cabinets to make it easier to disconnect and move the cabinets individually. We also have some color-coded patch cables and wire management devices on order that have not yet been received to make the wiring more neat.

EQUIPMENT CABINETS 2

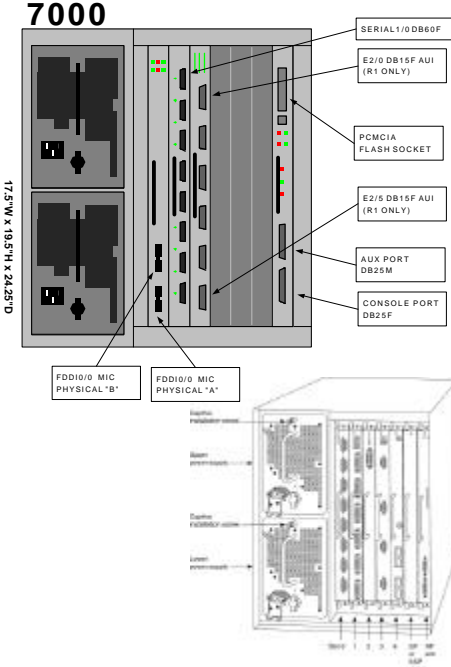


Here is a closer look.


Cisco 7000 Routers (R1,R2,R3,R4)



- Donated by CAIDA
- FDDI and Serial WAN ports
- Single 6-port Ethernet card donated by Cisco
- End of life
- Previously deployed on InternetMCI backbone.
- 5 slots available for interface cards
- 145 lbs each

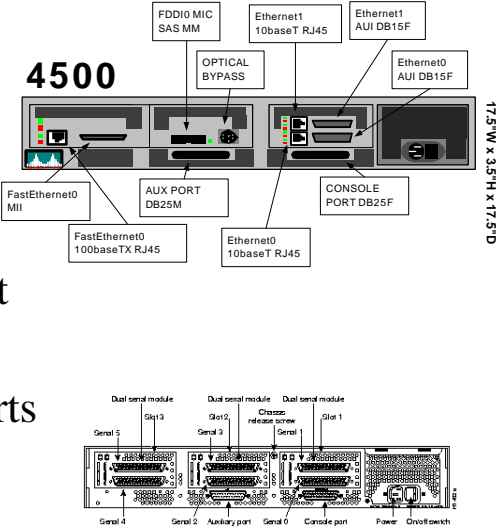


These are the four routers donated by CAIDA which included FDDI and serial WAN interface cards, but no ethernet cards. Phil Jensen, a systems engineer from the local Cisco office, donated a single 6-port EIP6 10Mbps ethernet card. The four routers are labeled “R1”, “R2”, “R3”, and “R4” They are identically configured except that R1 has the additional 6-port ethernet card. These routers reached their end of life a few years ago and are no longer supported. The latest software is version 12.1 which no longer supports this architecture. We are using version 11.1 software which is about 5 years old. Several of the newer software features such as network address translation or NAT are not supported. These units were previously deployed on the MCI Internet backbone. Each chassis has 5 slots available for interface cards. Each has dual power supplies you can see on the left for redundancy. Each fully-loaded router weighs 145 pounds and were a struggle to set in the equipment cabinets. It took four people to carefully lift each router into place.



Cisco 4500 Router (R5)

- Formerly CS Department Router with upgraded memory
- 1x FastEthernet port
- 2x Ethernet Ports
- 1x FDDI Port

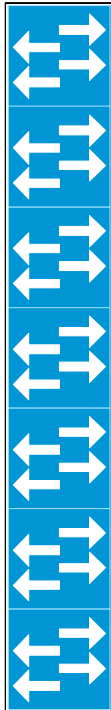


For about the past 5 years, the Computer Science Department was connected to the FSU FDDI backbone network with a Cisco 4500 router. Its main purpose was to provide access lists to secure the departmental network. Recently, the department has installed a Cisco Catalyst 6000 switch with a multilayer switching feature card (MSFC) which replaced the 4500 . After adding additional memory and upgrading the software, we have incorporated this router into the lab. This is still a current model and is supported by the latest IOS software. It has a fast ethernet port capable of ISL trunking, two 10 meg ethernet ports, and a single-attach FDDI port.

Cisco 2511 Router/Firewall (R6)

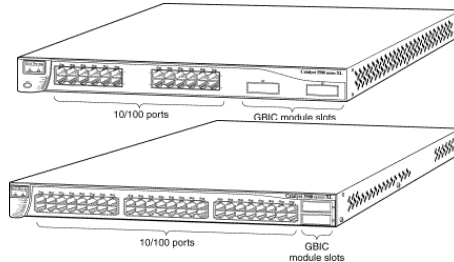
- Purchased used
- 1x Ethernet
- 2x Serial WAN
- 16x Async RS-232
- Out of band console access via “Inverse Telnet”
- Doubles as firewall

This cisco 2511 access router labeled “R6” is the only router that was purchased and we were able to find one used. It has an ethernet port for connecting to the computer science network, two serial WAN ports that connect to the other lab routers, and 16 asynchronous RS-232 ports. The RS-232 ports connect to the console ports on the other router and switch devices. You can TELNET to this router and use an IOS software feature called “inverse telnet” for an out-of-band connection to the other routers. Out of band access is very important since a lab network is often in flux where ordinary access through TELNET is not possible.



Cisco Catalyst 3524XL & 3548XL Ethernet Switches (cat1, cat2, cat3)

- Layer 2 10/100 Ethernet Switches
- Gigabit Ethernet Uplinks with 1000baseSX GBICs
- Supports VLANs
- Supports ISL and 802.1Q Trunking

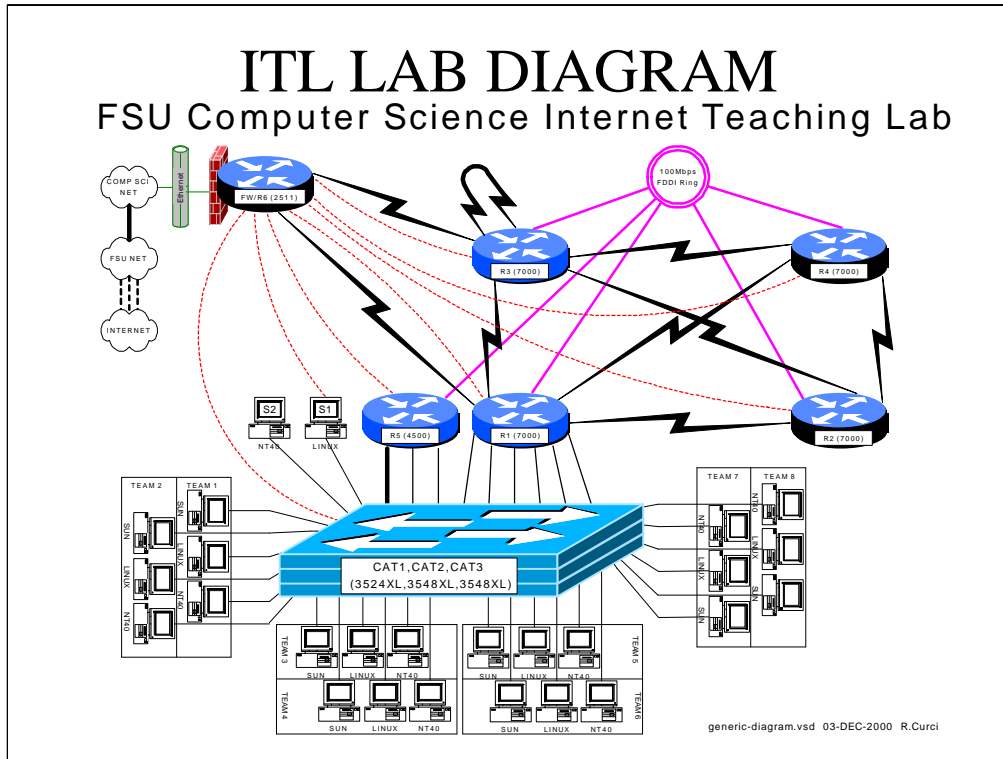


The Cisco catalyst 3500XL series is a layer 2 10/100 ethernet switch. Each has two gigabit ethernet uplinks. Each supports VLANs and trunking protocols. The 3524XL has 24 10/100 ports while the 3548XL has 48 10/100 ports. The 24-port switch labeled “cat1” is situated near the router equipment with connections to the ethernet and fast ethernet ports on routers R1 and R5 and will eventually be located in the computer machine room. No rmally each of these ports would be in a separate VLAN corresponding to a student team. The two 48-port switches will be located in the network lab room with connections to the student PCs and instructor PC. These 96 ports will allow for approximately 15 teams with 3 PCs each. Gigabit 1000baseSX trunks between switches allow the VLANs to span switches. The location of the equipment in the computer machine room is partly to reduce the n oise level in the network lab from the noisy 7000 routers and partly to improv e the physical security of the equipment.

UNIX and NT Computers

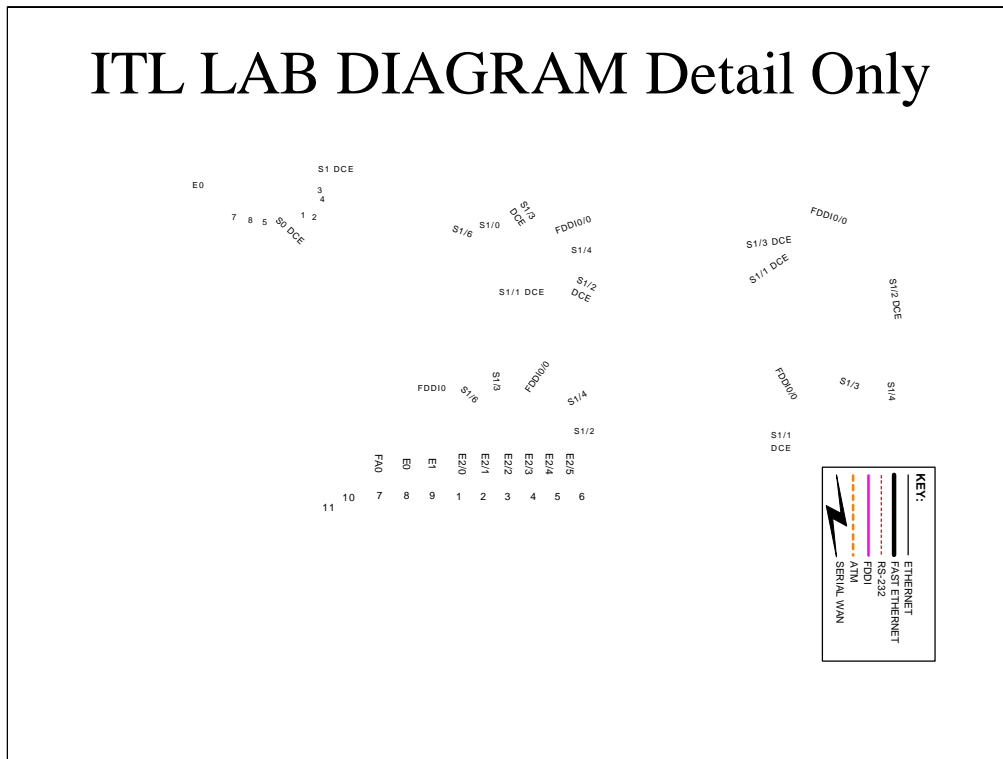


The student UNIX and NT PC computers are located in the newly renovated network lab in the basement of the LOVE building. These photos were taken before all the PCs were installed. There are 10 tables with 2 student team workspaces per table to handle 20 student teams with 3 PCs per team or 60 PCs total. The image in the bottom left corner shows how the lab PCs are depicted on many of the diagrams.



Here is a typical VISIO diagram depicting the lab. The six round dark blue icons are the routers. The rectangular light blue icon represents the stack of 3 catalyst ethernet switches. You can see two sample PC servers that I used for testing the sample lab exercises labeled “S1” and “S2”. You can also see near the bottom, several student PCs in groups of 3 representing teams that are only partially represented due to lack of space. Routers r1 through r4 are the 7000s; R5 is the 4500; and R6 is the 2511 firewall router. As you can see, the single choke point where the lab network connects to the outside computer science network ethernet has access lists depicted by the red brick “firewall”. The Computer Science Network connects to the FSU Network, which in turn connects through ATM to the Internet. As you can see, routers R1 through R5 connect to a common 100Mbps FDDI ring shown in purple. The black lightning bolts are serial WAN connections normally configured for 2Mbits/second and simulate T1 lines. The ethernet and fast ethernet ports on routers R1 and R5, as well as the PCs attach to catalyst switch ports that allows for a flexible grouping of VLANs that can be easily reconfigured in software. The red dashed lines are RS-232 serial links from the 2511 firewall to the routers and switches allowing for out-of-band communication with the console ports on the other devices even if the network is in a non-functional state.

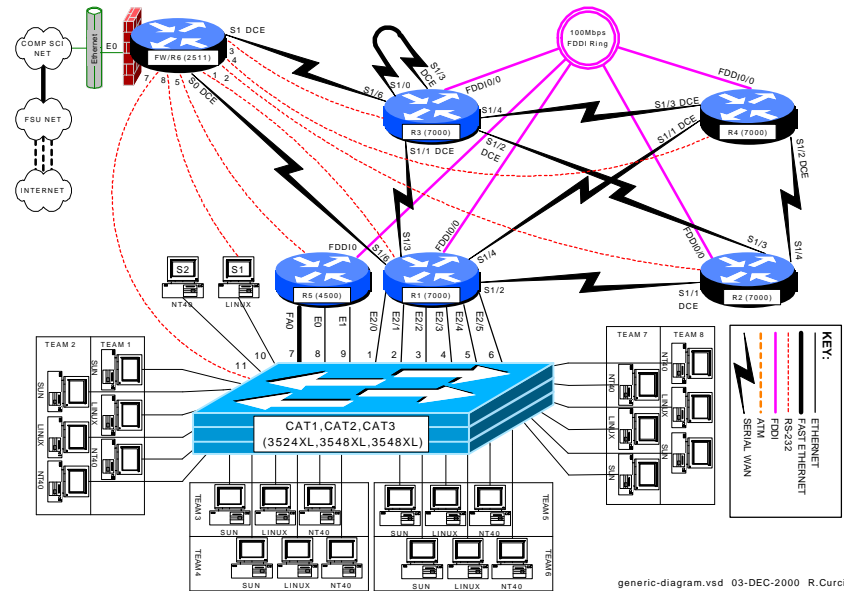
ITL LAB DIAGRAM Detail Only



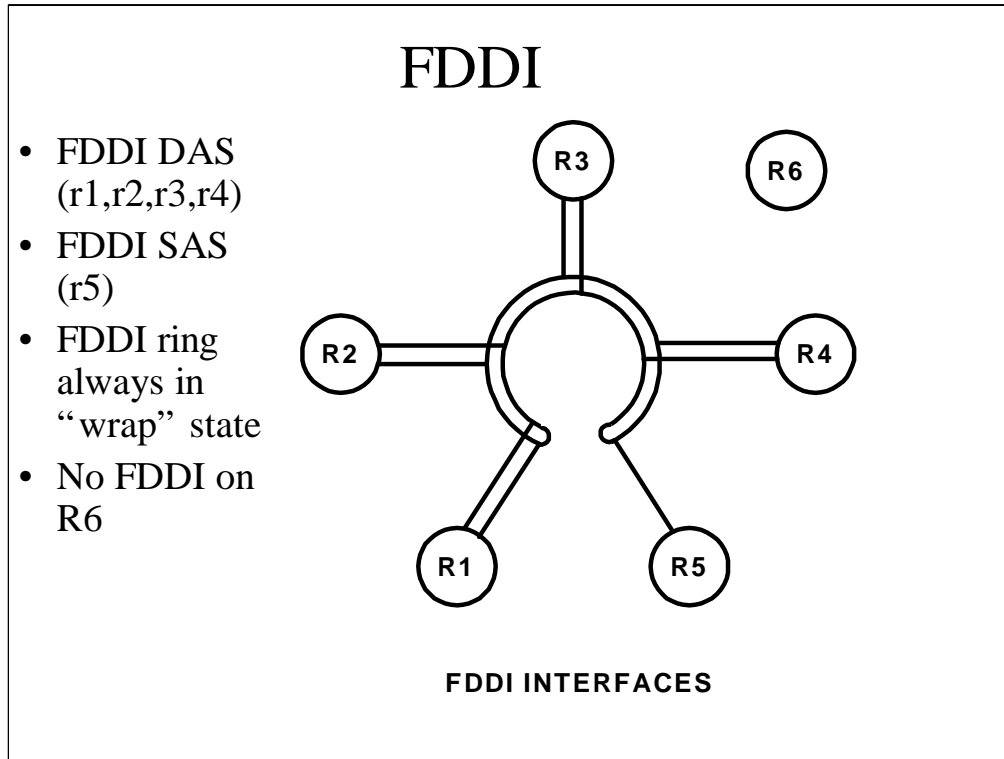
Here is a separate layer of the previous diagram with interface detail information which we will combine with the previous image in the next slide.

ITL LAB DIAGRAM w/DETAIL

FSU Computer Science Internet Teaching Lab



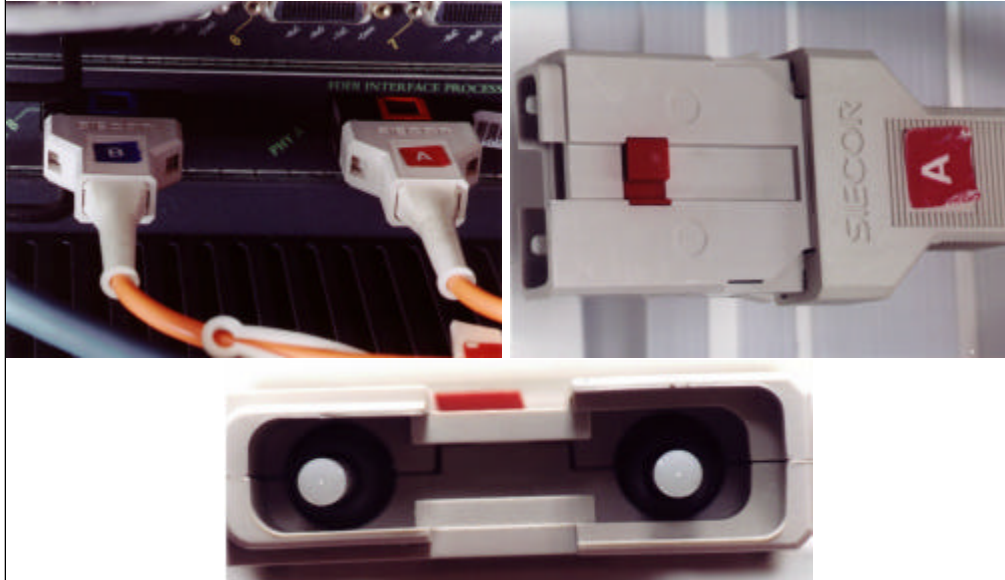
Here is the same diagram with interface details combined. This is actually the same VISIO drawing but with an extra layer containing the interface names and key displayed. This technique of using multiple drawing layers was handy in some cases for creating similar student and instructor versions of some of the sample lab diagrams.



I want to briefly discuss FDDI, serial WAN interfaces, and frame-relay because most people are not very familiar with them.

For those not familiar with FDDI, it utilizes fiber optic cable to normally build two counter-rotating rings for redundancy. When broken, the ring goes into what is called a “wrapped” state. As depicted on this diagram, you can see that router R5 is only singly attached to the ring because it has a single-attach interface whereas R1 through R4 have dual-attach FDDI interfaces. This means that our FDDI ring is always in the wrap state. Router R6 has no FDDI interfaces.

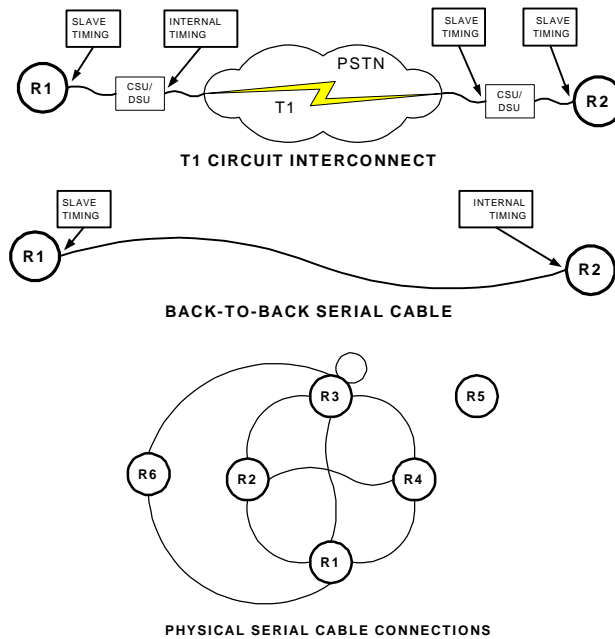
FDDI 2



Here are what the FDDI MIC cables look like. Usually the cables are labeled “A” and “B”, color-coded, and keyed to get in the right order. On the photo on the bottom, you can see the 62.5 micron multimode fiber in the center of the fibers.

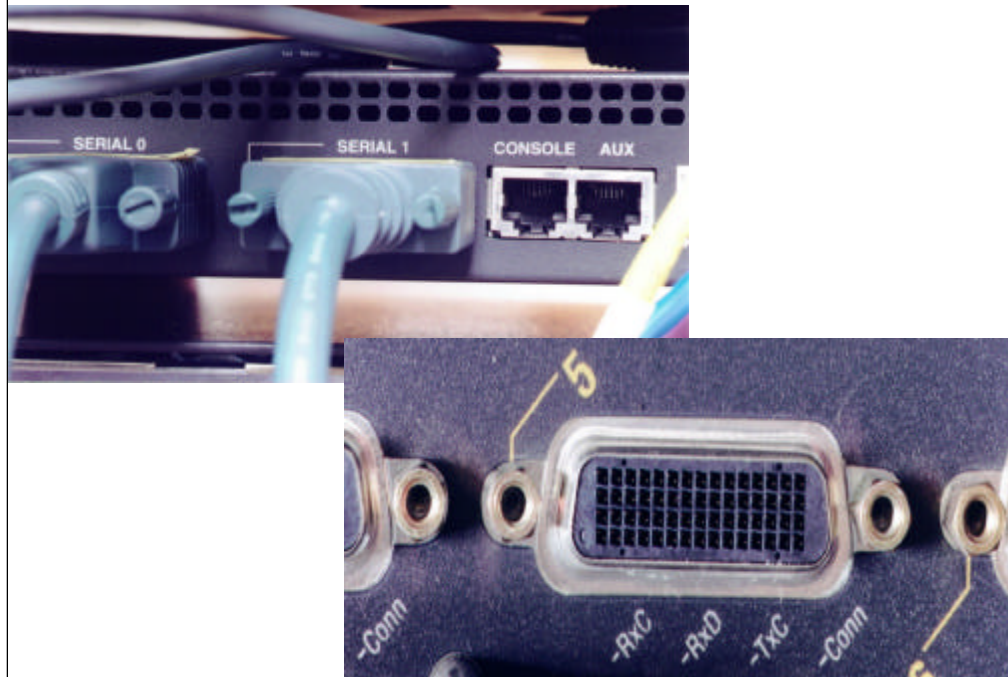
Serial WAN

- CSU/DSUs used on 56K, T1, and T3 WAN data circuits.
- Lab uses special back-to-back serial cables without CSU/DSUs



For those not familiar with serial WAN interfaces, I'd like to talk about them a bit. Normally, WAN circuits are provided by the phone company in the form of 56K, T1, or T3 lines. Normally a channel service unit/data service unit or CSU/DSU sits on either end of the circuit. One end of the CSU/DSU connects to a router's synchronous serial WAN port using the V.35 protocol. The other end connects to the TELCO equipment. Telco equipment robs bits in the data stream for inband signaling, framing, and testing. The CSU/DSU converts between a stream of bits on the router side to the proper framing on the telco side. With CSU/DSUs, routers are normally configured as data terminal equipment or DTE and slave their clocks off the CSU/DSU. In this lab, since we have no CSU/DSUs, we are using special back-to-back serial cables which one router is configured as DTE and the other DCE. In this case, the DCE must supply the clocking.

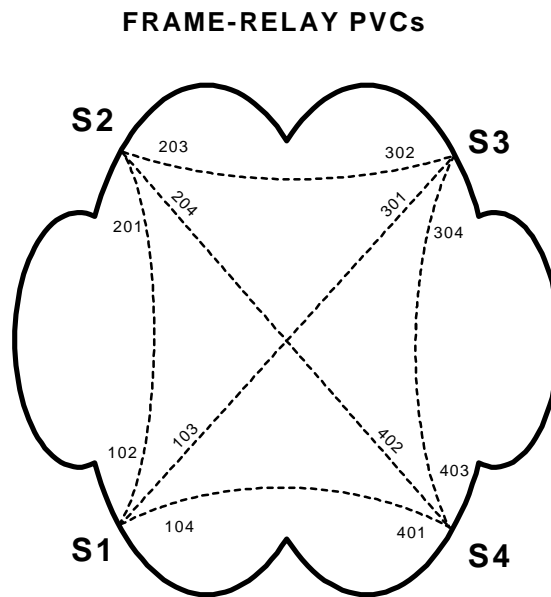
Serial WAN 2



Here are two photos of the serial WAN back-to-back cables and router connector. The connectors are called “DB60” because they have 60 pins and are inside a shell shaped like the letter “D”. Electrically, they use the V.35 signaling standard.

Frame-Relay

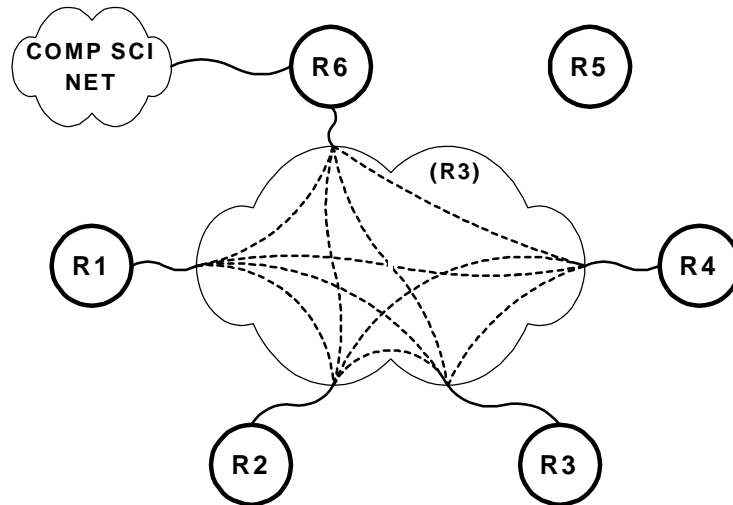
- WAN service sold by phone companies
- Physical data circuits connect routers to nearest frame switch
- PVCs are provisioned across the cloud between routers



Frame-Relay is a wide area network service sold by phone companies. A physical data circuit is established between routers and the nearest frame-relay switch in the cloud. Private virtual circuits are provisioned through the network to interconnect the routers. A full mesh of PVCs does not scale well and can be expensive due to $N(N-1)/2$ PVCs. The most common configuration is a hub-and-spoke topology where the PVCs home back to a single router.

Frame-Relay 2

- R3 can emulate Frame-Relay network
- All routers with serial WAN ports can participate, except R5



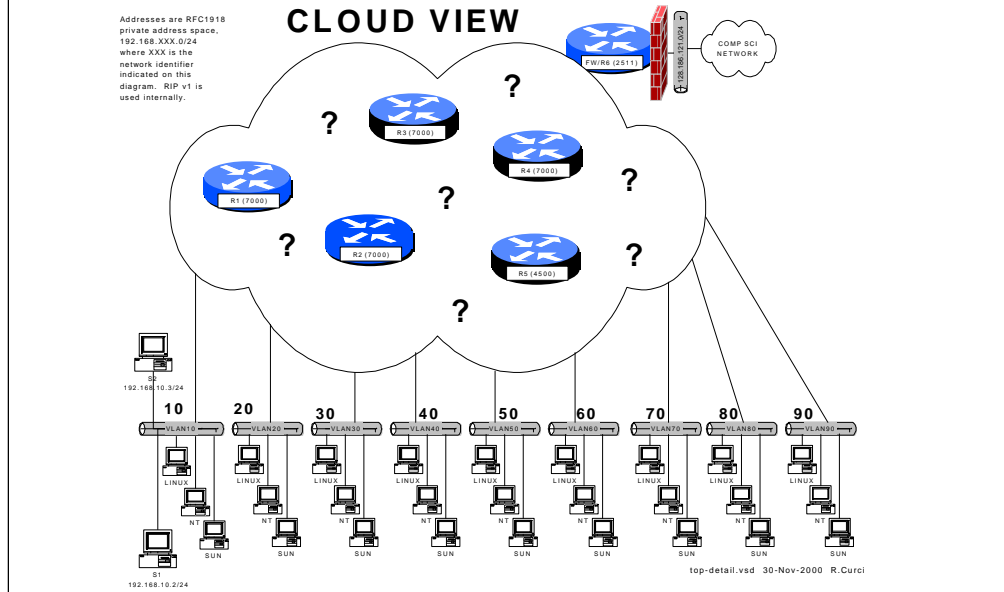
All routers with a serial WAN interface have a connection to router R3. Those ports on R3 can be configured to emulate a frame-relay switch network. The dotted lines in the figure represent PVCs provisioned in the network that can be created between any ports on the emulated frame-relay switch. Router R5 has no serial WAN interfaces and cannot participate. One of the labs explores frame-relay in detail.

Sample Lab Exercises

- Cisco Router Basics
- Cisco Router Debugging
- **Topology Discovery Lab**
- **Start-From-Scratch Lab (RIP)**
- **RIP Lab**
- IGP Lab (RIP, OSPF, IGRP, EIGRP, IS-IS)
- **Variable Length Subnet Mask Lab with OSPF**
- **BGP Lab with Tunnels**
- **ACL Lab**
- **Frame-Relay Lab**
- **Multiprotocol Lab (IPX, Appletalk)**
- Spanning Tree Lab
- Count-To-Infinity (RIP) Lab

Here are some of the sample lab exercises developed. Most consist of a Visio diagram and Microsoft Word document. The Word documents use a special style applied to sections with sample solutions and instructor notes which uses hidden text. This allows a single document to be maintained that can be printed with the hidden text disabled or enabled to produce a student or teacher version of the document. We will look at the boldfaced labs in a bit more detail.

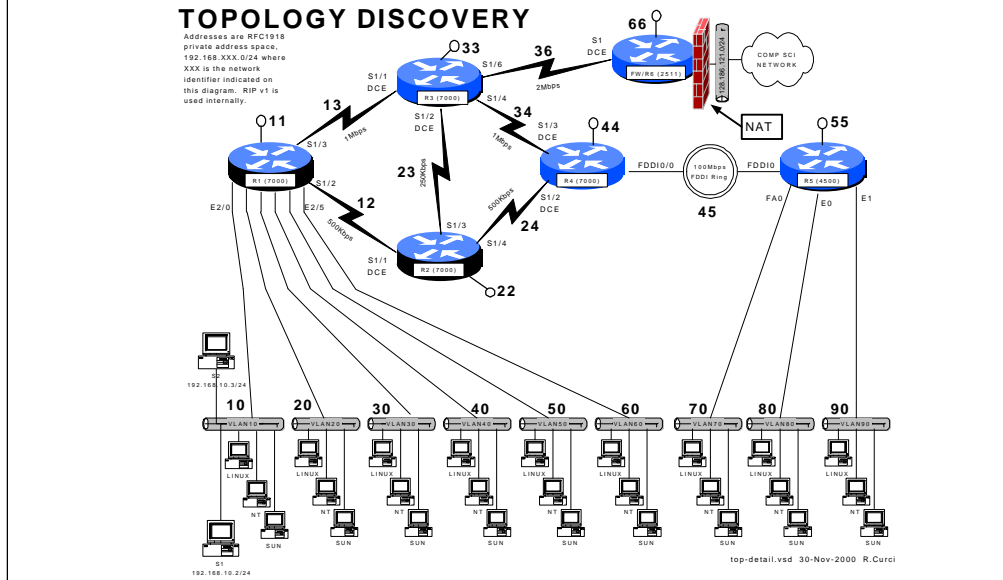
Topology Discovery Diagram (Student Version)



In this lab, the student is not given access to the routers and is given very limited information on the network as shown above. There are several tasks where the students use tools on their PCs to figure out the topology. For example, students can listen to RIP routing announcements to learn which lab networks are present. Utilities like traceroute can be used to find the network path between two points. Tools such as IPERF and TTCP can measure link bandwidth. SNMP can be used to query MIB variables to learn interface names, their bandwidth, system names, MAC addresses, routing table entries, etc. Normally, this lab would be used first before the student learns much detail about the lab equipment.

As you can see, we tell students the number of routers, that private class C IP networks beginning with 192.168 are used, that all networks use a 24-bit mask, and that RIP version 1 is deployed.

Topology Discovery Diagram (Instructor Version)



Here is the teacher version of the diagram with details of route r names, interface names, interface types, and bandwidth information present. Here is an example of where the Visio layering capability is used to produce two diagrams in a single document by selectively enabling layers.

Sample Instructor Notes

For each of your routers, look up the following MIB variables:

`system.sysDescr.0`

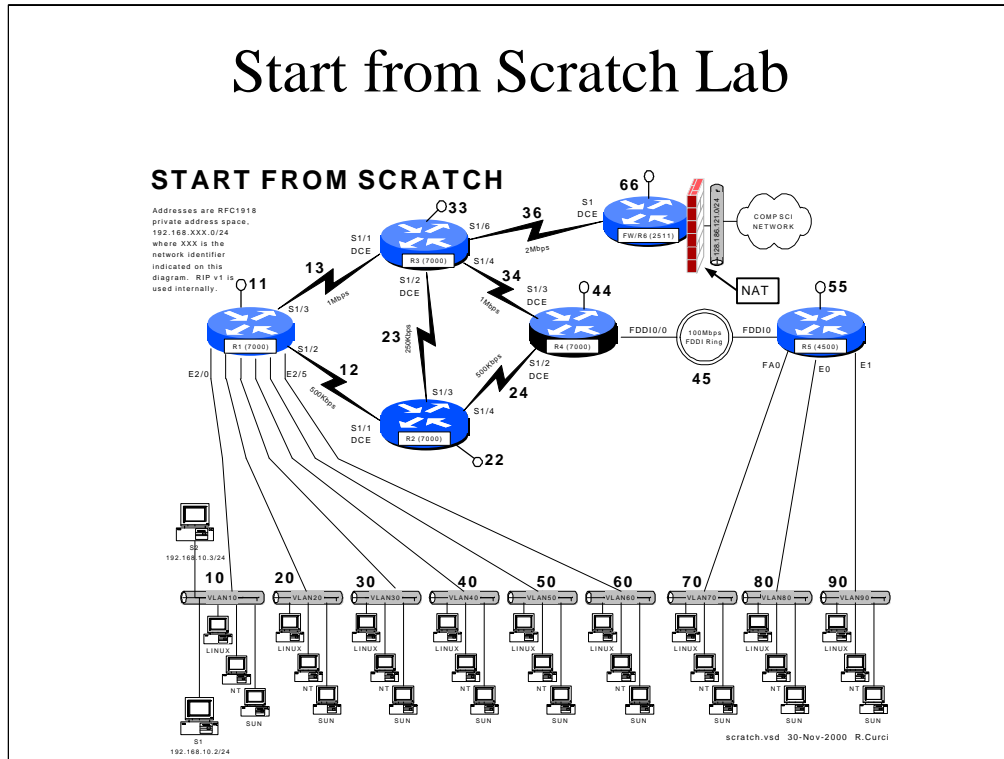
`system.sysName.0`

This will let you see the router names to eliminate any duplicates if you previously found more than one IP address for the same router. Using the system description, note the IOS software version of the router. You should now have enough information to draw a diagram of the six routers with the interface names, interface types (`ethernet`, `fdi`, `point-to-point`, `loopback`), how they connect to each other, and the IP addressing scheme.

```
# For each router, get the SNMP name and software version:
#
[root@sl top]# snmpget -v 1 192.168.11.1 public system.sysDescr.0
system.sysDescr.0 = Cisco Internetwork Operating System Software
IOS (tm) GS Software (GS7-J-M), Version 11.1(24), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Mon 04-Jan-99 21:19 by richv
[root@sl top]# snmpget -v 1 192.168.11.1 public system.sysName.0
system.sysName.0 = r1
#
# You can get the interface names along with their network address,
# IP address, and I/O stats:
#
```

Here is an example section of the instructor version of the topology discovery lab. Sections of text identified by the red bar on the right of the screen are hidden text not present in the student version of the lab document.

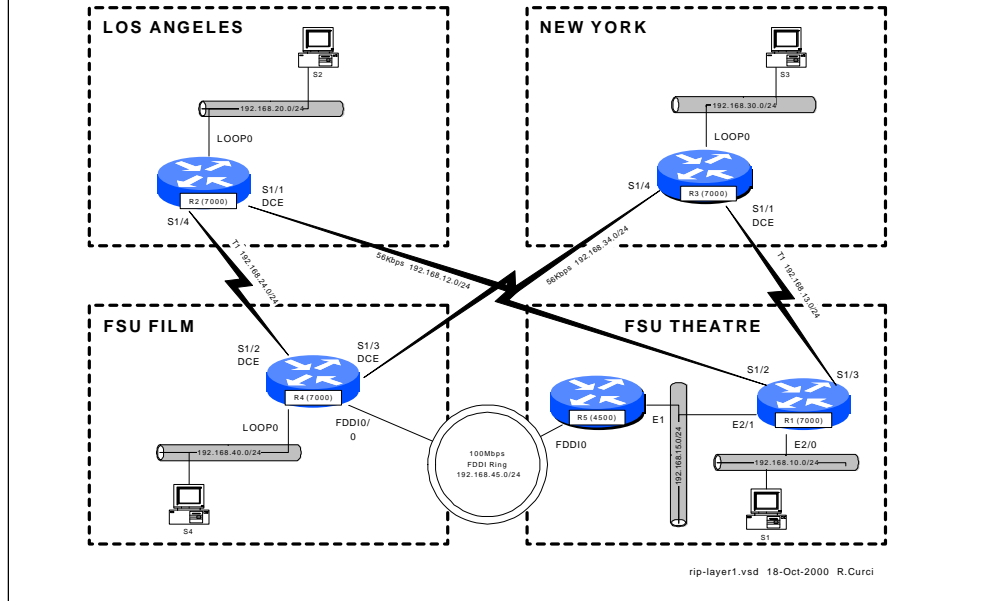
Start from Scratch Lab



This is the start-from-scratch lab. Once exposed to the lab network, the instructor erases the router configurations and the students work as a team to configure the network to the specifications detailed in this diagram. This lab also introduces the student to configuring RIP protocol, virtual loopback interfaces, and to the network addressing scheme. Networks between routers, for example, are numbered by concatenating the router identifiers with the smaller identifier first. Team ethernet networks are numbered as 10 times the team number. Students are exposed to some network lab conventions that allow the students to focus less on the mechanics of memorizing identifiers and more on the interesting topics.

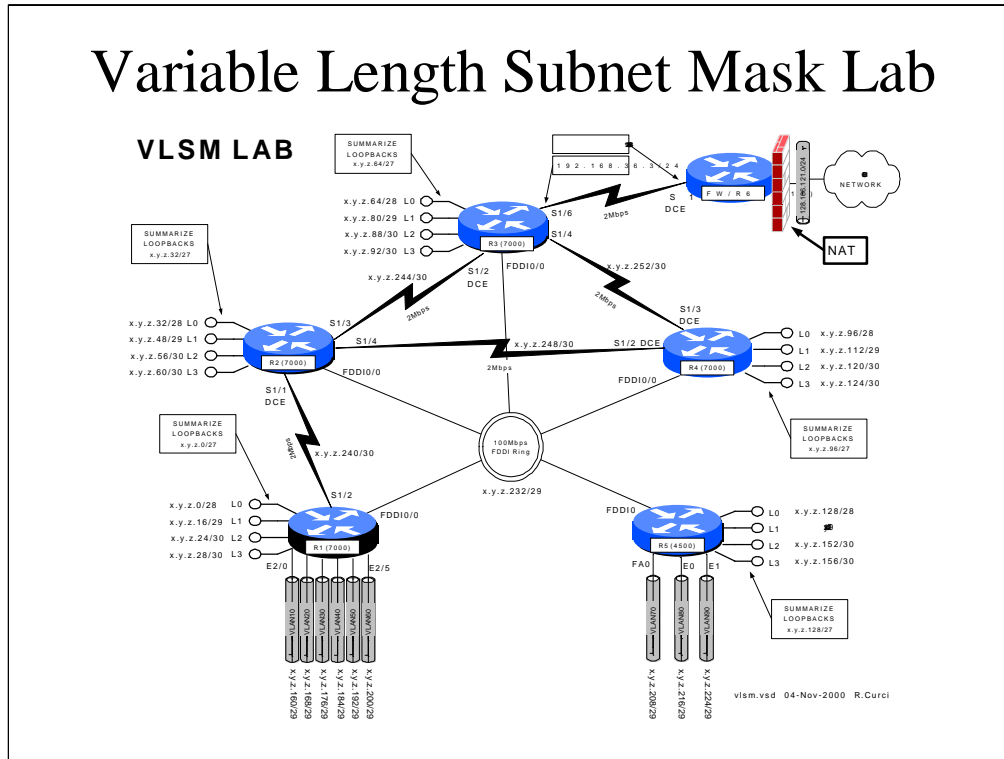
Routing Information Protocol LAB

CENTER FOR ENTERTAINMENT STUDIES



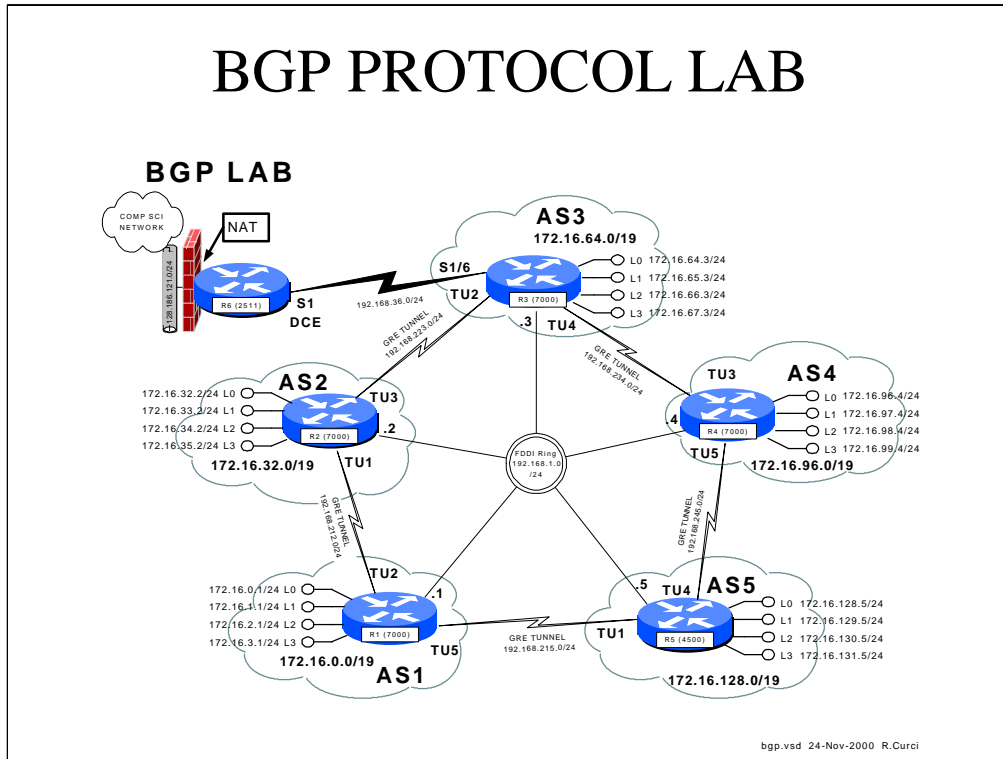
This lab explores the RIP routing protocol with a real -world example of a coast-to-coast network. Faculty from the FSU film school and theatre department help the governor improve his public speaking skills who is reelected. To show his appreciation, the governor convinces the legislature to fund the FSU center for entertainment studies including new branch campuses in Los Angeles CA (Hollywood) and New York City (Broadway). The students are network managers that must implement a cost effective wide area network to tie together the four locations. Things are arranged so that the limitations of the distance vector RIP protocol will cause problems. For example, traffic between the FSU theater department on campus and film school at the FSU University Center sometimes traverses slow WAN links to New York or Los Angeles. Students also have a monthly budget for WAN circuits and must figure out the best solution that balances cost, redundancy, and reliability. Even though this diagram looks very different than the previous one, only soft reconfiguration and use of virtual loopback interfaces is required with no physical lab cable moves.

Variable Length Subnet Mask Lab



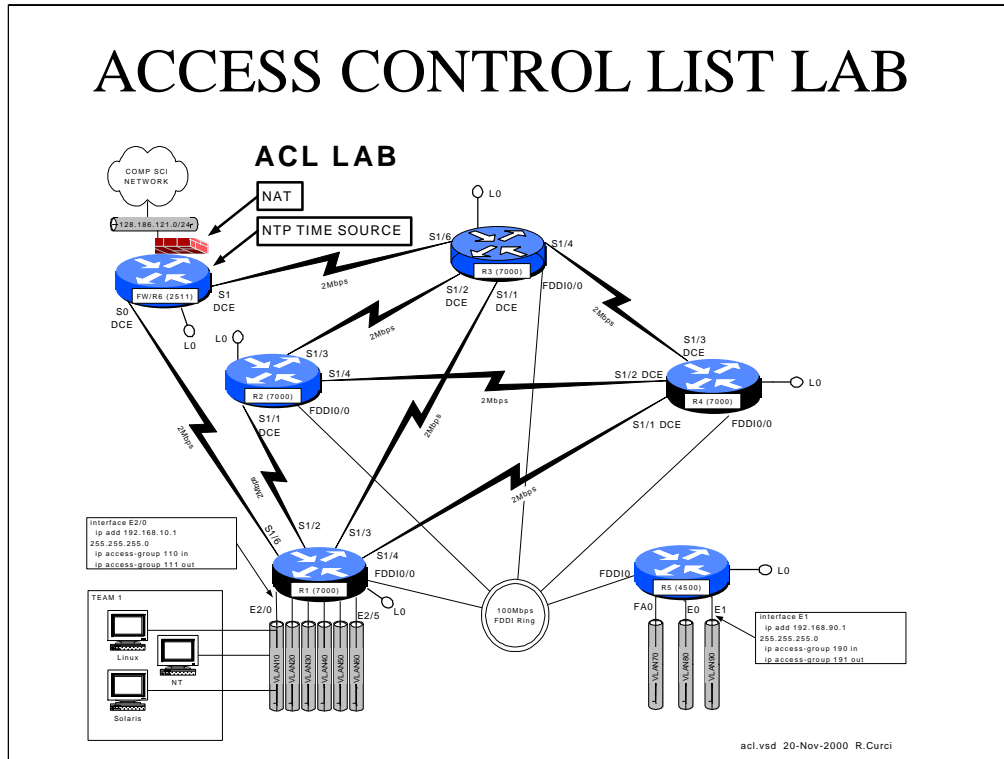
This lab explores variable length subnetting, OSPF areas, and route summarization. Most lab exercises use a class C network for each IP network which is convenient but a very inefficient use of IP address space. With the depletion of IP address space, techniques like VLSM and VLSM-aware routing protocols like OSPF are important to efficiently utilize IP address space. In this lab, the entire network must be readdressed using a single subnetted class C network. The lab is constructed so that the student must count the number of devices on each network segment to figure out the minimal size address block that will work and there are exactly enough addresses to solve the problem with none extra. Students also create virtual loopback addresses that are placed in different OSPF areas and summarized into OSPF area zero.

BGP PROTOCOL LAB



The BGP protocol is used on the backbone of the Internet today. Networks are grouped into autonomous systems. BGP route attributes such as AS PATH are used to calculate the best routes. An AS PATH is an N-tuple listing the sequence of autonomous systems from the router's AS to the remote AS. Routers in different autonomous systems exchanging BGP routes are called "BGP peers." Because IP address summarization is crucial to the scalability of the Internet in reducing backbone router tables, this exercise has the students summarize the addresses on the virtual loopback interfaces into single advertisements to the BGP peers. This lab also explores the GRE tunnel interfaces. Tunnels are virtual point-to-point links between routers. They appear on the router similar to a serial interface. They are implemented in software and encapsulate the data payload inside an IP packet.

ACCESS CONTROL LIST LAB

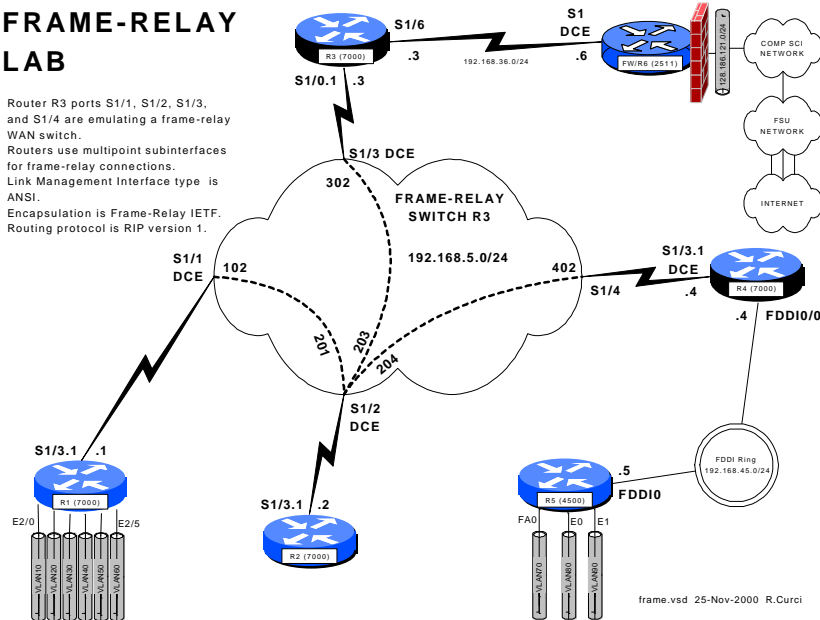


Access control lists or ACLs provide security. They can be applied to the input or output of an interface to selectively allow or deny particular traffic. In this lab, students must create and apply moderately complex access lists to their team's ethernet interface to allow traffic to pass according to a security policy. This helps give students an appreciation for the complexity of Internet firewalls.

FRAME-RELAY LAB

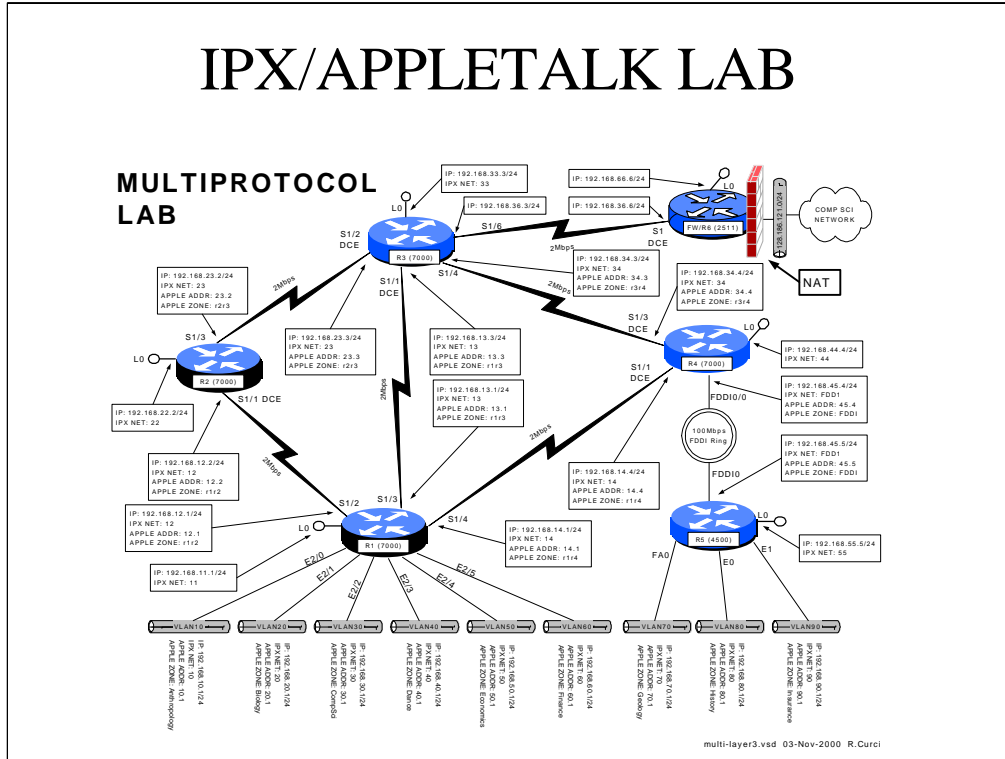
FRAME-RELAY LAB

Router R3 ports S1/1, S1/2, S1/3, and S1/4 are emulating a frame-relay WAN switch.
Routers use multipoint subinterfaces for frame-relay connections.
Link Management Interface type is ANSI.
Encapsulation is Frame-Relay IETF.
Routing protocol is RIP version 1.



Cisco routers can emulate frame-relay WAN switches. Unused ports on router R3 are configured to emulate a frame-relay switch. Routers R1, R2, R3, and R4 then have a single WAN connection to the frame-relay network, but there is only a partial mesh of frame-relay PVCs in a hub-and-spoke configuration. In a business with offices in many cities, for example, the hub might connect to the headquarters site with spokes to the branch sites. This lab creates problems with frame-relay virtual circuit mappings and RIP split-horizon problems for the students to figure out.

IPX/APPLETALK LAB



Although TCP/IP is the most popular network protocol, both IPX and Appletalk are also important. The IPX protocol is still widely used with Novell file servers, while Appletalk is widely used with Apple Macintosh computers for file sharing and printer sharing. For example, many of the networks at FSU have IPX and or Appletalk running on the same ethernet segments as TCP/IP. In this lab, students are exposed to IPX and Appletalk and configure these protocols on a TCP/IP only network.



DESIGN GOALS

- Low cost
- Secure design
- Extensible design
- Ease of soft reconfiguration
- Remote access
- Distributed design

Let's review the design goals given earlier and elaborate on how they were achieved:

Low Cost – the cisco 7000 routers were donated by CAIDA and ethernet card by Cisco. The Cisco 4500 was taken out of service when the department upgraded their network connection. The main equipment costs are the cisco 2511 router which was purchased used, and the three cisco catalyst 3500XL layer 2 switches.

Secure Design – the cisco 2511 firewall router's ethernet port is the only connection between the lab to the outside network and provides a single choke point. Access lists and network address translation provide access security.

Extensible Design – The lab design can be easily extended, especially by adding additional PCs given the flexible VLAN design.

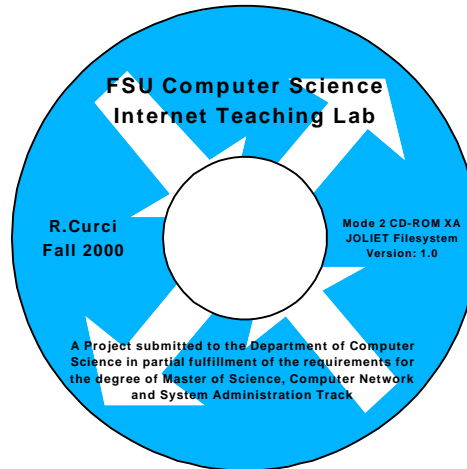
Ease of Soft Reconfiguration- All of the sample labs you will see later on in this presentation are constructed without the need to move any cables. Reconfiguration is accomplished by selectively enabling or disabling interfaces, through switch VLAN reconfiguration, frame-relay WAN emulation, and through creating of virtual loopback and tunnel interfaces.

Remote Access- With the proper access, it is possible to log into the R6 firewall router and then connect to the routers or PCs.

Distributed Design- The gigabit trunking between ethernet switches allows the routers to be located in a different room than the student PCs to reduce the noise level and improve security.

Project CD-ROM

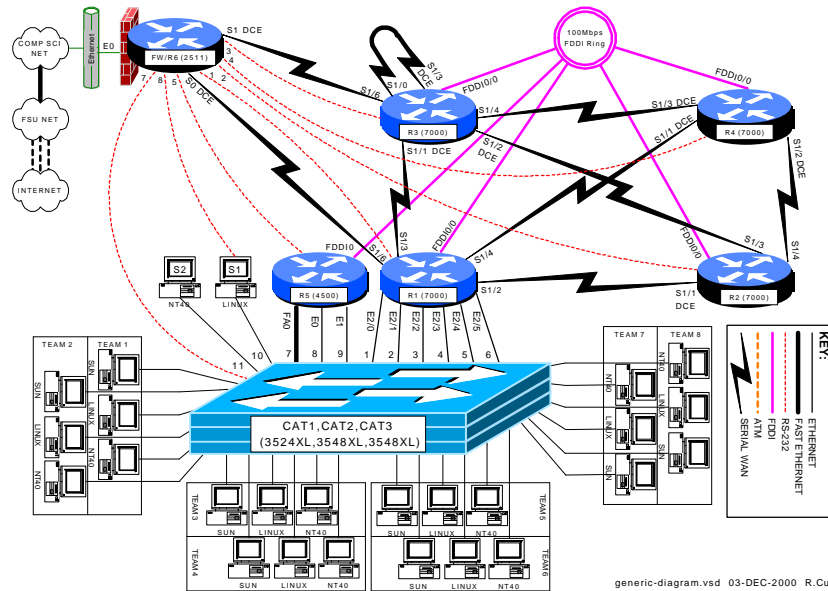
- Paper
- Slide show
- Labs
- Visio stencils
- Sysadm Utilities
- Instructor use only!



There is a project CD-ROM that contains the project paper, slide show, labs, Visio stencils, drawings, and many system admin utilities. The labs contain many extra files including router configurations, sample output from various SHOW commands, et.c. A lot of work went into this project and there are many additional files. This slideshow only scratches the surfac e.

THE END

FSU Computer Science Internet Teaching Lab



The End. Thank-You. Do you have any questions?

FSU Computer Science Internet Teaching Lab

Student Edition PDF File Index

Raymond R. Curci
12-Dec-2000

student-pdf-files.doc

LAB1: Cisco Router Basics

basic-student.pdf
basic-diagram.pdf
basic-fsm.pdf

LAB12: Spanning Tree 802.1D

spantree-student.pdf
spantree-diagram.pdf

LAB2: Cisco Router Debugging

debug-student.pdf
debug-diagram.pdf

LAB13: Count-To-Infinity

countinf-student.pdf
countinf-diagram.pdf

LAB3: Topology Discovery

top-student.pdf
top-diagram.pdf

LAB4: Start-From-Scratch

scratch-student.pdf
scratch-diagram.pdf

LAB5: Routing Information Protocol

rip-student.pdf
rip-diagram.pdf

LAB6: Interior Gateway Protocols

igp-student.pdf
igp-diagram.pdf

LAB7: Variable Length Subnet Masks

vlsm-student.pdf
vlsm-diagram.pdf

LAB8: Border Gateway Protocol

bgp-student.pdf
bgp-diagram.pdf

LAB9: Access Control List

acl-student.pdf
acl-diagram.pdf

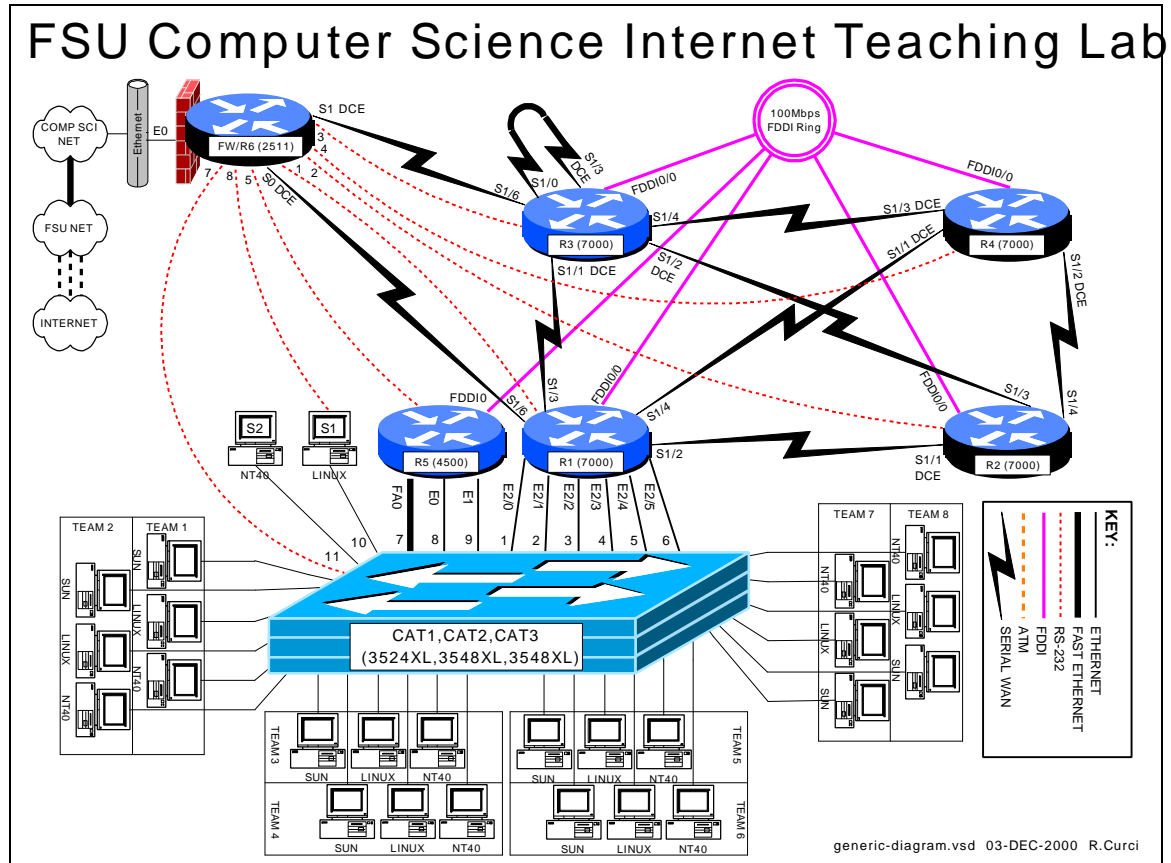
LAB10: Frame-Relay

frame-student.pdf
frame-diagram.pdf
frame-pvc.pdf

LAB11: Multiprotocol

multi-student.pdf
multi-diagram.pdf

INTERNET TEACHING LAB: CISCO ROUTER BASICS



OVERVIEW

In this lab, we will explore some of the basic information on how to configure a Cisco router. In particular, we will see how to access the FSU Computer Science Internet Teaching Lab routers through the Cisco 2511 firewall router, also known as R6. From that router, we will use a feature called “inverse telnet” to access other lab routers through external RS-232 cables. We will also explore some of the router modes including user mode, enable mode, global configuration mode, and sub configuration mode. For additional information you can access the Cisco IOS manuals online at <http://www.cisco.com>. (From the Cisco home page, choose *Technical Documents*→*Documentation Home Page*→*Cisco IOS Software Configuration*→*Cisco IOS Release 11.1*→*Cisco IOS Configuration Guides and Command References*).

BACKGROUND

The ITL lab consists of six Cisco routers labeled R1, R2, R3, R4, R5, and R6; three Cisco catalyst 3500XL series ethernet switches, and several PCs. Cisco routers run an operating system called Cisco IOS or Cisco Internetwork Operating System. Inside the

lab network, devices are numbered using IP private address space documented in the RFC1918 standard. Usually, the lab devices are numbered with the block of class C IP networks from 192.168.1.0/24 through 192.168.254.0/24. (If you are unfamiliar with the “/24” notation, it simply indicates the length of the subnet mask. For example, “/24” indicates a network mask of 255.255.255.0.) Routers R1, R2, R3, R4, and R5 are programmed by students to implement a series of lab exercises to learn about networking. Router R6 also called the “firewall” provides security and connects the lab network to the Computer Science departmental network and Internet. Only limited access is granted to students on this router to prevent changes that might compromise the integrity of the firewall. The firewall uses access lists to selectively block traffic on its ethernet interface. In particular, TELNET access is only permitted when originating from the FSU Computer Science departmental server XI.CS.FSU.EDU. Since the private IP address space is unknown on the Internet backbone, even without these access lists, the lab devices would be unreachable from the Internet. The firewall also performs another important function called “network address translation” or NAT. NAT is configured such that IP packets originating from the lab network will be translated where the source IP address of the packet is replaced by the R6 ethernet address so that it will be globally routable. When the destination server responds, R6 performs the translation in reverse. When enabled, this will allow PCs inside the lab network to access devices outside the lab when communication is initiated from inside the lab only. This will allow you to do things like download files with a web browser on the lab PCs from outside servers. For more background information, see the paper entitled “FSU Computer Science Internet Teaching Lab” which can be found at <http://www.cs.fsu.edu/~curci/itl>.

PART1 – Log into the Cisco 2511:

The Cisco 2511 firewall access router labeled R6 can be accessed in any of 3 ways:

1. Dumb Terminal or Terminal Emulator configured for 9600 baud and DEC VT100 emulation connected the router’s RS-232 console port.
2. TELNET to ethernet interface E0 from XI.CS.FSU.EDU.
3. TELNET to any router R6 interface from inside the lab network. (Only works when the lab routers are configured to provide connectivity.)

We will use the second method. TELNET from XI.CS.FSU.EDU to the R6 interface E0 will allow you to log into router R6. You can TELNET either using the DNS name ITL1.CS.FSU.EDU or the IP address 128.186.121.88. Access lists on interface E0 will allow access only from XI.CS.FSU.EDU, so you will not be able to TELNET in from any other system outside the lab network. When you are connected, the router prompts you for the user mode password that should have been given to you by your instructor. You will also want to enter the command “enable 2” to increase your security level which will enable some commands otherwise not allowed in the user mode.

```
xi% telnet itl1
Trying 128.186.121.88...
Connected to itl1.
Escape character is '^]'.

User Access Verification

Password: xxxxxxx
fw/r6>enable 2
Password: xxxxxxx
fw/r6#
```

Note that the boldface type above indicates the part that you must type, although you should substitute the password for the “xxxxxx”.

Note on enable levels:

Cisco routers have 16 privilege levels called “enable levels” numbered 0 through 15. Level 0 has the least privilege and cannot make any changes and is also called “user mode”. Level 15 is the most privileged and can make any changes and is often simply called “enable mode”. Intermediate levels are used to provide access between the two extremes. For example, in user mode you cannot list the startup configuration or change the configuration. However, you can set up an intermediate level that allows viewing the startup configuration but does not allow changing the configuration. That is what we have done on the firewall/R6 router with enable level 2. This prevents you from making changes to R6 but allows you to at least view the configuration to see what is going on. The command “enable X” prompts for a password and if accepted, changes to enable level X. If X is omitted, 15 is assumed. On the routers you will program, R1 through R5, we will only use enable levels 0 and 15 and refer to them as “user mode” and “enable mode”. Note that the command prompt changes between these two modes-- “user mode” has the “>” symbol while enable mode has the “#” symbol.

The RS-232 console ports on routers R1 through R5 connect to ports Line1 through Line5 on the 2511 respectively. You can connect to any of these routers across the RS-232 link by typing their name unless there is someone else already using the line. This feature is called “inverse telnet”. You can see if anyone else is logged into the firewall with “show user”. You can see any existing sessions you have with “show session”. Once connected to one of these lines, any characters you type are sent across the RS-232 link to the corresponding router and output from the router is displayed on your screen. The only exception is the special escape sequence that brings you back to router R6 – **SHIFT-CONTROL-6-x**. On your keyboard, press and hold the SHIFT key, press and hold the CONTROL key, then press the “6” key. Release all keys, then press “x”. You should now be back on router R6. The command “show session” will show you which sessions you have active. You can go back to your previous session by simply hitting return, or entering the integer session number displayed with the “show session” command. The command “clear line X” where X is the integer line number is sometimes necessary to clear an inactive session from an idle user. Here is a capture to demonstrate:

```

fw/r6#show user
      Line      User      Host(s)      Idle Location
      0 con 0      r1           2w4d
* 18 vty 0      idle        00:00:00 128.186.121.41

fw/r6#show session
% No connections open
fw/r6#r1
Trying r1 (128.186.121.88, 2001)... Open

r1# ← (RETURN and SHIFT-CONTROL-6-x typed here)
fw/r6#r2
Trying r2 (128.186.121.88, 2002)... Open

r2# ← (RETURN and SHIFT-CONTROL-6-x typed here)
fw/r6#r3
Trying r3 (128.186.121.88, 2003)... Open

r3> ← (RETURN and SHIFT-CONTROL-6-x typed here)
fw/r6#show session
Conn Host      Address      Byte  Idle Conn Name
  1 r1          128.186.121.88    0    0 r1
  2 r2          128.186.121.88    0    0 r2
* 3 r3          128.186.121.88    0    0 r3

fw/r6#clear line 3
[confirm]y [OK]
fw/r6#logout
(You have open connections) [confirm]y
Closing: r1 !
Closing: r2 !
Closing: r3 ! Connection closed by foreign host.
xi%

```

Since only one person can use an RS-232 line at a time, if your network is already functional, it may be better to use TELNET from R6 to any of the other lab routers or PCs. By default, Cisco routers allow a maximum of 5 concurrent inbound TELNET sessions.

```

fw/r6#telnet 192.168.55.5
Trying 192.168.55.5 ... Open

User Access Verification

Password: xxxxxxx
r5>enable
Password: xxxxxxx
r5#logout

```

Once logged into your team router go to enable mode. Use the command “show version” to see your router’s IOS version number and operating system image filename. A baseline router configuration file should be located on your router’s flash memory device on a file named “base-rX.cfg” where X is the integer ID corresponding to your router. You can also find a listing of the baseline configuration at the end of this document. Get a directory on your flash filesystem with the command “dir flash:” and verify that the baseline configuration file is present. View this file with “show file flash:baserX.cfg” If everything looks right, copy the baseline configuration file to your router’s startup

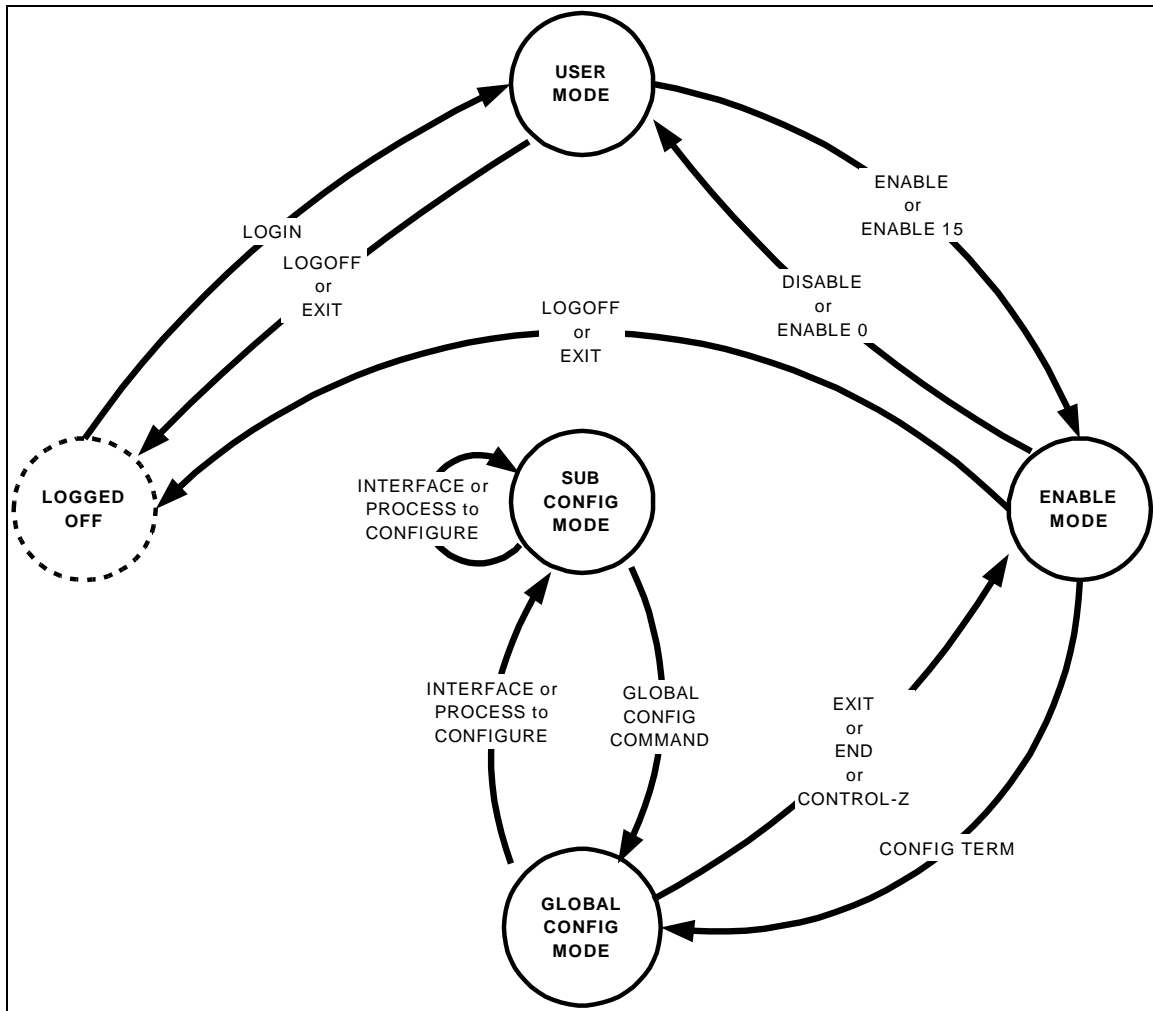
configuration with “copy flash:base-rX.cfg startup-config” and reboot with the new configuration using the “reload” command. Follow these steps carefully. After the last step, your router will take about 3 minutes to r eboot. The following is an example of these steps on router R3 with some of the unimportant messages removed:

```
xi% telnet it11.cs.fsu.edu
Trying 128.186.121.88...
Connected to it11.
User Access Verification
Password: xxxxxx
fw/r6>en 2
Password: xxxxxx
fw/r6#r3
Trying r3 (128.186.121.88, 2003)... Open
r3#enable
r3#show version
Cisco Internetwork Operating System Software
IOS (tm) GS Software (GS7-J-M), Version 11.1(24), RELEASE SOFTWARE (fcl)
r3 uptime is 2 days, 2 hours, 47 minutes
System restarted by power-on
System image file is "gs7-j-mz.111-24.bin", booted via flash
cisco RP1 (68040) processor (revision A0) with 65536K bytes of memory.
...
r3#dir flash:
-#- -length- ----date/time----- name
1  4025994  --- -- ---- -:--:-- gs7-j-mz.111-24.bin
2  1289    --- -- ---- -:--:-- base-r3.cfg
165776 bytes available (4028528 bytes used)
r3#show file flash:base-r3.cfg
version 11.1
service udp-small-servers
service tcp-small-servers
!
hostname r3
...
r3#copy flash:base-r3.cfg startup-config
Warning: distilled config is not generated
[OK]
r3#reload
Proceed with reload? [confirm]
%SYS-5-RELOAD: Reload requested
System Bootstrap, Version 5.0(5), RELEASE SOFTWARE
RP1 processor with 65536 Kbytes of main memory
Reading gs7-j-mz.111-24.bin from flash memory
...
Press RETURN to get started!
r3>
r3>enable
Password: xxxxxx
r3#
```

PART2 – IOS MODES:

The Cisco IOS software can operate in four modes:

1. User Mode
2. Enable Mode
3. Global Configure Mode
4. Sub Configure Mode



The diagram above shows you how to switch between router modes. The following example shows logging into a router (user mode), using the “enable” command to go to enable mode, and using the “configure terminal” command. I then enter a simple configuration to assign an IP address on two interfaces and enable the RIP routing protocol. Note how the command prompt changes as we change between modes. Whitespace is ignored, so I have added whitespace in front of the sub config mode commands for clarity. Note also that a command prefixed with the word “no” negates the meaning of the command such as “shutdown” and “no shutdown”.

Configuration to be entered:

```
ip classless
interface ethernet2/0
  ip address 192.168.10.1 255.255.255.0
  no shutdown
interface ethernet 2/1
  ip address 192.168.20.1 255.255.255.0
router rip
  network 192.168.10.0
  network 192.168.20.0
no ip domain-lookup
```

Here is the captured session:

```
fw/r6#telnet 192.168.11.1
Trying 192.168.11.1 ... Open
User Access Verification

Password: xxxxxx
r1>enable
Password: xxxxxx
r1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
r1(config)#ip classless
r1(config)#interface ethernet2/0
r1(config-if)#ip address 192.168.10.1 255.255.255.0
r1(config-if)#no shutdown
r1(config-if)#interface ethernet2/1
r1(config-if)#ip address 192.168.20.1 255.255.255.0
r1(config-if)#no shutdown
r1(config-if)#router rip
r1(config-router)#network 192.168.10.0
r1(config-router)#network 192.168.20.0
r1(config-router)#exit
r1(config)#no ip domain-lookup
r1(config)#exit
r1#logout
```

When entering commands, you need only enter enough letters for it to be unique. For example, you can use “config t” in place of “configuration terminal”. You can also type the question mark “?” at any point to see your options. If your terminal emulates a DEC VT100, you can also use the UP, DOWN, LEFT, and RIGHT arrow keys to recall previous commands and edit them. Here is a session capture that makes the same router configuration as shown above but demonstrates using abbreviated commands and the built-in “?” HELP facility.


```

fw/r6#telnet 192.168.11.1
Trying 192.168.11.1 ... Open

User Access Verification

Password: xxxxxxx
r1>en
Password: xxxxxxx
r1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
r1(config)#ip clas?
classless

r1(config)#ip classless
r1(config)#int e2/0
r1(config-if)#ip add 192.168.10.1 255.255.255.0
r1(config-if)#no shut
r1(config-if)#int e2/1
r1(config-if)#ip add 192.168.20.1 255.255.255.0
r1(config-if)#no shut
r1(config-if)#router rip
r1(config-router)#net 192.168.10.0
r1(config-router)#net 192.168.20.0
r1(config-router)#exit
r1(config)#no ip d?
default-gateway default-network dhcp-server domain-list domain-lookup
domain-name      dvmrp

r1(config)#no ip domain?
domain-list domain-lookup domain-name

r1(config)#no ip domain-lookup
r1(config)#^Z
r1#lo

```

Log into your router and modify the configuration to display a login message that says “Team X Router” replacing X with your team number using the “banner login” command. Also change your router’s command prompt from “rX” to “teamX” using the “hostname” command. Use the “show interface loopback0” and “show running-config” to view the configuration on your loopback0 interface. Delete your router’s loopback0 interface with “no interface loopback0” Verify it is gone with “show running-config”. Then put the interface back in with “interface loopback0” Make sure you remember to assign the interface an IP address and make sure it NOT shutdown. Since we have not saved any configuration changes in this part, if you get stuck, you can always use the “reload” command to reboot which will undo any changes you have made. Just remember that if you are prompted to save change, you should answer “NO”.

PART3 – Saving and Viewing Configurations:

Cisco routers have two configurations, the startup configuration, and the running configuration. Normally, when a router is booted, it reads in the startup configuration which is stored in flash memory. Once the router is running, the current configuration in RAM is called the running configuration. If no changes are made after booting, both the startup and running configurations will be the same. You can make changes interactively

to the running configuration. You can also commit the changes to the startup configuration in flash or reboot which will cause any changes you have made to be lost. Here are the relevant commands:

- **show startup-config**
List the startup configuration in flash to the screen.
- **show running-config**
List the running configuration currently executing in RAM to the screen.
- **copy running-config startup-config**
Copy the currently running configuration to the startup configuration in flash to commit any changes you have made. The committed changes will persist even after rebooting the router.
- **terminal length 24**
Set the router to pause every 24 lines when displaying messages larger than 24 lines.
- **terminal length 0**
Set the router to not pause when display messages, no matter how long they are even if they scroll off the screen. This is sometimes handy when using a terminal emulator to capture a command with lots of output.
- **reload**
Reboot the router.
- **write erase**
Completely erase the startup configuration. **Use with care!**
- **write**
An old deprecated command that is a synonym for “copy running-config startup-config”
- **write terminal**
An old deprecated command that is a synonym for “show running-config”

Your assignment is to capture your router's running configuration to a text file, erase the startup config and reboot so your router will have no configuration, then get the your text file config back into the router and commit the changes. Afterwards, verify that your router will reboot with the appropriate configuration. Use the following steps to guide you through the process.

1. Log into your router and go to enable mode.
2. Configure your terminal session to inhibit paging.
3. Configure your terminal emulator to capture text.
4. Display the running configuration to your screen while simultaneously capturing it to a text file.
5. Stop capturing text and edit the captured text file with a text editor, removing any extraneous text.
6. Completely erase your router's startup configuration with "erase startup - config"
7. Reboot your router with "reload"
8. After rebooting, you may see an error message indicating that the startup configuration is missing and get prompted by the auto configuration dialog. You should be able to simply press control-C to cancel the dialog.
9. Log into your router, go to enable mode, and list the running configuration to your screen. Compared to your captured text file in step 5 and explain which part of the configuration is still there and which part is missing.
10. Go to global configuration mode and use copy and paste to put the configuration back into your router.
11. List the running configuration and compared to your saved configuration from step 5. How do they differ? Fix any differences so the running configuration is identical to your saved configuration from step 5.
12. Save your changes by copying the running configuration to the startup configuration.
13. Reboot your router and verify it reboots with the correct configuration.
14. Log into your router and go to enable mode. Configure your session to not page every 24 lines. Set your terminal emulator program to capture text. Display the running configuration to your screen while simultaneously capturing to a text file. Get the text file into some text editor and clean up any extraneous text.

PART4 – Miscellaneous Commands:

Read up on the following commands and try them out on your router. Provide a brief explanation of what each does.

1. telnet
2. ping
3. traceroute
4. show version
5. show clock
6. show diagbus
7. show interface
8. show ip interface brief
9. show ip routing
10. show ip protocol

BASELINE ROUTER CONFIGURATION:

For completeness, here is a listing of the baseline router configuration mentioned in part 1 for routers R1, R2, R3, R4 and R5. The section labeled “COMMON:” is needed on all routers. The sections labeled “R1:”, “R2”, etc, are the router specific sections. These configurations should already be present on each router’s flash memory on file “base-rX.cfg” where X is the integer identifier of the router.

COMMON:

```
service udp-small-servers
service tcp-small-servers
enable password cisco
no ip domain-lookup
no ip classless
logging buffered
snmp-server community public RO
line con 0
  exec-timeout 0 0
line aux 0
line vty 0 4
  password cisco
  login
```

R1:

```
hostname r1
interface Loopback0
  ip address 192.168.11.1 255.255.255.0
  no shutdown
interface Fddi0/0
  ip address 192.168.1.1 255.255.255.0
  no shutdown
interface Serial1/2
  description Link to R2 S1/1
  ip address 192.168.12.1 255.255.255.0
  bandwidth 2000
  no shutdown
interface Serial1/3
  description Link to R3 S1/1
  ip address 192.168.13.1 255.255.255.0
  bandwidth 2000
  no shutdown
interface Serial1/4
  description Link to R4 S1/1
  ip address 192.168.14.1 255.255.255.0
  bandwidth 2000
  no shutdown
interface Serial1/6
  description Link to R6 S0
  ip address 192.168.16.1 255.255.255.0
  bandwidth 2000
  no shutdown
interface E2/0
  description Vlan 10 to cat1 FA0/1
  ip address 192.168.10.1 255.255.255.0
  no shutdown
interface E2/1
  description Vlan 20 to cat1 FA0/2
  ip address 192.168.20.1 255.255.255.0
  no shutdown
interface E2/2
  description Vlan 30 to cat1 FA0/3
  ip address 192.168.30.1 255.255.255.0
  no shutdown
interface E2/3
  description Vlan 40 to cat1 FA0/4
  ip address 192.168.40.1 255.255.255.0
```

```
no shutdown
interface E2/4
  description Vlan 50 to cat1 FA0/5
  ip address 192.168.50.1 255.255.255.0
  no shutdown
interface E2/5
  description Vlan 60 to cat1 FA0/6
  ip address 192.168.60.1 255.255.255.0
  no shutdown
router rip
  network 192.168.11.0
  network 192.168.12.0
  network 192.168.13.0
  network 192.168.14.0
  network 192.168.16.0
  network 192.168.1.0
  network 192.168.10.0
  network 192.168.20.0
  network 192.168.30.0
  network 192.168.40.0
  network 192.168.50.0
  network 192.168.60.0
```

R2:

```
hostname r2
interface Loopback0
  ip address 192.168.22.2 255.255.255.0
  no shutdown
interface Fddi0/0
  ip address 192.168.1.2 255.255.255.0
  no shutdown
interface Serial1/1
  description Link to R1 S1/2
  ip address 192.168.12.2 255.255.255.0
  bandwidth 2000
  clockrate 2000000
  no shutdown
interface Serial1/3
  description Link to R3 S1/2
  ip address 192.168.23.2 255.255.255.0
  bandwidth 2000
  no shutdown
interface Serial1/4
  description Link to R4 S1/2
  ip address 192.168.24.2 255.255.255.0
  bandwidth 2000
  no shutdown
router rip
  network 192.168.12.0
  network 192.168.22.0
  network 192.168.23.0
  network 192.168.24.0
  network 192.168.1.0
```

R3:

```
hostname r3
interface Loopback0
```

```

ip address 192.168.33.3 255.255.255.0
no shutdown
interface Fddi0/0
ip address 192.168.1.3 255.255.255.0
no shutdown
interface Serial1/0
description Link to self
no ip address
bandwidth 2000
no shutdown
interface Serial1/1
description Link to R1 S1/3
ip address 192.168.13.3 255.255.255.0
bandwidth 2000
clockrate 2000000
no shutdown
interface Serial1/2
description Link to R2 S1/3
ip address 192.168.23.3 255.255.255.0
bandwidth 2000
clockrate 2000000
no shutdown
interface Serial1/3
description Link to self
no ip address
bandwidth 2000
clockrate 2000000
no shutdown
interface Serial1/4
description Link to R4 S1/3
ip address 192.168.34.3 255.255.255.0
bandwidth 2000
no shutdown
interface Serial1/6
description Link to R6 S1
ip address 192.168.36.3 255.255.255.0
bandwidth 2000
no shutdown
router rip
network 192.168.33.0
network 192.168.13.0
network 192.168.23.0
network 192.168.34.0
network 192.168.36.0
network 192.168.1.0

```

R4:

```

hostname r4
interface Loopback0
ip address 192.168.44.4 255.255.255.0
no shutdown
interface Fddi0/0
description Link to R5 FDDI0
ip address 192.168.1.4 255.255.255.0
no shutdown
interface Serial1/1

```

```

description Link to R1 S1/4
ip address 192.168.14.4 255.255.255.0
bandwidth 2000
clockrate 2000000
no shutdown
interface Serial1/2
description Link to R2 S1/4
ip address 192.168.24.4 255.255.255.0
bandwidth 2000
clockrate 2000000
no shutdown
interface Serial1/3
description Link to R3 S1/4
ip address 192.168.34.4 255.255.255.0
bandwidth 2000
clockrate 2000000
no shutdown
router rip
network 192.168.44.0
network 192.168.14.0
network 192.168.24.0
network 192.168.34.0
network 192.168.1.0

```

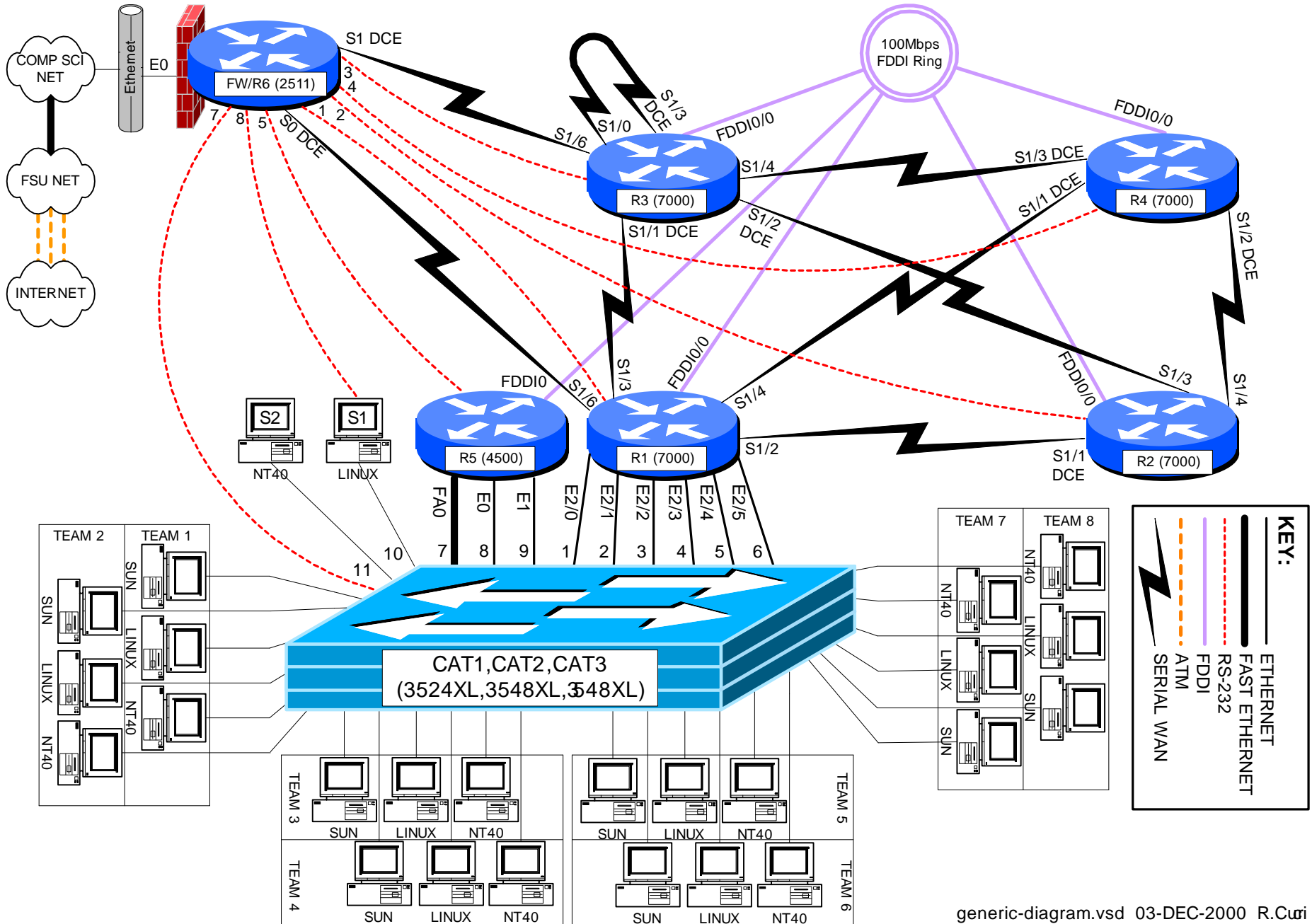
R5:

```

hostname r5
interface loopback0
ip address 192.168.55.5 255.255.255.0
no shutdown
interface FastEthernet0
description Vlan70 to cat1 FA0/7
ip address 192.168.70.1 255.255.255.0
media-type 100BaseX
no shutdown
interface Ethernet0
description Vlan80 to cat1 FA0/8
ip address 192.168.80.1 255.255.255.0
media-type 10BaseT
no shutdown
interface Ethernet1
description Vlan90 to cat1 FA0/9
ip address 192.168.90.1 255.255.255.0
media-type 10BaseT
no shutdown
interface Fddi0
description Link to R4 FDDI0/0
ip address 192.168.1.5 255.255.255.0
no keepalive
no shutdown
router rip
network 192.168.55.0
network 192.168.70.0
network 192.168.80.0
network 192.168.90.0
network 192.168.1.0

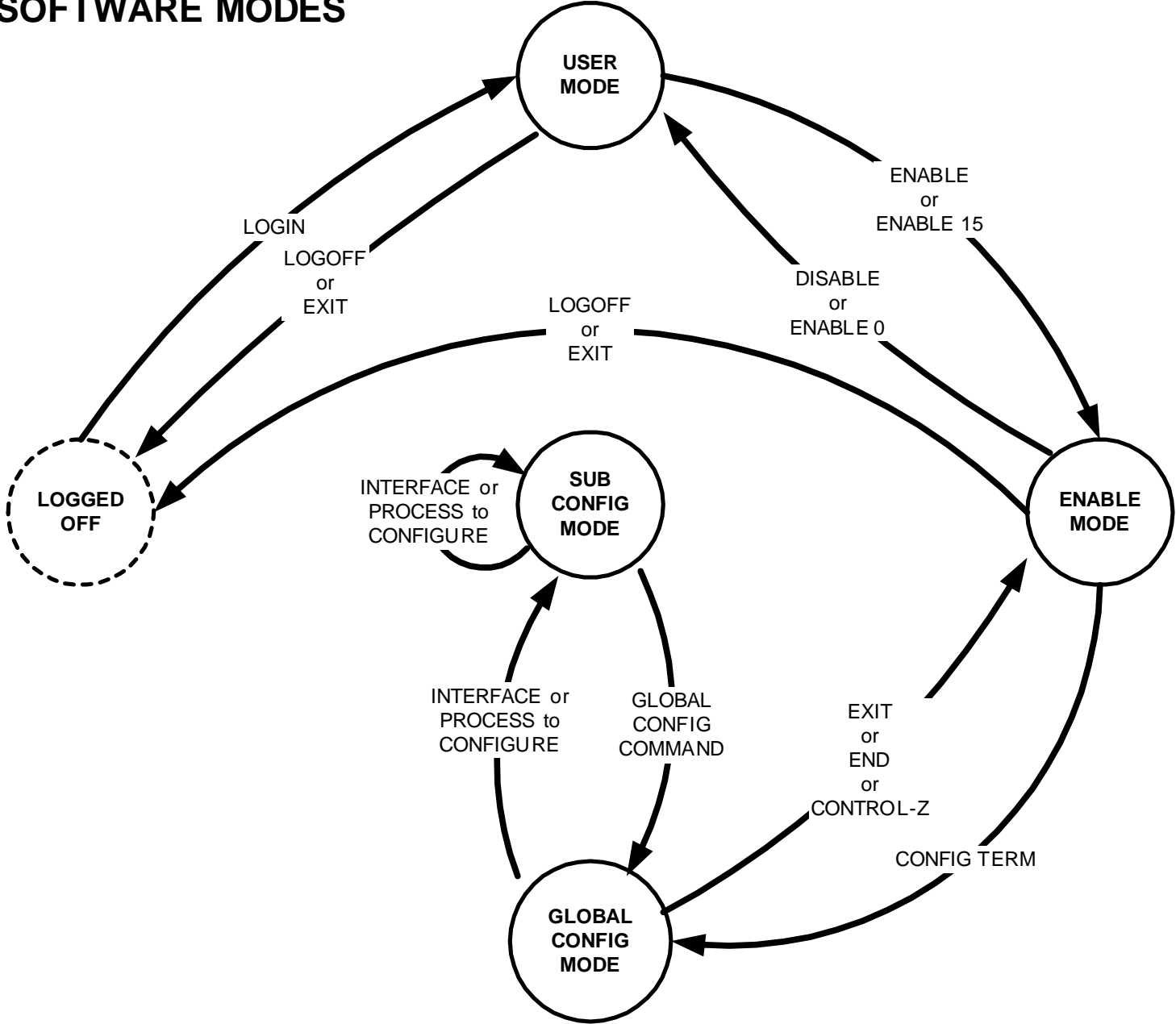
```

FSU Computer Science Internet Teaching Lab

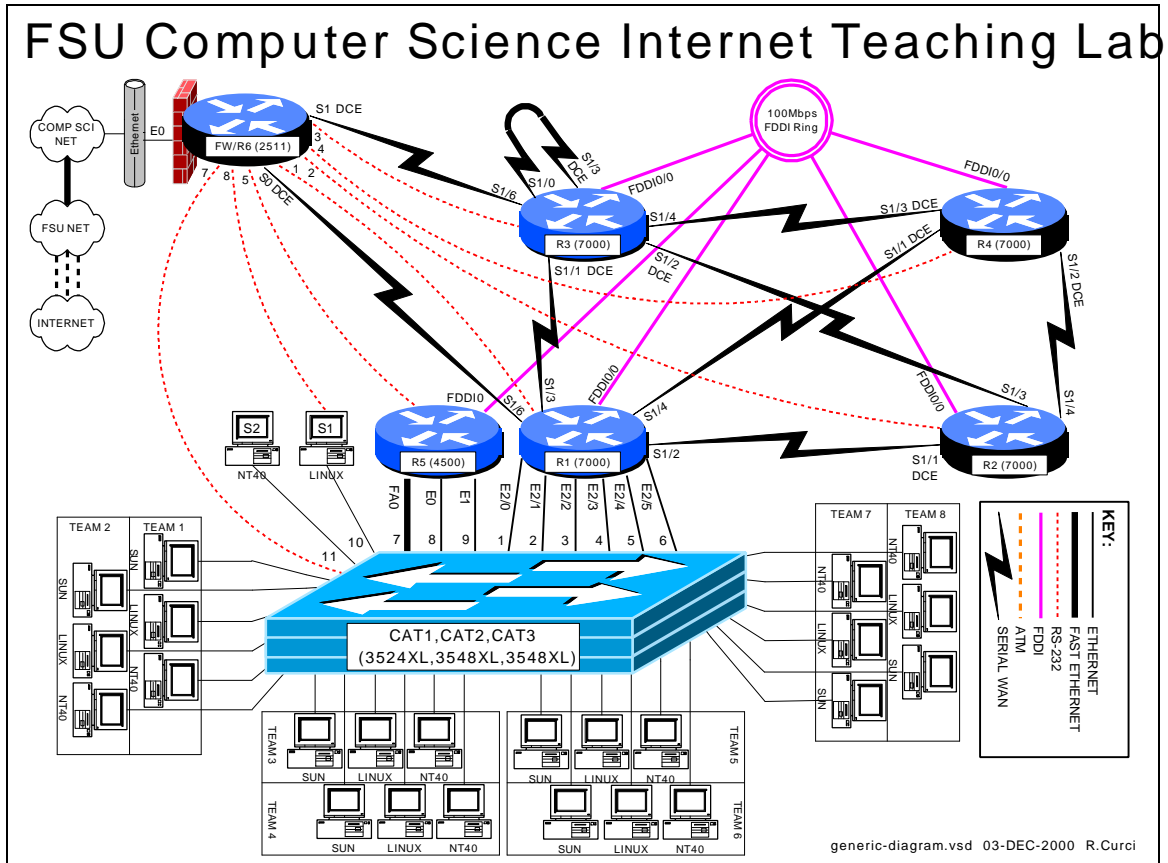


CISCO ROUTER BASICS

CISCO IOS SOFTWARE MODES



INTERNET TEACHING LAB: CISCO ROUTER DEBUGGING



OVERVIEW

Debug mode is a feature of the Cisco IOS software to locate router configuration errors and software bugs. Log messages are similar to debug messages and are generally alerts to problems. You can think of log messages as debug messages that cannot be turned off. Problems are diagnosed by reviewing descriptive messages generated by the router. There are hundreds of different debug options that can be individually turned on and off depending on what part of the system is under examination. It is possible to turn on all debug modes simultaneously, however, this is rarely appropriate as the volume of information would be too voluminous. Debug mode should generally not be used on a production network as it is easy to generate hundreds of error messages per second and cause a router to crash and reboot. We will also explore some of the “show” commands used for debugging problems. **This lab assignment assumes you have the base router configuration from the “Cisco Router Basics” loaded with the RIP routing protocol.** The following is a sample of some debug and log messages. I have removed the timestamps to fit the messages on the page.

(Sample of debug and log messages)

```
r1#term monitor
```

```
r1#debug all
```

```
This may severely impact network performance. Continue? [confirm]
```

```
All possible debugging has been turned on
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/2, changed state to down
```

```
%LINK-3-UPDOWN: Interface Serial1/2, changed state to up
```

```
%SYS-5-CONFIG_I: Configured from memory by console
```

```
%SYS-5-RESTART: System restarted --
```

```
Cisco Internetwork Operating System Software
```

```
IOS (tm) GS Software (GS7-J-M), Version 11.1(24), RELEASE SOFTWARE (fc1)
```

```
%ENVM-2-SUPPLY: Upper Power Supply is Non-Operational
```

```
%LINK-4-FDDISTAT: Interface Fddi0/0, FDDI state c_wrap_b detected?
```

```
IP: s=192.168.16.6 (Serial1/6), d=224.0.0.10, len 64, dispose 31
```

```
SMT I: Fddi0/0, FC=SMT, DA=0000.309c.fb2d, SA=0000.309c.9e3f,
```

```
IP: s=192.168.16.6 (Serial1/6), d=255.255.255.255, len 176, rcvd 2
```

```
UDP: rcvd src=192.168.16.6(520), dst=255.255.255.255(520), length=152
```

```
RIP: received v1 update from 192.168.16.6 on Serial1/6
```

```
0.0.0.0 in 5 hops
```

```
192.168.13.0 in 16 hops (inaccessible)
```

```
192.168.66.0 in 1 hops
```

```
Serial1/2: HDLC myseq 8, mineseen 8*, yourseen 11, line up
```

```
RIP: sending v1 update to 255.255.255.255 via Serial1/2 (192.168.12.1)
```

```
default, metric 6
```

```
network 192.168.66.0, metric 2
```

```
RIP: Update contains 21 routes
```

```
RIP: Update queued
```

```
RIP: Update sent via Serial1/2
```

```
CDP-PA: Packet received from cat1 on interface Ethernet2/0
```

```
r1#undebug all
```

PART 1 – SHOW COMMANDS:

Although not technically debug commands, there are several “show” commands that are helpful with debugging and worth mentioning. Read about the following “show” commands using either the hardcopy Cisco manuals or online manuals at www.cisco.com and try them out on your router. Include a brief description what each of these commands does for your assignment:

1. show version
2. show controller [cbus | serial]
3. show cdp neighbors [detail]
4. show interface
5. show ip interface [brief]
6. show ip protocol
7. show memory
8. show processes cpu
9. show diagbus (7000 only)
10. show tech-support

Using information gathered on your router using the above “show” commands, answer the following questions:

1. What IOS software is your router running? What is the filename of the IOS image? How much RAM? FLASH? What is the value of the configuration register? What model CPU does your router have?
2. For each of your router’s serial WAN interfaces, what kind of cable is attached (DTE, DCE, or none)?
3. Which adjacent routers are sending CDP messages to your router? What IOS software version is running on the adjacent CDP routers?
4. What is the MAC address of your router’s FDDI interface?
5. For each of your router’s active interfaces, is IP Split-Horizon enabled?
6. For the RIP protocol running on your router, what are the values of the RIP protocol *update*, *invalid*, *holddown*, and *flush* timers?
7. How much TOTAL, USED, and FREE RAM is in your router?
8. What is the average CPU utilization for the last 5 minutes?
9. On your 7000 router, what card is physically located in slot 0? What is its hardware revision and serial number?

PART 2 – SET THE CLOCK:

Debug messages are often examined on multiple router devices to study the sequence of events. It is often very useful to configure the debug messages to include a timestamp in order to correlate events in different log files. Setting the router clock is important to make the correlation possible. The current system clock can be displayed with the “show clock” command and set with the “clock set” command. Like UNIX, the Cisco router internally maintains the time as a long integer indicating the number of seconds that have elapsed since January 1st, 1970 GMT (Greenwich Mean Time). Sometimes GMT is called UTC (Universal Time Coordinated). By setting the appropriate time zone, number of hours offset from UTC, and daylight savings time information, the router can display the correct local time. Configure your router’s time zone and daylight savings time information. Configure so that your router will display the local time appropriately and adjust automatically between standard time and daylight savings time. Manually set your router’s clock.

PART 3 – NETWORK TIME PROTOCOL:

In the previous section, we saw how to manually set the router clock and timezone information. Sometimes it is helpful to automatically keep the clocks in sync or synchronize them more accurately than can be done manually. Cisco routers include software that implements the NTP (Network Time Protocol) version 3. NTP can typically maintain the clock accuracy within a few milliseconds. NTP devices maintain relationships with other NTP devices such as “master”, “client”, and “peer”. Each NTP device has a stratum number which indicates the clock’s accuracy and believability. We

will configure routers R1, R2, R3, R4, and R5 as NTP clients of router R6, a stratum 4 NTP server. Configure your router to be an NTP client of NTP server R6. Verify that your clock is synchronized using the “show ntp status”, “show ntp associations”, and “show ntp associations detail” commands. A full discussion of NTP is beyond the scope of this document, however, additional information can be found at <http://www.eecis.udel.edu/~ntp/>.

PART 4 – TIMESTAMPS:

Timestamps can be prepended to debug or log messages. A timestamp can be either an indication of the uptime (how much time has elapsed since the router was booted) or the current date and time. The date and time can be in UTC or the local timezone. Optionally, the timezone and/or the number of milliseconds can be included. Configure your router so that timestamps for both DEBUG and LOG messages will display the local time including the timezone and millisecond information. Verify that it is working.

PART 5 -- OUTPUT OPTIONS:

Debug and log messages generated have three different modes of output: (1) console screen, (2) internal circular buffer, or (3) syslog server.

1. Console Screen

Using the console screen is probably the simplest way to view messages as they are generated. The command “term monitor” enables the display of messages while “term no monitor” inhibits the messages.

2. Internal Circular Buffer

Part of a router’s RAM memory can be allocated to be a circular logging buffer using the configuration command “logging buffer XXXX” where XXXX indicates the size of the buffer. The contents of the buffer can be displayed with the “show logging” command.

3. Syslog Server

A syslog server is a TCP/IP service that accepts log messages and appends them to log files. Both UNIX and NT server systems can be configured as syslog servers. Syslog servers can be used to centralize the collection of messages from many systems to ease system administration. Syslog uses the concepts of facility and severity level. Facility classifies the messages by subsystem to allow the server to append the proper log file. The severity level provides an indication of the importance of an error message where the system manager can set a severity level threshold on both the router and syslog server. On the router, messages with lower priority than the threshold are never sent to the syslog server. A threshold set on the syslog server indicates the minimal importance necessary for a message

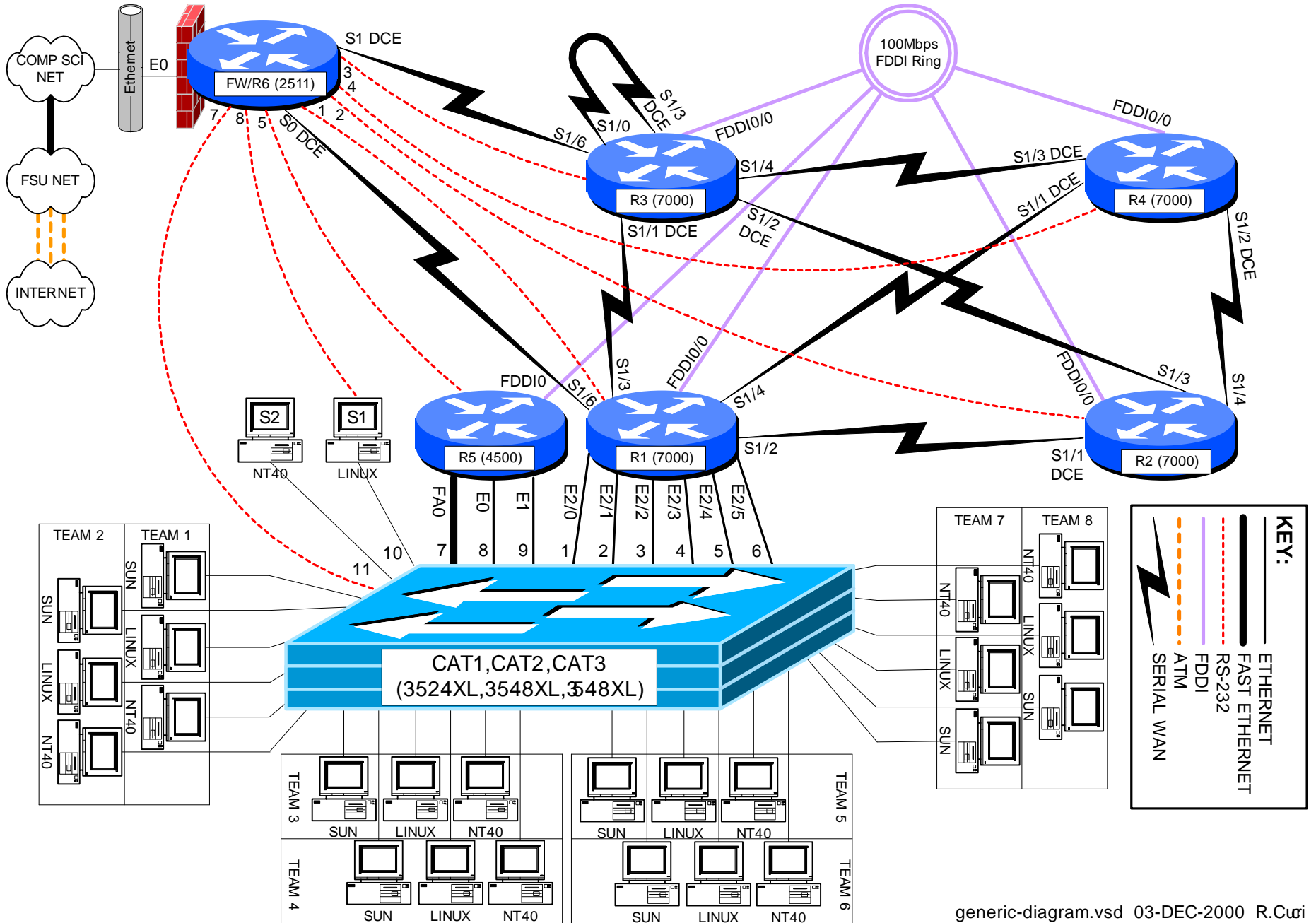
to be logged to a file which is otherwise discarded. By default, Cisco routers use the syslog facility “local7” and severity “informational”, but these parameters are adjustable. Severity “informational” will send more messages except those with severity “debug”. In this part, we will use severity level “debug” so that all messages are important enough to be forwarded from the router to the syslog server and all will be logged by the syslog server.

Configure your router so that debug messages will be logged to three different locations (1) to the console screen, (2) to the internal circular buffer, and (3) to your Linux system using facility “local7” and “severity debug”. Your Linux server should append the messages to file /var/log/cisco.log. We will work on generating messages in the next part.

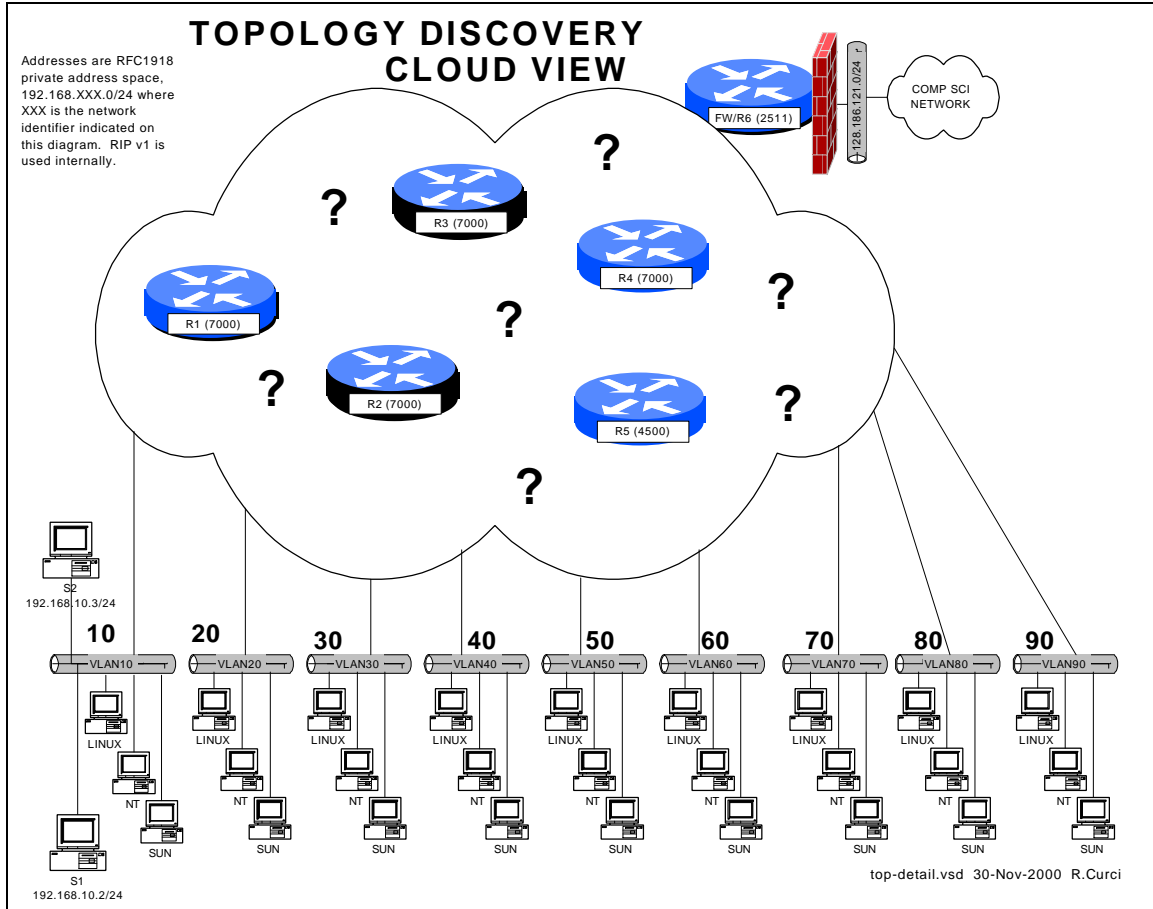
PART 5 – DEBUG MODE:

The command “debug” is used to enable the various debug modes. You can see the options with “debug ?”. Each debug mode can be individually enabled or disabled using “debug xxxxx” to turn on a mode or “no debug xxxxx” to turn one off. The command “show debug” displays which debug modes are currently enabled. You can use the command “debug all” to turn on all debug modes, but it is generally not useful as it can generate hundreds of messages per second. You can turn off all debug modes with the command “no debug all” or “undebug all”. Turn on icmp debugging “debug ip icmp” and ping one of your router’s interfaces. Turn off debugging. Review the messages on your console screen, in your circular buffer, and on your syslog server’s /var/log/cisco.log file. Are the entries identical? If not, explain what is different

FSU Computer Science Internet Teaching Lab



INTERNET TEACHING LAB: TOPOLOGY DISCOVERY



Overview

Each team has a set of computers on its own ethernet VLAN connected to a working lab network of six Cisco routers. Your task is to configure the TCP/IP protocol on your computers and verify you can communicate between your PCs, with other team PCs, the routers, and systems outside the lab on the Internet. Once you have basic connectivity, your job is to download and use network tools to discover the IP addressing scheme and network topology of the lab network. You will be given hints but no login access to the routers for this assignment. Your journal should detail how you discovered different aspects of the addressing scheme and topology and include a detailed network diagram showing the routers, connections between routers, bandwidth of the connections, and IP addresses for all router interfaces, including where the team VLANs attach to the lab network.

Hints

For this exercise inside the lab network, we will be using RFC1918 private address space for all router and computer interfaces. The only exceptions are a single real address on

router FW/R6 which is performing network address translation (NAT) to allow lab computers to access the Internet for downloading files, and a special UNIX server with two ethernet ports. The special UNIX server does not route, but you can TELNET in from outside the lab, then TELNET to your team computers, allowing indirect access.

The six Cisco routers are running the IOS operating system and intentionally have many of the security features disabled to make your job easier. The routers connect to each other through different physical network media at different bandwidths. All layer 3 networks use a 24 bit network mask. Several router features that might normally be disabled have been turned on such as “service tcp-small-servers”, “ip directed-broadcast”, and “ip source-route”. SNMP is enabled on all routers with a read-only community string “public”.

PART1 – GETTING STARTED:

Address your team computers using the following table by replacing TEAM with your integer team number:

LINUX	192.168.X.Y	$X = 10 * \text{TEAM}$	$Y = X + 1$
NT	192.168.X.Y	$X = 10 * \text{TEAM}$	$Y = X + 2$
SOLARIS	192.168.X.Y	$X = 10 * \text{TEAM}$	$Y = X + 3$

For example, team 5’s NT system should be addressed with 192.168.50.52/24.

To test basic connectivity, verify you can PING(1) each of the other teams’ local gateway IP address.

On each of your computers, install the RIP version 1 routing protocol. Under UNIX, you can use either GateD or Routed in passive mode. Under NT 4.0, use “RIP for Internet Protocol”. Your computers should learn a list of routes including a special “default route” sometimes abbreviated “0.0.0.0”. Make sure you have removed any static default routes and are learning the default dynamically. Build a table of routes including the RIP metric. This metric indicates the number of router hops from your computer to each of the networks and will help in figuring out the topology.

Hint: The UNIX utility ripquery(1) may be helpful.

PART2 – FIND THE SIX ROUTERS:

Given the network list from part 1, PING(1) the broadcast address for each network you found above. Normally, you will hear responses from the IP address of the router interface closest to your computer connected to the destination network. If you see more than one IP address in the responses, it is an indication that there are multiple routers on the broadcast network with different paths back to your computer.

Use the TRACEROUTE(1) utility to find some of the connections between the routers. (This utility is named TROUTE.EXE under NT). For each lab network, select an IP address and trace the route to it, making a note of the IP addresses of the routers in the path. Be sure to trace the route toward the Internet by tracing to a computer science server outside the lab. You should be able to find an IP address for each of the six routers. Note that routers generally have multiple interfaces each with its own IP address, so you may find multiple IP addresses that belong to the same router.

Download install the NMAP(1) utility for UNIX. You can find it at www.insecure.org/nmap. Use this tool to scan the 192.168.0.0/16 address space to find all devices and attempt to guess their operating systems. Be careful not to scan outside the 192.168.0.0/16 lab network as most System Administrators treat scanning as an attack and will likely trigger many intrusion detection alarms. Under Florida Law, port scanning is treated as unauthorized intrusion.

PART4 – Simple Network Management Protocol (SNMP)

All of the lab routers will respond to SNMP version 1 queries. SNMP version 1 uses a simple password protection scheme called a “community”. Each router is programmed to be an SNMP “agent” and will respond to the read-only community string “public”. SNMP agents store data in a Management Information Base, or MIB. The MIB contains a lot of information including the router name, the uptime, software version, interface names, interface IP addresses, routing tables, etc. Many SNMP tools are available for Linux:

```
snmpbulkget      snmpget          snmpset          snmptest         snmpusm
snmpbulkwalk    snmpgetnext      snmpstatus       snmptranslate    snmpwalk
snmpdelta       snmpnetstat      snmpmtable       snmptrap
```

For example, you can display individual MIB variables with SNMPGET(1):

```
LINUX$ snmpget 192.168.10.1 public system.sysDescr.0
system.sysDescr.0 = Cisco Internetwork Operating System Software
IOS (tm) GS Software (GS7-J-M), Version 11.1(24), RELEASE SOFTWARE (fcl)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Mon 04-Jan-99 21:19 by richv
LINUX$
```

For each of your routers, look up the following MIB variables:

system.sysDescr.0

system.sysName.0

This will let you see the router names to eliminate any duplicates if you previously found more than one IP address for the same router. Using the system description, note the IOS software version of the router. You should now have enough information to draw a diagram of the six routers with the interface names, interface types (ethernet, fddi, point-to-point, loopback), how they connect to each other, and the IP addressing scheme.

PART5 – Bandwidth Measurement (IPERF/PCHAR):

IPERF(1) is a tcp performance measurement tool. It is an updated version of the Test TCP program (TTCP) written by Terry Slattery in 1985 at the US Navy Ballistic Research Lab. You can find the latest version at <http://dast.nlanr.net/Projects/Iperf/>. You will find both UNIX source code that complies under Linux and SUNOS, and Microsoft Windows executable files (iperf.exe and iperfthreaded.exe). Normally, you start one copy of IPERF(1) in server mode, and the other in client mode specifying the server's IP address. This utility in client mode will also work with an ordinary TCP/IP device supporting the trivial TCP DISCARD service on TCP port 9 which is enabled on all lab routers. Measure the performance from your computer to each router to help determine the bandwidth between links on your network. Note that if the packets traverse several links, the slowest link in the path will be the determining factor.

PCHAR(1) is a utility similar to TRACEROUTE(1), but tries to determine the bandwidth between adjacent hops in the path. It is an updated version of PATHCHAR(1) written by Van Jacobson at Lawrence Berkeley Labs, namesake of the IP Van Jacobson header compression. You will need to either change permissions on PCHAR(1) to be SUID root or execute it while logged in as root. It can be found at <http://www.employees.org/~bmah/Software/pchar/> Be patient with this program as it can take a long time to run using the default settings.

PART6 – Windows NT Network Management / WhatsUp Gold:

Download and install the utility “WhatsUp Gold” on your NT machine. You can download a 30-day evaluation copy from www.ipswitch.com. You will find both a self-installing Win95/98/NT/2000 executable and a users guide in Adobe Acrobat PDF format. If you don't have Adobe Acrobat Reader already loaded, you can find it at www.adobe.com. As of this writing, the latest software is version 5. Test out the following tools and verify the results are consistent with your topology drawing:

- traceroute tool
- snmp tool
- scan tool
- throughput tool

How does the SCAN tool compare to the UNIX NMAP utility?

Run the throughput tool to test each router using both the ICMP and TCP/discard/port -9 modes. How do these measurements compare to each other and to tests you made earlier with the UNIX IPERF utility?

Use this software to create a live map of your network including the six routers and IP networks. Change the polling method to TCP/IP—SNMP since this provides more information than the default ICMP method. Edit the symbols on the map to abbreviate the router names such as “R1” and network names using the third octet of the IP network number. This will help give you more room to fit all the icons on the screen. Configure the system to poll the devices every 10 seconds (Make sure you are not polling any devices outside of the lab environment). If configured properly, you should be able to view the map where the icon color indicates the status (i.e. green=good) and you should also be able to right-click your mouse on the router icons to Telnet, Ping, Traceroute, etc., to the highlighted device.

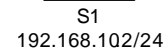
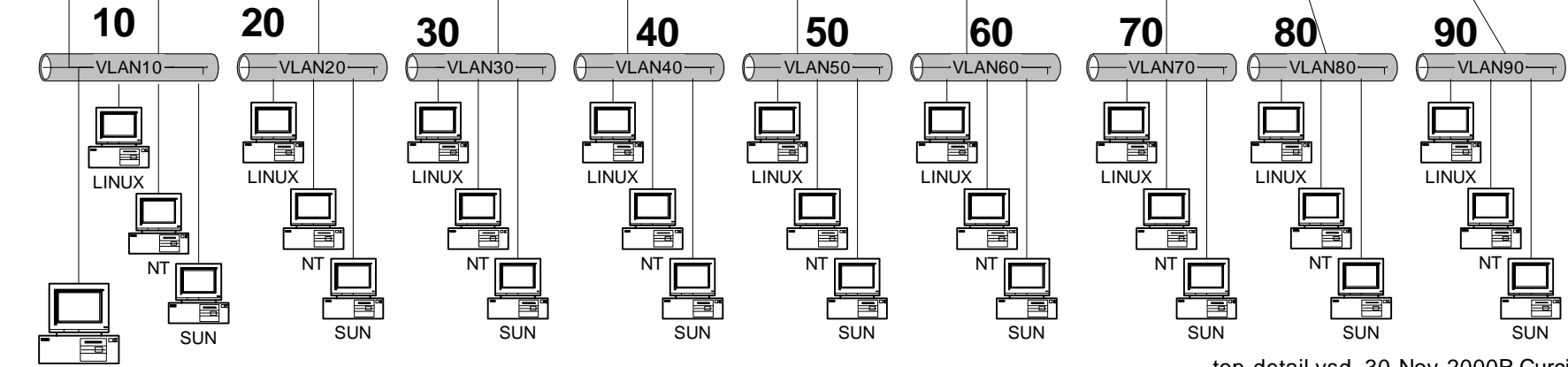
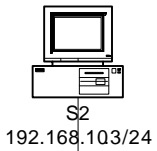
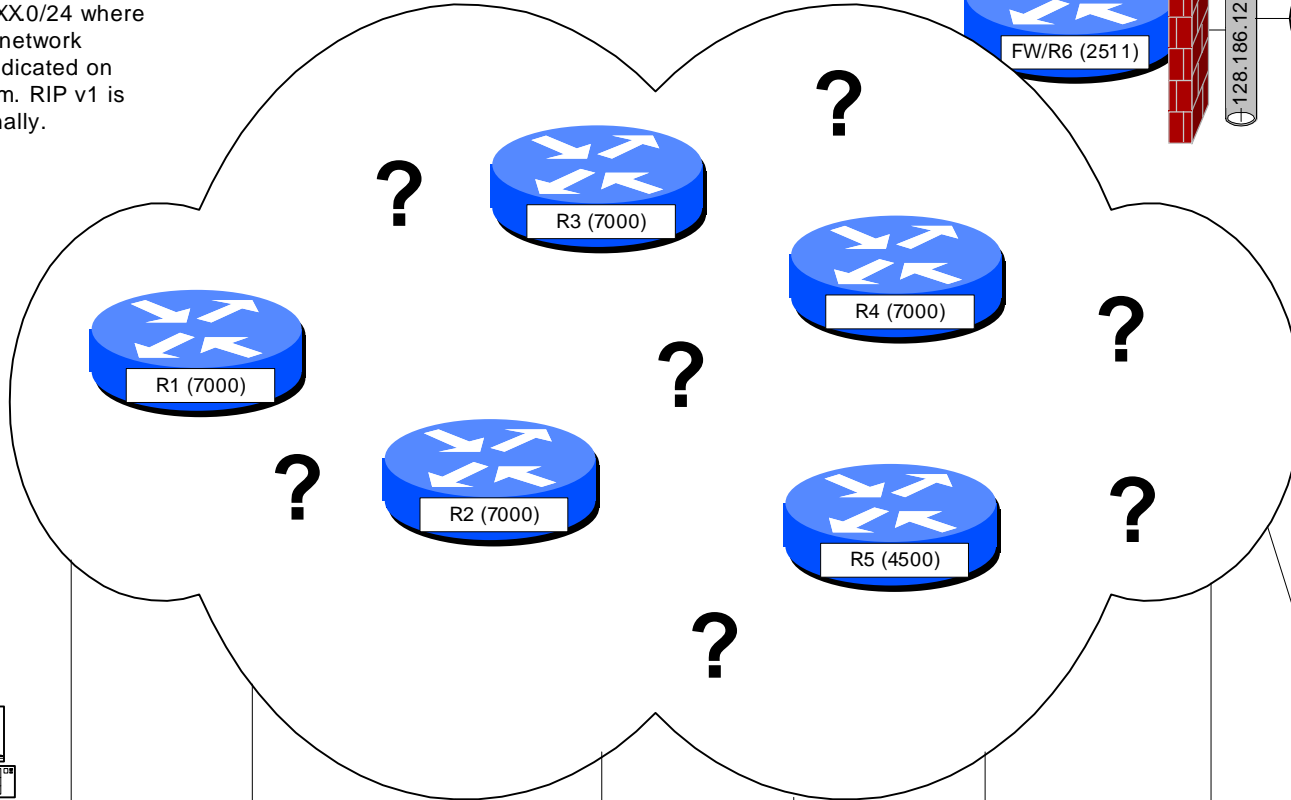
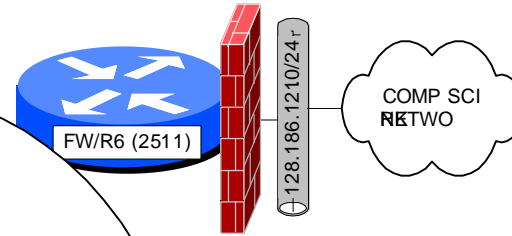
Configure the system such that if any of the routers go down on weekdays between 9am and 5pm, the system will send you an automatic e-mail message.

Configure the system to implement a WWW server such that you can check the status of your network remotely with the use of a web browser.

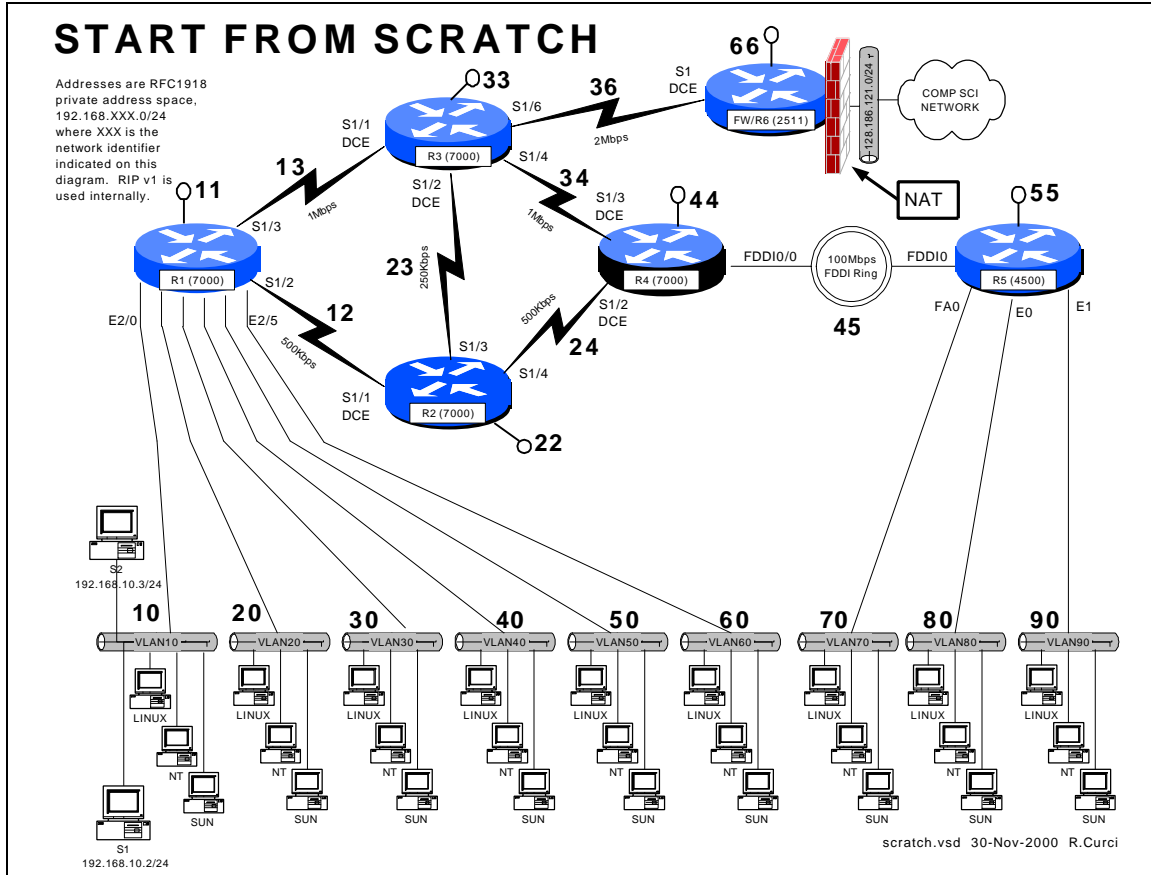
Hint: The “discover devices/intelligently scan network devices with SNMP seed router” function may save you time to initially build your map.

CLOUD VIEW

Addresses are RFC1918 private address space, 192.168.XXX.0/24 where XXX is the network identifier indicated on this diagram. RIP v1 is used internally.



INTERNET TEACHING LAB: START-FROM-SCRATCH LAB



Overview

Your instructor has deleted the configuration on all lab routers except for the firewall/r6 router. Since the lab network is not functional, you will need to access your router by telnetting from xi.cs.fsu.edu to the firewall/r6 router at ITL1.cs.fsu.edu (128.186.121.88). Once logged in, you will need to connect using reverse telnet to access your router's console port to get basic TCP/IP with RIP v1 working. To prove you have successfully completed this assignment, submit a copy of your router's output to the following commands: "show running-config", "show ip interface brief", "show cdp neighbor", and "show ip route".

PART0 – Numbering Convention

Each router is numbered with a small integer. Networks that tie together two routers use a network number composed of the router numbers concatenated with the lower number first. Loopback addresses are numbered with the IP network consisting of the router ID repeated. On network between routers, the last octet of the IP address is the same as the router. On serial connections between routers, the higher numbered router is always the

DCE side which provides the clocking. On PC LAN segments, the router IP addresses use the number have the last octet equal to 1.

PART1 – Out-Of-Band Login

Begin by logging into xi.cs.fsu.edu from a computer on a functional computer network. From xi.cs.fsu.edu, you can telnet to IT1.cs.fsu.edu (128.186.121.88). Once logged in, type the name of your router such as “r1”. Aliases are define to connect to to the appropriate console port. Routers “r1” thru “r5” correspond t o lines “1” thru “5” respectively. If this does not work, you may need to enable security level 2 and clear the line manually with the command “clear line X” where X is the appropriate line. Once connected to your router, you may need to press control -C to abort an auto configuration dialog and hit return:

```
xi% telnet it11
Trying 128.186.121.88...
Connected to it11.
Escape character is '^]'.
User Access Verification
Password:
fw/r6>enable 2
Password:
fw/r6#clear line 1
[confirm]y [OK]
fw/r6#r1
Trying r1 (128.186.121.88, 2001)... Open
User Access Verification
Password:
Router>en
Password:
Router#
```

Use “enable” to put your router in privileged mode to allow you to make changes. Go into configuration mode and add the basic configuration information as shown below. Configuration mode is entered with the command “config term” and exited with control-Z. Notice how the prompt changes to indicate the router mode. The command “show run” displays the running configuration. “term length 24” will make the router page every 24 lines, while “term length 0” will inhibit paging. The running configuration on a router whose configuration has been erased is shown below.

```
Router>enable
Router#term len 24
Router#show running-config
Building configuration...

Current configuration:
!
version 11.1
service udp-small-servers
service tcp-small-servers
!
hostname Router
!
...
```

```

line con 0
line aux 0
line vty 0 4
  login
!
end

```

PART2 – Enter the routine configuration.

There are some configuration parts that will be common to all routers. In this example, we are adding three passwords:

- enable password (like a superuser password)
- console password (used when logging in via RS232 console)
- vty password (used when accessed via TELNET)

Two other handy commands are “no ip domain-lookup” to prevent the router from trying to lookup any typos with DNS, and “exec-timeout 0 0” which disables a login port from logging you out automatically.

```

Router#config term
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#enable password cisco
Router(config)#hostname r1 <----- USE APPROPRATE ROUTER NAME
r1(config)#enable password cisco
r1(config)#no ip domain-lookup
r1(config)#line con 0
r1(config-line)#password cisco
r1(config-line)#login
r1(config-line)#exec-timeout 0 0
r1(config-line)#line vty 0 4
r1(config-line)#password cisco
r1(config-router)#^Z
r1#
%SYS-5-CONFIG_I: Configured from console by console

```

Here is the plain text that you should be able to copy/paste:

```

enable password cisco
no ip domain-lookup
line con 0
  password cisco
  login
  exec-timeout 0 0
line vty 0 4
  password cisco

```

PART3 – Enter the router specific configuration.

Now enter the specific configuration for your router as appropriate below. I have included the “no shutdown” command because interfaces are left in a shutdown state by default.

```

R1:
int loopback0
  ip address 192.168.11.1 255.255.255.0

```

```

no shutdown
int serial1/2
ip address 192.168.12.1 255.255.255.0
no shutdown
int serial 1/3
ip address 192.168.13.1 255.255.255.0
no shutdown
int ethernet 2/0
ip address 192.168.10.1 255.255.255.0
no shutdown
int ethernet 2/1
ip address 192.168.20.1 255.255.255.0
no shutdown
int ethernet 2/2
ip address 192.168.30.1 255.255.255.0
no shutdown
int ethernet 2/3
ip address 192.168.40.1 255.255.255.0
no shutdown
int ethernet 2/4
ip address 192.168.50.1 255.255.255.0
no shutdown
int ethernet 2/5
ip address 192.168.60.1 255.255.255.0
no shutdown
router rip
network 192.168.10.0
network 192.168.20.0
network 192.168.30.0
network 192.168.40.0
network 192.168.50.0
network 192.168.60.0
network 192.168.12.0
network 192.168.13.0
network 192.168.11.0

```

R2:

```

int loopback0
ip address 192.168.22.2 255.255.255.0
no shutdown
int serial1/1
ip address 192.168.12.2 255.255.255.0
clock rate 2000000
no shutdown
int serial 1/3
ip address 192.168.23.2 255.255.255.0
no shutdown
int serial 1/4
ip address 192.168.24.2 255.255.255.0
no shutdown
router rip
network 192.168.12.0
network 192.168.22.0
network 192.168.23.0
network 192.168.24.0

```

R3:

```

int loopback0
ip address 192.168.33.3 255.255.255.0
no shutdown
int serial1/1
ip address 192.168.13.3 255.255.255.0
clock rate 2000000

```



```

no shutdown
int serial 1/2
 ip address 192.168.23.3 255.255.255.0
 clock rate 2000000
no shutdown
int serial 1/4
 ip address 192.168.34.3 255.255.255.0
no shutdown
int serial 1/6
 ip address 192.168.36.3 255.255.255.0
no shutdown
router rip
 network 192.168.13.0
 network 192.168.23.0
 network 192.168.33.0
 network 192.168.34.0
 network 192.168.36.0

```

R4:

```

int loopback0
 ip address 192.168.44.4 255.255.255.0
no shutdown
int serial1/2
 ip address 192.168.24.4 255.255.255.0
 clock rate 2000000
no shutdown
int serial 1/3
 ip address 192.168.34.4 255.255.255.0
 clock rate 2000000
no shutdown
int fddi0/0
 ip address 192.168.45.4 255.255.255.0
no shutdown
router rip
 network 192.168.24.0
 network 192.168.34.0
 network 192.168.44.0
 network 192.168.45.0

```

R5:

```

int loopback0
 ip address 192.168.55.5 255.255.255.0
no shutdown
int FDDI0
 ip address 192.168.45.5 255.255.255.0
no shutdown
int fastethernet 0
 ip address 192.168.70.1 255.255.255.0
 media-type 100baseX
no shutdown
int ethernet 0
 ip address 192.168.80.1 255.255.255.0
 media-type 10baseT
no shutdown
int ethernet 1
 ip address 192.168.90.1 255.255.255.0
 media-type 10baseT
no shutdown
router rip
 network 192.168.45.0
 network 192.168.55.0
 network 192.168.70.0
 network 192.168.80.0

```

```

network 192.168.90.0

R6:
int loopback0
 ip address 192.168.66.6 255.255.255.0
 no shutdown
int serial 1
 ip address 192.168.36.6 255.255.255.0
 clock rate 2000
 no shutdown
router rip
 network 192.168.36.0
 network 192.168.66.0
 default-metric 5

```

PART4 – Test the network.

By default, Cisco routers send out Cisco Discovery Protocol (CDP) packets. As your router hears CDP packets, it maintains a table of adjacent devices. Display your CDP neighbors with the command “show cdp neighbor”. You should see a listing like this if all is working correctly.

```

r1#show cdp nei
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
r2                  Ser 1/2          179        R           RP1       Ser 1/1
r3                  Ser 1/3          149        R           RP1       Ser 1/1
cat1                Eth 2/5          172        T S        WS-C3524-XFas 0/6
cat1                Eth 2/4          172        T S        WS-C3524-XFas 0/5
cat1                Eth 2/3          171        T S        WS-C3524-XFas 0/4
cat1                Eth 2/2          171        T S        WS-C3524-XFas 0/3
cat1                Eth 2/1          171        T S        WS-C3524-XFas 0/2
cat1                Eth 2/0          171        T S        WS-C3524-XFas 0/1

```

```

r2#show cdp nei
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
r3                  Ser 1/3          135        R           RP1       Ser 1/2
r1                  Ser 1/1          164        R           RP1       Ser 1/2
r4                  Ser 1/4          144        R           RP1       Ser 1/2

```

```

r3#show cdp nei
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
r2                  Ser 1/2          151        R           RP1       Ser 1/3
r1                  Ser 1/1          150        R           RP1       Ser 1/3
r4                  Ser 1/4          129        R           RP1       Ser 1/3
fw/r6              Ser 1/6          136        R           2511     Ser 1

```

```

r4#show cdp nei
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

```

```

r2          Ser 1/2          139          R          RP1          Ser 1/4
r3          Ser 1/3          169          R          RP1          Ser 1/4
r5          Fddi0/0         124          R          4500         Fddi0

```

```
r5#show cdp nei
```

```

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

```

```

Device ID      Local Infrfce  Holdtme  Capability  Platform  Port ID
r4             Fddi0         153      R           RP1        Fddi0/0
cat1           Eth 1         168      T S         WS-C3524-XFas 0/9
cat1           Eth 0         167      T S         WS-C3524-XFas 0/8
cat1           Fas 0         167      T S         WS-C3524-XFas 0/7

```

```
fw/r6#show cdp nei
```

```

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

```

```

Device ID      Local Infrfce  Holdtme  Capability  Platform  Port ID
r3             Ser 1         136      R           RP1        Ser 1/6
c2900.cs.fsu.edu Eth 0         179      S           WS-C2924M-Fas 0/2

```

You can display the status of your interfaces with “show ip int brief” for an abbreviated listing, or “show ip int” for a detailed listing. If everything is working, you should have a status of “interface up and line protocol up” on the active interfaces. If you see the status as “administratively down”, it means that your interface is shutdown which can be fixed with a “no shutdown” command issued under the appropriate interface. It is normal for interfaces not used in this lab to be in the default “shutdown” state.

```
r1#show ip int brief
```

```

Interface      IP-Address      OK? Method Status
Protocol
Fddi0/0        unassigned      YES unset  administratively down down
Serial1/0      unassigned      YES unset  administratively down down
Serial1/1      unassigned      YES unset  administratively down down
Serial1/2      192.168.12.1    YES manual  up          up
Serial1/3      192.168.13.1    YES manual  up          up
Serial1/4      unassigned      YES unset  administratively down down
Serial1/5      unassigned      YES unset  administratively down down
Serial1/6      unassigned      YES unset  administratively down down
Serial1/7      unassigned      YES unset  administratively down down
Ethernet2/0    192.168.10.1    YES manual  up          up
Ethernet2/1    192.168.20.1    YES manual  up          up
Ethernet2/2    192.168.30.1    YES manual  up          up
Ethernet2/3    192.168.40.1    YES manual  up          up
Ethernet2/4    192.168.50.1    YES manual  up          up
Ethernet2/5    192.168.60.1    YES manual  up          up
Loopback0     192.168.11.1    YES manual  up          up

```

```
r1#show int ethernet2/0
```

```

Ethernet2/0 is up, line protocol is up
  Hardware is cxBus Ethernet, address is 0000.0c39.dfc4 (bia 0000.0c39.dfc4)
  Internet address is 192.168.10.1/24
  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 00:00:05, output 00:00:05, 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
 278 packets input, 36107 bytes, 0 no buffer
Received 73 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 input packets with dribble condition detected
498 packets output, 103025 bytes, 0 underruns
0 output errors, 0 collisions, 4 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier
0 output buffer failures, 0 output buffers swapped out
```

Verify that everything is working by trying to PING each router IP address from both your router and PC. By default, PING will send 5 ICMP echo packets. If the destination responds, exclamation marks “!” are displayed, otherwise a timeout is indicated by a period “.” Try using the TRACEROUTE utility to trace the path to the other routers. Both the PING and TRACEROUTE commands can be entered without the destination argument to give you extended option choices such as changing the packet size, number of packets, source interface, etc.

```
r1#ping 192.168.11.1
Sending 5, 100-byte ICMP Echoes to 192.168.11.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
r1#ping 192.168.22.2
Sending 5, 100-byte ICMP Echoes to 192.168.22.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
r1#ping 192.168.33.3
Sending 5, 100-byte ICMP Echoes to 192.168.33.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
r1#ping 192.168.44.4
Sending 5, 100-byte ICMP Echoes to 192.168.44.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms
r1#ping 192.168.55.5
Sending 5, 100-byte ICMP Echoes to 192.168.55.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms
r1#ping 192.168.66.6
Sending 5, 100-byte ICMP Echoes to 192.168.66.6, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms

r1#traceroute 192.168.55.5
Type escape sequence to abort.
Tracing the route to 192.168.55.5
 0 192.168.13.2 0 msec
    192.168.12.2 0 msec
    192.168.13.2 0 msec
 1 192.168.24.4 8 msec
    192.168.34.2 4 msec
    192.168.24.4 4 msec
 2 192.168.45.5 4 msec * 0 msec
```

Display the routing table with “show ip route” and verify you have a route to each IP network.

```

r3#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       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
Gateway of last resort is 192.168.36.6 to network 0.0.0.0
R    192.168.70.0/24 [120/2] via 192.168.34.2, 00:00:24, Serial1/4
R    192.168.90.0/24 [120/2] via 192.168.34.2, 00:00:24, Serial1/4
R    192.168.80.0/24 [120/2] via 192.168.34.2, 00:00:25, Serial1/4
R    192.168.40.0/24 [120/1] via 192.168.13.1, 00:00:00, Serial1/1
R    192.168.44.0/24 [120/1] via 192.168.34.2, 00:00:25, Serial1/4
R    192.168.45.0/24 [120/1] via 192.168.34.2, 00:00:25, Serial1/4
C    192.168.33.0/24 is directly connected, Loopback0
C    192.168.34.0/24 is directly connected, Serial1/4
...

```

When you have everything working, save the configuration. Cisco routers have both a running configuration and startup configuration. Issue the command:

“copy running-config startup-config” to save your configuration in non-volatile memory so it will retain the configuration upon rebooting. You should also capture your configuration to a text file on your PC using your terminal emulator’s logging function. The command “show running-config” will display the config to your screen. To prevent the screen from paging every 24 lines, you may want to first set the terminal length to zero, display the config, then set it back to 24 lines.

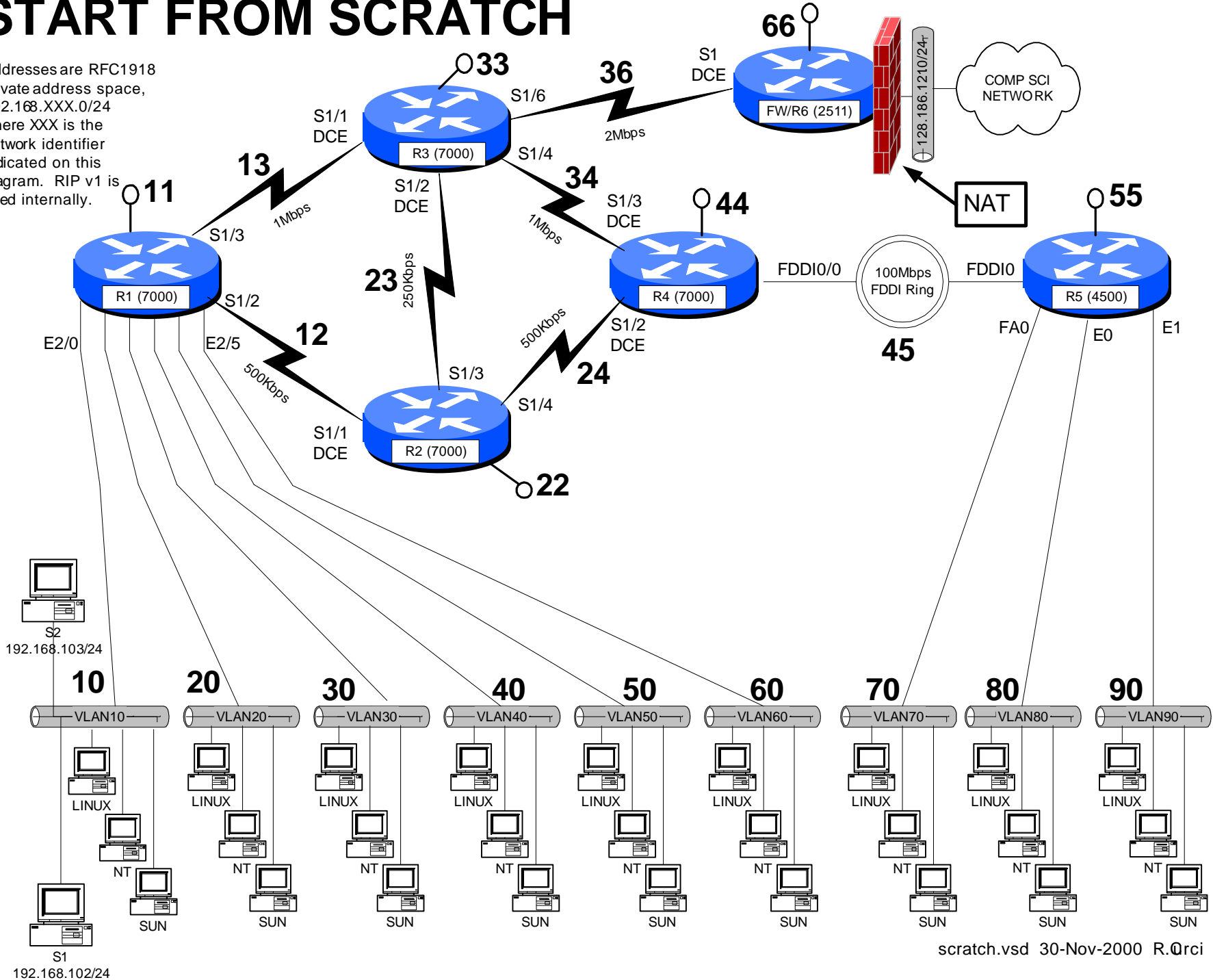
```

Router# term length 0
Router# show running-config
...lots of config displayed here...
Router# term length 24

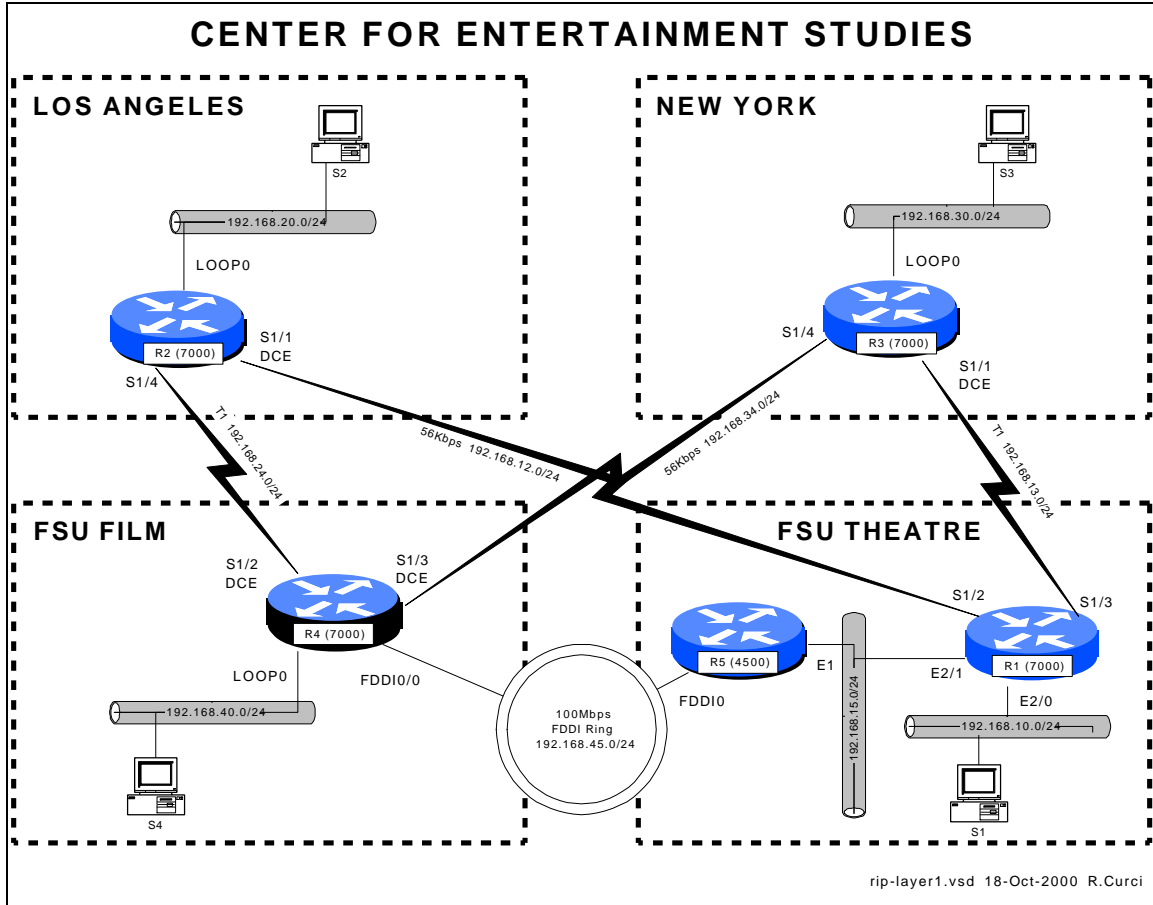
```

START FROM SCRATCH

Addresses are RFC1918 private address space, 192.168.XXX.0/24 where XXX is the network identifier indicated on this diagram. RIP v1 is used internally.



INTERNET TEACHING LAB: ROUTING INFORMATION PROTOCOL



Overview

Governor Bush has just been reelected thanks to an effective TV campaign with the help of FSU faculty from the School of Theatre and Film School. In return he has obtained funding for the new FSU Center for Entertainment Studies which will oversee the Film School and School of Theatre. These two schools will retain their existing office space at separate locations tied together with a 100Mbps FSU FDDI backbone. Theatre is located on the FSU Campus while Film is located at the FSU University Center. In this document, these locations will be referenced as “FILM” and “THEATRE”. Funding has been obtained to expand the program and open branch campuses in Los Angeles and New York City.

You have just been hired as the Network Manager for the Center and your first task is to network your ethernet-based computers at all four geographical locations using the TCP/IP protocol. Your highest bandwidth needs are between “THEATRE” and “FILM”. “NEW YORK” mostly needs to communicate with “THEATRE” while “LOS ANGELES” mostly needs to communicate with “FILM”. All locations must be able to talk with all others, but the major needs are outline above. You have two routers at “THEATRE” and one at each of the other locations. Each site has one router with

available serial ports for connecting WAN circuits. You have a budget of \$7,000 per month for WAN circuit monthly recurring costs and determine the following prices:

MONTHLY RECURRING COSTS			
CITY1	CITY2	56K bps	T1 1.44Mbps
TALLAHASSEE	LOS ANGELES	\$500	\$3,000
TALLAHASSEE	NEW YORK	\$500	\$3,000
LOS ANGELES	NEW YORK	\$500	\$3,000

You decide to buy a T1 from “NEW YORK” to “THEATRE” and a second T1 from “LOS ANGELES” to “FILM”, each terminating on different routers. Since you have \$1000/month left in your budget you decide to spend it on two slower speed 56K circuits: “NEW YORK” to “FILM” and “LOS ANGELES” to “THEATRE”. For extra redundancy, you decide to terminate these backup circuits on different routers on the Tallahassee end as depicted in the wiring diagram. You decide to use the RIP routing protocol and get everything up and running.

Here are your IP address assignments. Note some of the conventions to make addressing a little bit easier. Generally speaking, network masks are /24 unless otherwise specified. Interfaces between routers use the two router numbers in the third octet, i.e. a links from router X to router Y is network 192.168.XY.0 where X is the lower numbered router. Also, on interfaces between routers, the last octet of the address corresponds to the router. For example, note that all interfaces on r4 that go to other router have “4” as the last octet.

IP ADDRESS ASSIGNMENTS		
ROUTER	PORT	IP ADDRESS
R1	E2/0	192.168.10.1/24
R1	E2/1	192.168.15.1/24
R1	S1/2	192.168.12.1/24
R1	S1/3	192.168.13.1/24
R2	LOOP0	192.168.20.1/24
R2	S1/1	192.168.12.2/24
R2	S1/4	192.168.14.2/24
R3	LOOP0	192.168.30.1/24
R3	S1/1	192.168.13.3/24
R3	S1/2	192.168.34.3/24
R4	LOOP0	192.168.40.1/24
R4	S1/2	192.168.24.4/24
R4	S1/3	192.468.34.4/24
R4	FDDI0/0	192.168.45.4/24
R5	E1	192.168.15.5/24
R5	FDDI0	192.168.45.5/24

Your users are complaining that sometimes the network is slow. Investigate using the built-in router tools “ping”, “traceroute”, “ttcp”, “show ip route”, “show cdp neighbor”, and “show ip protocol”. Measure the throughput between the different routers to quantify

what is “slow.” Why are some things “slow”? What can be done to correct these problems? What are some of the tradeoffs you have encountered between throughput and fault tolerance.

The initial router configurations for all five routers are in file *rip-config.txt*. You should be able to cut and paste the configurations into the routers. Output from “show ip route” and “show ip protocol” are on file *sh-ip-route.txt*. Output from “show cdp neighbor” are on file *sh-cdp-nei.txt*. By just looking at the diagram and routing tables, you should be able to manually determine the route IP packets will take one hop at a time through the network.

INITIAL ROUTER CONFIGURATIONS:

COMMON:

```
service udp-small-servers
service tcp-small-servers
enable password cisco
no ip domain-lookup
no ip classless
logging buffered
line con 0
  exec-timeout 0 0
line aux 0
line vty 0 4
  password cisco
  login
```

R1:

```
hostname r2
interface Loopback0
  description S3 LAN
  ip address 192.168.20.1 255.255.255.0
  no shutdown
interface Fddi0/0
  no ip address
  no shutdown
interface Serial1/1
  description Link to R1 S1/2
  ip address 192.168.12.2 255.255.255.0
  bandwidth 56
  clockrate 56000
  no shutdown
interface Serial1/4
  description Link to R4 S1/2
  ip address 192.168.24.2 255.255.255.0
  bandwidth 1544
  no shutdown
router rip
  network 192.168.20.0
  network 192.168.24.0
  network 192.168.12.0
```

R2:

```
hostname r2
interface Loopback0
  description S3 LAN
  ip address 192.168.20.1 255.255.255.0
  no shutdown
interface Fddi0/0
  no ip address
  no shutdown
interface Serial1/1
  description Link to R1 S1/2
  ip address 192.168.12.2 255.255.255.0
  bandwidth 56
  clockrate 56000
  no shutdown
interface Serial1/4
  description Link to R4 S1/2
  ip address 192.168.24.2 255.255.255.0
  bandwidth 1544
  no shutdown
router rip
  network 192.168.20.0
  network 192.168.24.0
  network 192.168.12.0
```

R3:

```
hostname r3
```

```
interface Loopback0
  description S4 LAN
  ip address 192.168.30.1 255.255.255.0
  no shutdown
interface Fddi0/0
  no ip address
  no shutdown
interface Serial1/1
  description Link to R1 S1/3
  ip address 192.168.13.3 255.255.255.0
  bandwidth 1544
  clockrate 2000000
  no shutdown
interface Serial1/4
  description Link to R4 S1/3
  ip address 192.168.34.3 255.255.255.0
  bandwidth 56
  no shutdown
router rip
  network 192.168.30.0
  network 192.168.34.0
  network 192.168.13.0
```

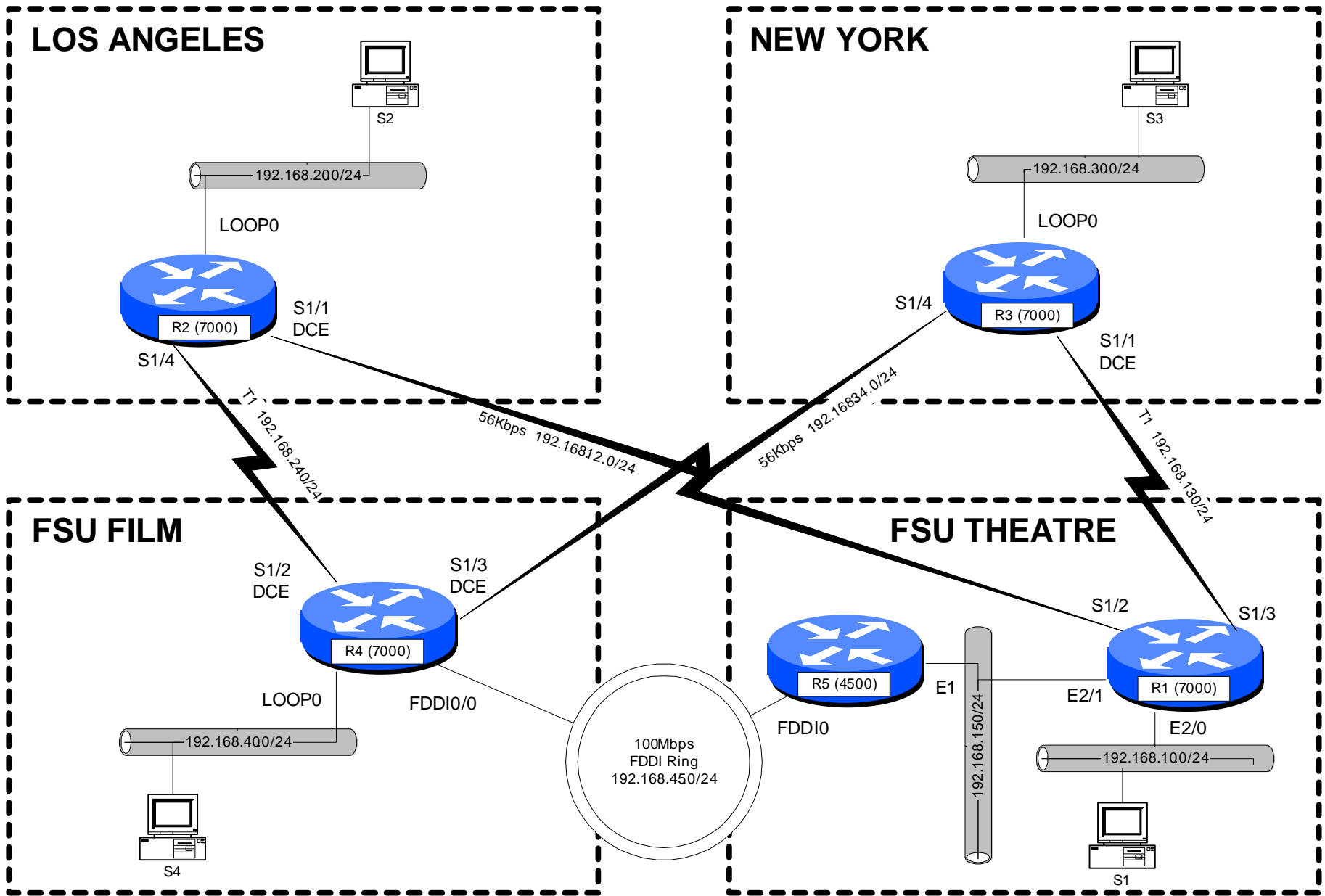
R4:

```
hostname r4
interface Loopback0
  description S2 LAN
  ip address 192.168.40.1 255.255.255.0
  no shutdown
interface Fddi0/0
  description Link to R5 FDDIO
  ip address 192.168.45.4 255.255.255.0
  no shutdown
interface Serial1/2
  description Link to R2 S1/4
  ip address 192.168.24.4 255.255.255.0
  bandwidth 1544
  clockrate 2000000
  no shutdown
interface Serial1/3
  description Link to R3 S1/4
  ip address 192.168.34.4 255.255.255.0
  bandwidth 56
  clockrate 56000
  no shutdown
router rip
  network 192.168.24.0
  network 192.168.34.0
  network 192.168.40.0
  network 192.168.45.0
```

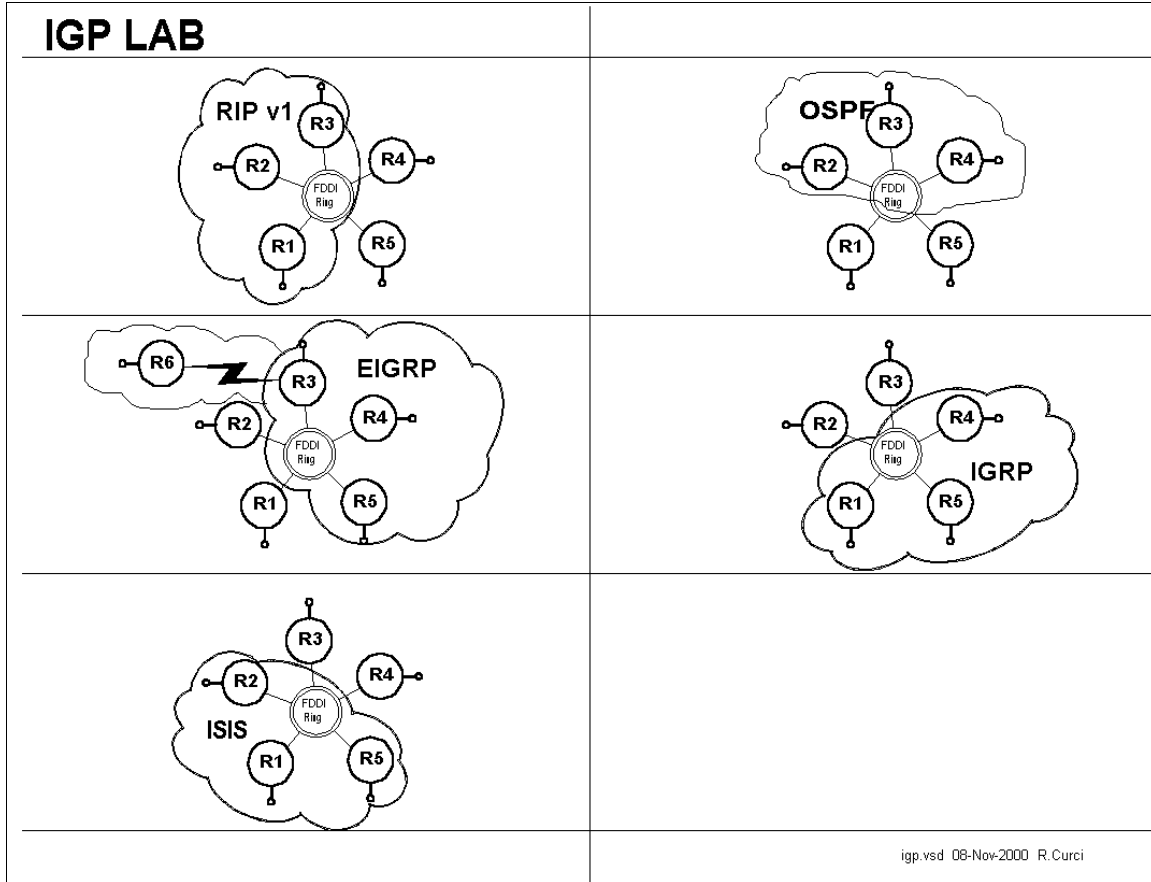
R5:

```
hostname r5
interface Ethernet1
  description Link to R1 E2/1
  ip address 192.168.15.5 255.255.255.0
  media-type 10BaseT
  no shutdown
interface Fddi0
  description Link to R4 FDDIO/0
  ip address 192.168.45.5 255.255.255.0
  no keepalive
  no shutdown
router rip
  network 192.168.45.0
  network 192.168.15.0
```

CENTER FOR ENTERTAINMENT STUDIES



INTERNET TEACHING LAB: Interior Gateway Protocol (IGP) LAB



Overview

In this lab, we will explore some common interior gateway protocols—

- RIP version 1 (Routing Information Protocol)
- OSPF (Open Shortest Path First)
- EIGRP (Enhanced Interior Gateway Routing Protocol)
- IGRP (Interior Gateway Routing Protocol)
- ISIS (Intermediate System – Intermediate System)

You will be configuring routers R1 through R5, while router R6 is preconfigured for EIGRP protocol on serial port S1 and will supply a default route for the lab network. For each of the above 5 routing protocols, three of the routers will participate as follows:

- RIP: R1,R2,R3
- OSPF: R2,R3,R4
- EIGRP: R3,R4,R5
- IGRP: R4,R5,R1
- ISIS: R5,R1,R2

Each of your routers will have a loopback and FDDI interface that needs to participate in all three appropriate routing protocols. Additionally, R1 and R5 will need the IGRP protocol on all ethernet and fast ethernet interfaces. Router R3 will need EIGRP on interface S1/6 to learn the default route to the outside world.

PART1 – IP ADDRESSING

Configure IP addresses as listed in the table below. Loopback0 interfaces need to be created if they do not exist and any other loopback addresses removed. Any interfaces not explicitly mentioned below, should be shut down. Once addressed, verify you have appropriate physical connectivity with “show cdp neighbors”. Verify that additional interfaces are shut down with “show ip interface brief.” At this point, you should be able to view your IP routing table with “show ip route” and should only see directly connected routes. Verify that you can PING the other router’s FDDI IP addresses. You will not be able to PING the other router’s loopback addresses because you will not have routes for them until later in this lab exercise. Make certain you have no static routes including default routes.

Rtr	Interface	IP Address/Mask	Routing Protocol(s)
R1	Loopback0	192.168.11.1/24	RIPv1,OSPF,IGRP
	Fddi0/0	192.168.1.1/24	RIPv1,OSPF,IGRP
	Ethernet2/0	192.168.10.1/24	IGRP
	Ethernet2/1	192.168.20.1/24	IGRP
	Ethernet2/2	192.168.30.1/24	IGRP
	Ethernet2/3	192.168.40.1/24	IGRP
	Ethernet2/4	192.168.50.1/24	IGRP
	Ethernet2/5	192.168.60.1/24	IGRP
R2	Loopback0	192.168.22.2/24	RIPv1,OSPF,ISIS
	Fddi0/0	192.168.1.2/24	RIPv1,OSPF,ISIS
R3	Loopback0	192.168.33.3/24	RIPv1,OSPF,EIGRP
	Fddi0/0	192.168.1.3/24	RIPv1,OSPF,EIGRP
	Serial1/6	192.168.36.3/24	EIGRP
R4	Loopback0	192.168.44.4/24	OSPF,EIGRP,IGRP
	Fddi0/0	192.168.1.4/24	OSPF,EIGRP,IGRP
R5	Loopback0	192.168.55.5/24	EIGRP,IGRP,ISIS
	Fddi0	192.168.1.5/24	EIGRP,IGRP,ISIS
	FastEther0	192.168.70.1/24	IGRP
	Ethernet0	192.168.80.1/24	IGRP
	Ethernet1	192.168.90.1/24	IGRP

Debug Mode

Cisco routers have a debug mode that can be helpful in debugging routing protocols, especially distance vector protocols. This mode should never be used on a production network because a large number of messages can be generated that can even cause a router to crash. To turn on your console window to receive debug messages, use the command “term monitor” or to turn it off “term no monitor.” To turn on a particular debug mode, use the command “debug XXX” such as “debug ip routing” or turn it off with “undebug all”. The command “debug ?” will show you your options. You can turn on more than one debug mode, or even turn them all on with “debug all”. To see which debug modes are active, use “show debug.”

PART2 – RIP (R1,R2,R3 Only)

Configure RIP on your router’s FDDI and Loopback0 interface. The following commands may be helpful.

- show ip route
- show ip route rip
- show ip protocol
- debug ip rip
- debug ip rip events

PART3 – OSPF (R2,R3,R4 Only)

Configure OSPF on your router’s FDDI and Loopback0 interface. Use process ID 100. Place all OSPF interfaces in the special OSPF backbone area 0. The following commands may be helpful.

- show ip route
- show ip route ospf
- show ip protocol
- show ip ospf neighbor
- show ip ospf interface
- show ip ospf database
- show ip ospf database database-summary
- debug ip ospf event
- debug ip ospf packet

PART4 – EIGRP (R3,R4,R5 Only)

Configure EIGRP on your router’s FDDI and Loopback0 interfaces. Use autonomous system number 100. The following commands may be helpful.

- show ip route

- show ip route eigrp
- show ip protocol
- show ip eigrp interfaces
- show ip eigrp neighbors
- show ip eigrp topology
- show ip eigrp traffic
- debug ip eigrp neighbor

PART5 – IGRP (R4,R5,R1 Only)

Configure IGRP on your router's FDDI and Loopback0 interfaces. On R1 and R5, also configure all ethernet and fast ethernet ports for IGRP. Use autonomous system 100. The following commands may be helpful.

- show ip route
- show ip route igrp
- show ip protocol
- debug ip igrp events
- debug ip igrp transactions

PART6 – ISIS (R5,R1,R2 Only)

Configure ISIS on your router's FDDI and Loopback0 interfaces. Use "100" for your ISO Routing Tag. ISIS incorporates an area number and MAC address into a "Network Entity Title" We will use area 1 and make up a dummy MAC address in the form NNNN.NNNN.NNNN for router N. Use the following Network Entity Title, substituting your router number for the letter N: "00.0001.NNNN.NNNN.NNNN.00". In this example, the "00.0001" represents the area number in hex, while the "NNNN.NNNN.NNNN.00" is an identifier for your router in hex. The following commands may be helpful.

- show ip route
- show ip route isis
- show ip protocol
- show isis database

PART7 – Route Redistribution (R3 Only)

Router R3 should be receiving EIGRP routes from R6 including a default route (0.0.0.0) and a route for R6's Loopback0 interface 192.168.66.6. Some of the routers, however, may not be getting these routes. On R3 only, redistribute all RIP routes into both RIP and OSPF. For RIP, use a hop count/metric of 10. Verify with "show ip route" that you can see both 0.0.0.0 and 192.168.66.6/24 from all routers.

PART8 – Verification

Verify that everything is working. You can display the routing tables with "show ip route" which should look like the the output below. Note that the letter designation to the left of each routing entry indicates which protocol put the route in the routing table. When the same route is learned by multiple protocols, the protocol with the lowest administrative distance is used. Administrative distance is like a believability factor. Administrative distances for some common protocols are listed in the table below. You will notice in the output below, that the "show ip route" output entries indicate two numbers in square brackets, administrative distance and route metric.

PROTOCOL	ADMIN.DIST.
Connected	0
Static	1
EIGRP	90
IGRP	100
ISIS	115
OSPF	110
ISIS	115

Codes: **C** - connected, **S** - static, **I** - IGRP, **R** - RIP, **M** - mobile, **B** - BGP
D - EIGRP, **EX** - EIGRP external, **O** - OSPF, **IA** - OSPF inter area
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

R1:

```
Gateway of last resort is 192.168.1.3 to network 0.0.0.0
R   192.168.66.0/24 [120/10] via 192.168.1.3, 00:00:06, Fddi0/0
I   192.168.90.0/24 [100/1110] via 192.168.1.5, 00:01:08, Fddi0/0
I   192.168.80.0/24 [100/1110] via 192.168.1.5, 00:01:08, Fddi0/0
C   192.168.40.0/24 is directly connected, Ethernet2/3
I   192.168.44.0/24 [100/610] via 192.168.1.4, 00:01:19, Fddi0/0
R   192.168.33.0/24 [120/1] via 192.168.1.3, 00:00:06, Fddi0/0
R   192.168.36.0/24 [120/10] via 192.168.1.3, 00:00:06, Fddi0/0
C   192.168.60.0/24 is directly connected, Ethernet2/5
C   192.168.50.0/24 is directly connected, Ethernet2/4
I   192.168.55.0/24 [100/610] via 192.168.1.5, 00:01:08, Fddi0/0
C   192.168.10.0/24 is directly connected, Ethernet2/0
C   192.168.11.0/24 is directly connected, Loopback0
C   192.168.1.0/24 is directly connected, Fddi0/0
C   192.168.30.0/24 is directly connected, Ethernet2/2
C   192.168.20.0/24 is directly connected, Ethernet2/1
i L1 192.168.22.0/24 [115/20] via 192.168.1.2, Fddi0/0
R*  0.0.0.0/0 [120/10] via 192.168.1.3, 00:00:06, Fddi0/0
```


R2:

```
Gateway of last resort is 192.168.1.3 to network 0.0.0.0
O E2 192.168.66.0/24 [110/100] via 192.168.1.3, 00:26:50, Fddi0/0
O E2 192.168.90.0/24 [110/100] via 192.168.1.5, 00:26:50, Fddi0/0
O E2 192.168.80.0/24 [110/100] via 192.168.1.5, 00:26:50, Fddi0/0
O E2 192.168.40.0/24 [110/100] via 192.168.1.4, 00:26:50, Fddi0/0
    192.168.44.0/32 is subnetted, 1 subnets
O      192.168.44.4 [110/2] via 192.168.1.4, 00:26:50, Fddi0/0
    192.168.33.0/24 is variably subnetted, 2 subnets, 2 masks
O E2 192.168.33.0/24 [110/100] via 192.168.1.3, 00:26:50, Fddi0/0
O      192.168.33.3/32 [110/2] via 192.168.1.3, 00:26:50, Fddi0/0
O E2 192.168.36.0/24 [110/100] via 192.168.1.3, 00:26:50, Fddi0/0
O E2 192.168.60.0/24 [110/100] via 192.168.1.4, 00:26:50, Fddi0/0
O E2 192.168.50.0/24 [110/100] via 192.168.1.4, 00:26:50, Fddi0/0
O E2 192.168.55.0/24 [110/100] via 192.168.1.5, 00:26:50, Fddi0/0
O E2 192.168.10.0/24 [110/100] via 192.168.1.4, 00:26:50, Fddi0/0
i L1 192.168.11.0/24 [115/20] via 192.168.1.1, Fddi0/0
C      192.168.1.0/24 is directly connected, Fddi0/0
O E2 192.168.30.0/24 [110/100] via 192.168.1.4, 00:26:50, Fddi0/0
O E2 192.168.20.0/24 [110/100] via 192.168.1.4, 00:26:50, Fddi0/0
C      192.168.22.0/24 is directly connected, Loopback0
R*    0.0.0.0/0 [120/10] via 192.168.1.3, 00:00:08, Fddi0/0
```

R3:

```
Gateway of last resort is 192.168.36.6 to network 0.0.0.0
D      192.168.66.0/24 [90/2297856] via 192.168.36.6, 01:24:50, Serial1/6
D      192.168.90.0/24 [90/284160] via 192.168.1.5, 01:24:50, Fddi0/0
D      192.168.80.0/24 [90/284160] via 192.168.1.5, 01:24:50, Fddi0/0
D EX 192.168.40.0/24 [170/286720] via 192.168.1.4, 01:09:33, Fddi0/0
    [170/286720] via 192.168.1.5, 01:09:33, Fddi0/0
    192.168.44.0/24 is variably subnetted, 2 subnets, 2 masks
O      192.168.44.4/32 [110/2] via 192.168.1.4, 00:26:52, Fddi0/0
D      192.168.44.0/24 [90/156160] via 192.168.1.4, 01:24:50, Fddi0/0
C      192.168.33.0/24 is directly connected, Loopback0
C      192.168.36.0/24 is directly connected, Serial1/6
D EX 192.168.60.0/24 [170/286720] via 192.168.1.4, 01:09:33, Fddi0/0
    [170/286720] via 192.168.1.5, 01:09:33, Fddi0/0
D EX 192.168.50.0/24 [170/286720] via 192.168.1.4, 01:09:33, Fddi0/0
    [170/286720] via 192.168.1.5, 01:09:33, Fddi0/0
D      192.168.55.0/24 [90/156160] via 192.168.1.5, 01:24:50, Fddi0/0
D EX 192.168.10.0/24 [170/286720] via 192.168.1.4, 01:09:33, Fddi0/0
    [170/286720] via 192.168.1.5, 01:09:33, Fddi0/0
R      192.168.11.0/24 [120/1] via 192.168.1.1, 00:00:10, Fddi0/0
C      192.168.1.0/24 is directly connected, Fddi0/0
D EX 192.168.30.0/24 [170/286720] via 192.168.1.4, 01:09:33, Fddi0/0
    [170/286720] via 192.168.1.5, 01:09:33, Fddi0/0
D EX 192.168.20.0/24 [170/286720] via 192.168.1.4, 01:09:33, Fddi0/0
    [170/286720] via 192.168.1.5, 01:09:33, Fddi0/0
    192.168.22.0/24 is variably subnetted, 2 subnets, 2 masks
O      192.168.22.2/32 [110/2] via 192.168.1.2, 00:26:52, Fddi0/0
R      192.168.22.0/24 [120/1] via 192.168.1.2, 00:00:27, Fddi0/0
D*EX 0.0.0.0/0 [170/2195456] via 192.168.36.6, 01:24:50, Serial1/6
```

R4:

```
Gateway of last resort is 192.168.1.3 to network 0.0.0.0
D      192.168.66.0/24 [90/2300416] via 192.168.1.3, 01:24:08, Fddi0/0
D      192.168.90.0/24 [90/284160] via 192.168.1.5, 01:24:08, Fddi0/0
D      192.168.80.0/24 [90/284160] via 192.168.1.5, 01:24:08, Fddi0/0
I      192.168.40.0/24 [100/1110] via 192.168.1.1, 00:00:36, Fddi0/0
C      192.168.44.0/24 is directly connected, Loopback0
    192.168.33.0/24 is variably subnetted, 2 subnets, 2 masks
```

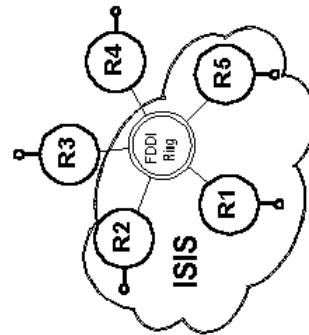
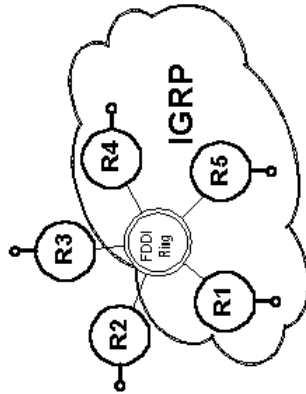
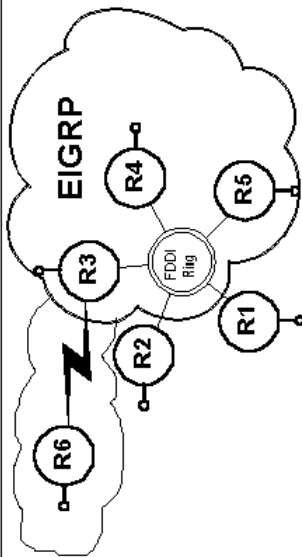
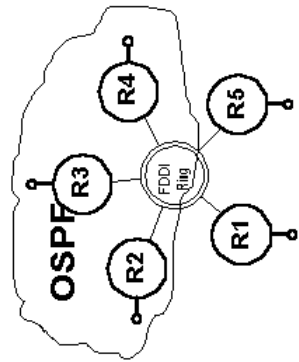
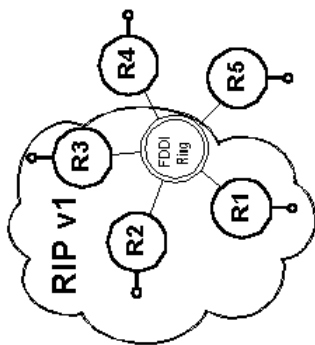
D 192.168.33.0/24 [90/156160] via 192.168.1.3, 01:24:08, Fddi0/0
O 192.168.33.3/32 [110/2] via 192.168.1.3, 00:26:55, Fddi0/0
D 192.168.36.0/24 [90/2172416] via 192.168.1.3, 01:24:08, Fddi0/0
I 192.168.60.0/24 [100/1110] via 192.168.1.1, 00:00:36, Fddi0/0
I 192.168.50.0/24 [100/1110] via 192.168.1.1, 00:00:36, Fddi0/0
D 192.168.55.0/24 [90/156160] via 192.168.1.5, 01:24:08, Fddi0/0
I 192.168.10.0/24 [100/1110] via 192.168.1.1, 00:00:36, Fddi0/0
I 192.168.11.0/24 [100/610] via 192.168.1.1, 00:00:36, Fddi0/0
C 192.168.1.0/24 is directly connected, Fddi0/0
I 192.168.30.0/24 [100/1110] via 192.168.1.1, 00:00:36, Fddi0/0
I 192.168.20.0/24 [100/1110] via 192.168.1.1, 00:00:36, Fddi0/0
192.168.22.0/32 is subnetted, 1 subnets
O 192.168.22.2 [110/2] via 192.168.1.2, 00:26:55, Fddi0/0
D*EX 0.0.0.0/0 [170/2198016] via 192.168.1.3, 01:24:08, Fddi0/0

R5:

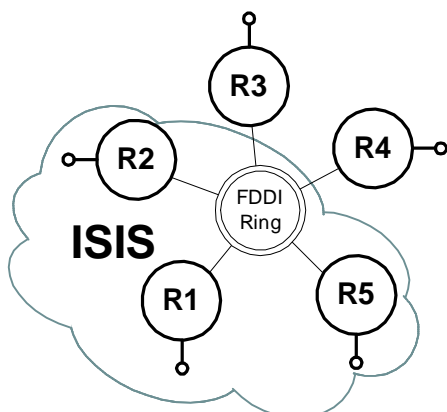
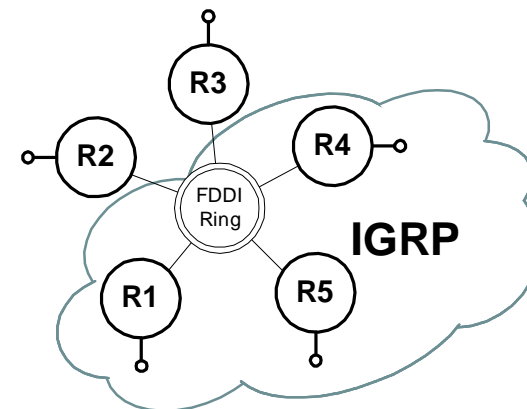
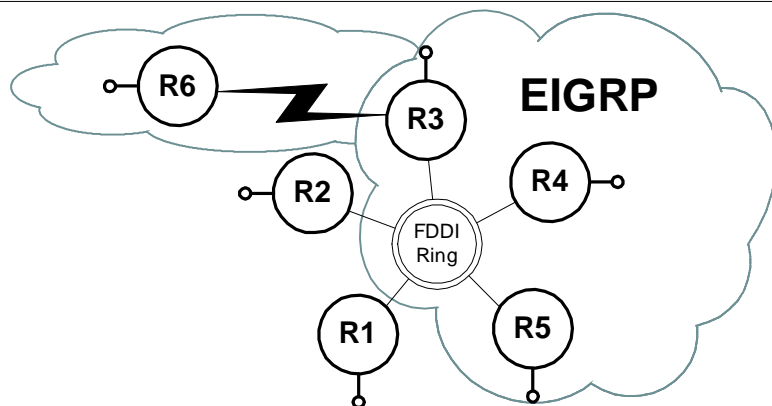
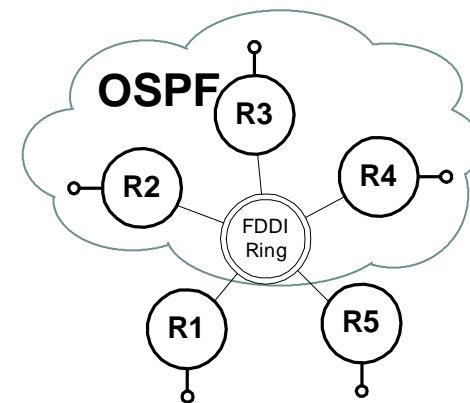
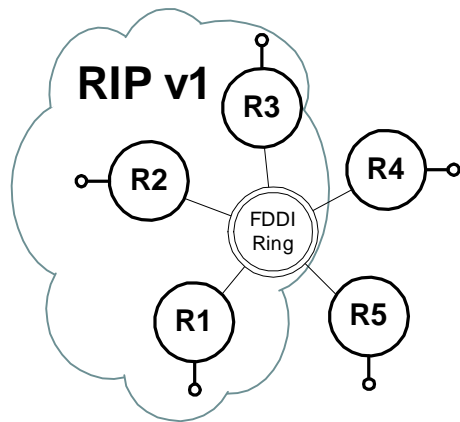
Gateway of last resort is 192.168.1.3 to network 0.0.0.0

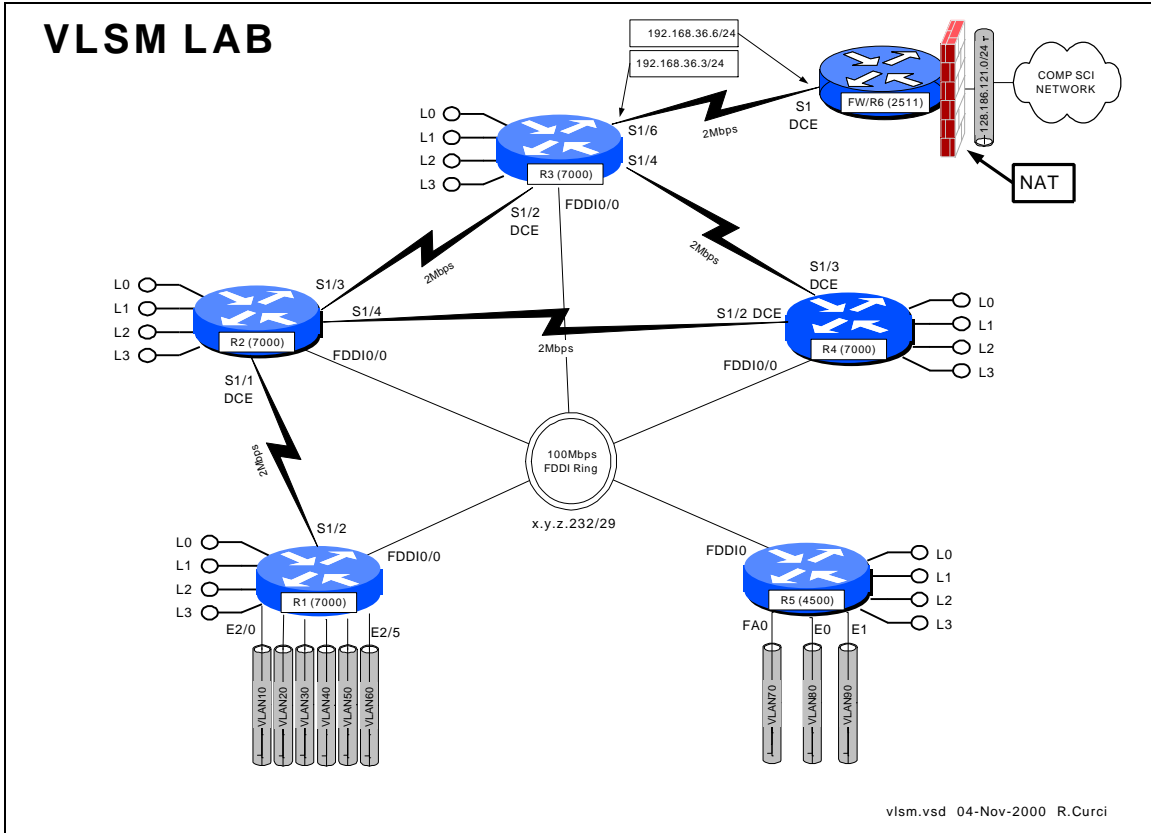
D 192.168.44.0/24 [90/156160] via 192.168.1.4, 03:57:37, Fddi0
C 192.168.90.0/24 is directly connected, Ethernet1
I 192.168.30.0/24 [100/1110] via 192.168.1.1, 00:00:38, Fddi0
I 192.168.60.0/24 [100/1110] via 192.168.1.1, 00:00:38, Fddi0
I 192.168.10.0/24 [100/1110] via 192.168.1.1, 00:00:38, Fddi0
I 192.168.40.0/24 [100/1110] via 192.168.1.1, 00:00:38, Fddi0
I 192.168.11.0/24 [100/610] via 192.168.1.1, 00:00:38, Fddi0
C 192.168.55.0/24 is directly connected, Loopback0
C 192.168.80.0/24 is directly connected, Ethernet0
I 192.168.20.0/24 [100/1110] via 192.168.1.1, 00:00:38, Fddi0
D 192.168.66.0/24 [90/2300416] via 192.168.1.3, 01:26:36, Fddi0
D 192.168.36.0/24 [90/2172416] via 192.168.1.3, 01:26:38, Fddi0
i L1 192.168.22.0/24 [115/20] via 192.168.1.2, Fddi0
I 192.168.50.0/24 [100/1110] via 192.168.1.1, 00:00:38, Fddi0
C 192.168.1.0/24 is directly connected, Fddi0
D 192.168.33.0/24 [90/156160] via 192.168.1.3, 01:26:38, Fddi0
D*EX 0.0.0.0/0 [170/2198016] via 192.168.1.3, 01:26:36, Fddi0

IGP LAB



IGP LAB





Overview

The Internet is running out of IP address space and your network addressing scheme with all /24 subnets is wasteful. Configure routers R1,R2,R3,R4, and R5 as shown above. Create a new IP addressing scheme for the network that efficiently utilizes class C network 192.168.100.0/24. Do not worry about router R6 or the R3 -R6 serial link which can be numbered as shown in the diagram. You will need to use a technique called “variable length subnet masking” (VLSM) where you subdivide your network address space into subnetworks of different sizes. When you have finished this exercise, capture the output of the following commands to prove you completed the assignment.

- show running-config
- show ip interface brief
- show cdp neighbors
- show ip ospf neighbor
- show ip route
- show ip protocol

PART 1 – IP Addressing

On each router R1 through R5, create four loopback interfaces that will support the following number of hosts.

INTERFACE	# HOSTS
loopback0	14 hosts
loopback1	6 hosts
loopback2	2 hosts
loopback3	2 hosts

Begin by looking at each network and deciding how many host addresses must be supported to figure out the size of each subnet. You must do this with maximum efficiency as there are no extra addresses, only exactly enough to solve this problem. For each of the five routers, select the loopback subnets such that they can be summarized. If you do not understand the concept of summarization, read up on CIDR – Classless Internet Domain Routing. You will need to use the command “ip classless” on your router to make it ignore the classfull (i.e. Class A, B, C) network mask assumptions. Since we will be using all subnets including subnet zero, you will also need the command “ip subnet-zero” in your configuration.

PART 2 – OSPF Routing

When using variable length subnet masks in your network, you will need an IP routing protocol that supports VLSM such as OSPF (Open Shortest Path First). Configure OSPF as your only routing protocol. All FDDI, Ethernet, FastEthernet, and Serial interfaces should be in area 0. Place the loopback addresses on each of the five routers in a separate area corresponding to the router identifier. For example, the loopback addresses on router 3 should be in area 3. You may wish to use the following commands to help debug your OSPF configuration:

- show ip ospf neighbor
- show ip ospf database
- show ip ospf database-summary
- show ip ospf interface
- show ip route
- show ip route ospf
- show ip protocol

PART 3 – Address Summarization

In large networks like the Internet, the number of network routes that fit in the routing table becomes a limiting factor. In the mid 1980s with the exponential growth of the Internet, many predicted the collapse of the Internet backbone due to the growing size of

the routing tables. This problem was helped by the creation of CIDR – Classless Internet Domain Routing, which summarizes network blocks without regard to the classfull network designations. As of this writing, there are approximately 90,000 routes on the Internet, a number that would be much higher without CIDR. Routing protocols like OSPF are very scalable when used with hierarchical network addressing schemes that support summarization. Your routers should be advertising their loopback addresses as individual routes, each creating its own routing table entry in the routing tables of the other routers. For each of the five routers, reconfigure OSPF to advertise a single summary route for all four loopback addresses instead of advertising them individually. Because each router is participating in more than one OSPF area, it is an autonomous system boundary router (ASBR). ASBRs can summarize the routes within their non-zero areas into the core area zero to reduce the number of routes the core area zero routers must keep in their tables. Verify everything is working by studying the output of the commands “show ip route”, “show ip protocol”, “show ip ospf neighbor”, “show ip ospf database database-summary”, “show ip ospf interface”, etc. If you simply type “show ip ospf ?” you will see the various options available.

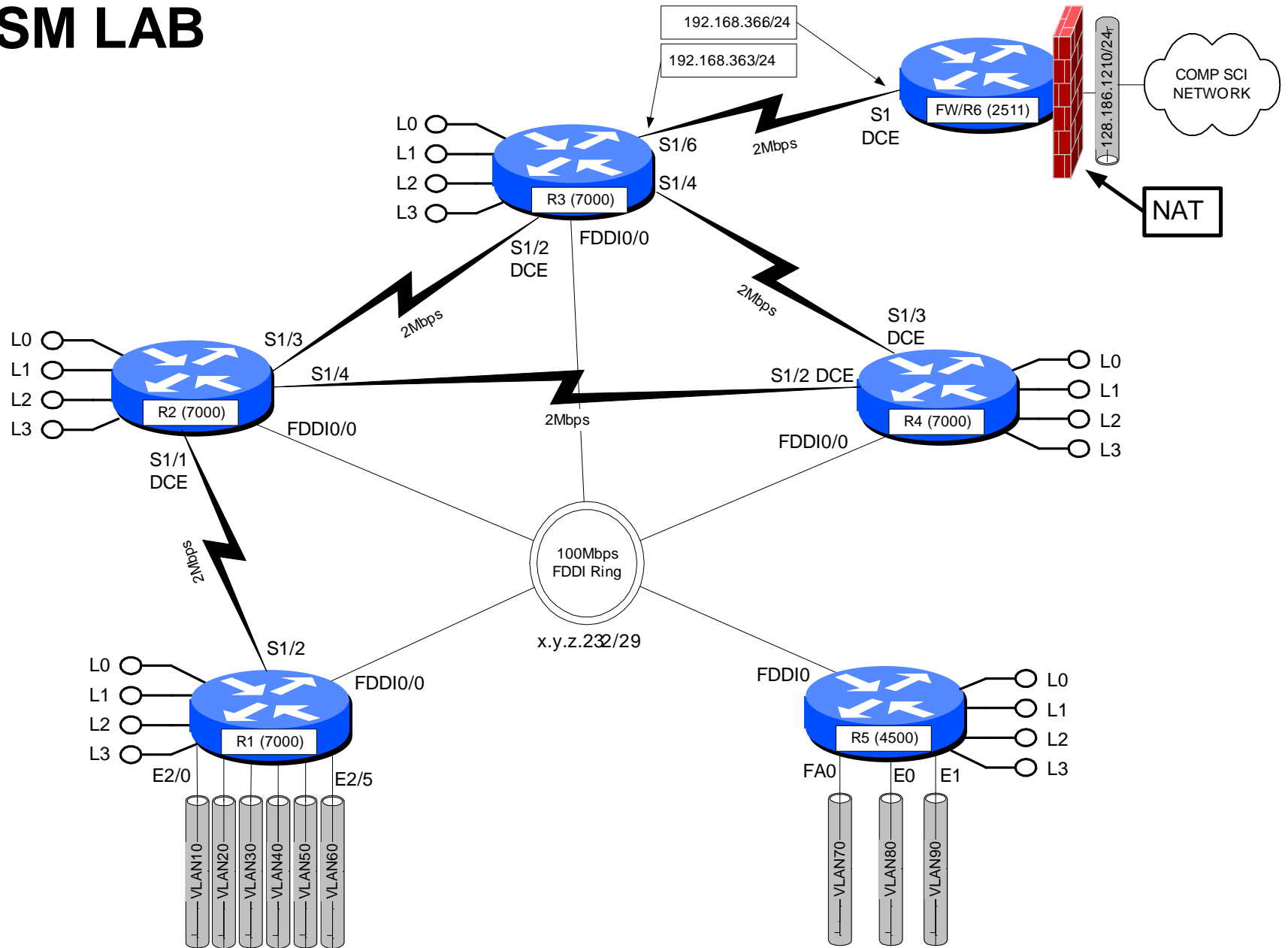
PART 4 – Network Assurance

Assign new IP addresses for your PCs using the new IP address scheme. Note that not only your IP address, but also your gateway, broadcast address, netmask, and network addresses have changed. Verify everything is reachable by scanning the lab network from a UNIX PC using the NMAP utility. This utility can be found at www.insecure.org/nmap. Be sure to only scan within the lab network because probes outside the lab will cause firewalls and intrusion detection systems to complain and are presently treated by law enforcement as attempted unauthorized access.

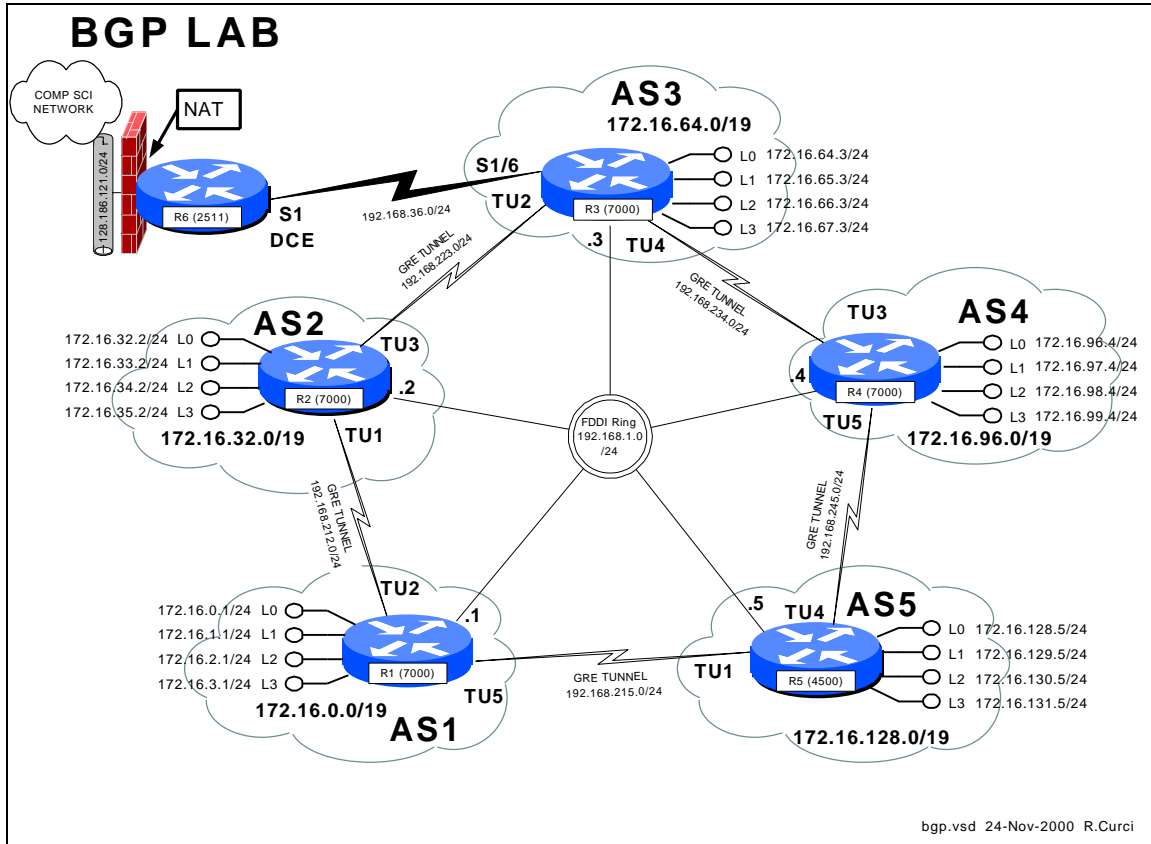
PART 5 – GateD / Extra Credit

Configure your Linux system to use GateD by modifying file `/etc/gated.conf`. Configure your system to use the OSPF routing protocol on its ethernet port which should be in area zero. Use the command “`netstat -rn`” to display your routing table. You should see routes for all networks in area zero plus the summary routes for non-area zero networks. You should also see a default route sometimes listed as ‘0.0.0.0’. Be sure to remove any static default route on your system, as you should learn the default dynamically from OSPF.

VLSM LAB



INTERNET TEACHING LAB: BGP LAB



Overview

In this lab, we will explore the Border Gateway Protocol (BGP) and Generic Route Encapsulation (GRE) tunnels. Each router r1 through r5 will physically connect to a common FDDI ring. A set of 5 GRE tunnels will be implemented connecting r1→r2, r2→r3, r3→r4, r4→r5, and r5→r1. These tunnels do not use TCP or UDP, but instead a separate protocol number 47 that operates over IP. Once established, tunnels are treated by the router like any other point-to-point interface. Each router r1 through r5 will be in a separate autonomous system each with its own /19 CIDR block of IP address space. Each router r1 through r5 will be configured to peer using exterior BGP with its two neighbors. BGP version 4 is the exterior routing protocol deployed on the backbone of the Internet. BGP organizes the network into autonomous systems identified by autonomous system numbers (ASNs). ASNs are uniquely assigned by the American Registry for Internet Numbers (ARIN). Only organizations with more than one Internet Service Provider (ISP) who are “multihomed” are eligible to receive a registered ASN. You can find out more about BGP in the Cisco routing protocols configuration guide. As of this writing, the definitive source of information for this protocol is the textbook Internet Routing Architectures by Bassam Halabi published by Cisco Press in 1997.

Here is the FSU autonomous system number registration record at ARIN:

```
acns% whois -h whois.arin.net 2553
Florida State University (ASN-FSU)
  Academic Computing & Network Services
  Room 200, Sliger Building
  2035 East Paul Dirac Drive
  Tallahassee, FL 32310

Autonomous System Name: FSU-AS
Autonomous System Number: 2553

Coordinator:
  Garner, Lee [Systems Programmer] (LG36-ARIN) garner@ACNS.FSU.EDU
  850-644-2592 (FAX) 850-644-8722

Record last updated on 25-Jan-1995.
Database last updated on 24-Nov-2000 18:13:50 EDT.
```

Here is a summary of BGP peering sessions on the FSU BFS-7507 router. Note that our peer at IP address 199.44.5.225 (Sprint) is sending us over 92,000 prefixes.

```
bfs-7507#show ip bgp sum
BGP router identifier 128.186.253.5, local AS number 2553
BGP table version is 10339797, main routing table version 10339797
93124 network entries and 293284 paths using 19684376 bytes of memory
44120 BGP path attribute entries using 2294812 bytes of memory
23517 BGP AS-PATH entries using 634144 bytes of memory
32 BGP community entries using 852 bytes of memory
1772 BGP route-map cache entries using 28352 bytes of memory
34843 BGP filter-list cache entries using 418116 bytes of memory
109503 received paths for inbound soft reconfiguration
BGP activity 657129/958415 prefixes, 6401589/6108305 paths, scan interval 15 secs

Neighbor      V    AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
128.186.250.194 4   7202   72889   72879 10339797  0    0  7w1d      1
128.186.250.201 4   3996   73232   72886 10339797  0    0  3w3d      39
128.186.253.7   4   2553 2966677 2230491 10339797  0    0  3w0d      74247
192.80.53.41    4  11537 128228   72861 10339774  0    0  3w2d      4025
192.80.53.62    4   6356   72699   72929 10339792  0    0  5d13h     3
192.80.53.66    4   5661   72870   72878 10339797  0    0  7w1d      1
192.80.53.70    4   7939   72919   72922 10339774  0    0  1w0d      1
192.80.53.106   4   3506 216733 3135856 10339774  0    0  7w1d     12960
199.44.5.225    4   3447 2356372  72883 10339774  0    0  7w1d     92501
```

FSU is only advertising a small number of networks to our ISP (Sprint). This helps prevent us from unintentionally becoming a transit AS:

```
bfs-7507#show ip bgp neighbor 199.44.5.225 advertised-routes
BGP table version is 10339840, local router ID is 128.186.253.5
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
*> 128.186.0.0    0.0.0.0           0      32768 i
*> 144.174.0.0    192.80.53.106     0      155    0 3506 i
*> 146.201.0.0    0.0.0.0           20     32768 i
*> 192.80.53.0    0.0.0.0           0      32768 i
bfs-7507#
```

PART1 – Basic IGP (RIP) Configuration

Each router r1 through r5 will have only its physical FDDI interface enabled. The only exception is router r3 who will additionally have its serial port enabled to connect with r6 for Internet connectivity. When finished with this part, verify that you can PING the loopback0 IP address on r6, 192.168.66.6. Test by PINGing the FDDI IP broadcast address 192.168.1.255. You should hear responses from the other 4 FDDI-connected routers if all is well.

The following commands may be helpful in debugging this part:

- show cdp neighbor
- ping w.x.y.z
- show ip protocol
- show ip route
- show ip route RIP

For each router, you will need both the common part of the configuration and router specific portion as appropriate that follows:

COMMON:

```
service udp-small-servers
service tcp-small-servers
enable password cisco
no ip domain-lookup
ip classless
ip subnet-zero
logging buffered
clock timezone EST -5
clock summer-time EDT recurring
ntp server 192.168.66.6
snmp-server community public RO
line con 0
  exec-timeout 0 0
line aux 0
line vty 0 4
  password cisco
  login
```

R1:

```
hostname r1
interface Fddi0/0
  ip address 192.168.1.1 255.255.255.0
  no shutdown
interface E2/0
  description Vlan 10 to cat1 FA0/1
  ip address 192.168.10.1 255.255.255.0
  no shutdown
interface E2/1
  description Vlan 20 to cat1 FA0/2
  ip address 192.168.20.1 255.255.255.0
  no shutdown
interface E2/2
  description Vlan 30 to cat1 FA0/3
  ip address 192.168.30.1 255.255.255.0
  no shutdown
interface E2/3
  description Vlan 40 to cat1 FA0/4
  ip address 192.168.40.1 255.255.255.0
  no shutdown
interface E2/4
  description Vlan 50 to cat1 FA0/5
  ip address 192.168.50.1 255.255.255.0
  no shutdown
interface E2/5
  description Vlan 60 to cat1 FA0/6
  ip address 192.168.60.1 255.255.255.0
  no shutdown
router rip
  network 192.168.1.0
  network 192.168.10.0
  network 192.168.20.0
  network 192.168.30.0
  network 192.168.40.0
  network 192.168.50.0
  network 192.168.60.0
```

R2:

```
hostname r2
interface Fddi0/0
  ip address 192.168.1.2 255.255.255.0
```

```
no shutdown
router rip
  network 192.168.1.0
```

R3:

```
hostname r3
interface Fddi0/0
  ip address 192.168.1.3 255.255.255.0
  no shutdown
interface Serial1/6
  description Link to R6 S1
  ip address 192.168.36.3 255.255.255.0
  bandwidth 2000
  no shutdown
router rip
  network 192.168.36.0
  network 192.168.1.0
```

R4:

```
hostname r4
interface Fddi0/0
  description Link to R5 FDDIO
  ip address 192.168.1.4 255.255.255.0
  no shutdown
router bgp 4
  network 172.16.96.0 mask 255.255.224.0
  neighbor 192.168.234.3 remote-as 3
  neighbor 192.168.234.3 version 4
  neighbor 192.168.245.5 remote-as 5
  neighbor 192.168.245.5 version 4
  ip route 172.16.96.0 255.255.224.0 null0
router rip
  network 192.168.1.0
```

R5:

```
hostname r5
interface FastEthernet0
  description Vlan70 to cat1 FA0/7
  ip address 192.168.70.1 255.255.255.0
  media-type 100BaseX
  no shutdown
interface Ethernet0
  description Vlan80 to cat1 FA0/8
  ip address 192.168.80.1 255.255.255.0
  media-type 10BaseT
  no shutdown
interface Ethernet1
  description Vlan90 to cat1 FA0/9
  ip address 192.168.90.1 255.255.255.0
  media-type 10BaseT
  no shutdown
interface Fddi0
  description Link to R4 FDDIO/0
  ip address 192.168.1.5 255.255.255.0
  no keepalive
  no shutdown
router rip
  network 192.168.70.0
  network 192.168.80.0
  network 192.168.90.0
  network 192.168.1.0
```

PART2 – GRE Tunnel and Loopback Interfaces

GRE tunnel and loopback interfaces are virtual interfaces created in the Cisco IOS software. On each router, establish two GRE tunnel interfaces and four loopback interfaces as shown on your network diagram and table below. GRE Tunnel interfaces are normally used to encapsulate non-IP traffic through an IP-only core network or to encapsulate private IP addresses through the public Internet. Recent versions of the Linux operating system also support GRE tunnels. The tunnel interfaces in this lab will encapsulate IP traffic in frames that will physically traverse the FDDI ring but will appear to the routers as point-to-point interfaces. You will assign an IP address to each tunnel interface just like a serial point-to-point interface. Anchor the tunnels using the FDDI IP addresses as specified in the following table. Be sure you can PING both your tunnel endpoints and the IP address assigned to the tunnel interfaces on the other side. Do **NOT** enable RIP on any tunnel or loopback interfaces (**NOT** on any 172.16.x.y interfaces). We will use BGP for routing across the tunnels in the next part. Note that CDP does not work across tunnel interfaces. The following commands may be helpful in debugging this section:

- ping
- show ip interface
- show ip interface brief
- clear counters
- show interface

Notice that the loopback and tunnel interfaces have status=up and protocol=up:

```
r1#show ip int brief
Interface          IP-Address      OK? Method Status  Protocol
Fddi0/0            192.168.1.1    YES manual up      up
Loopback0          172.16.0.1     YES manual up      up
Loopback1          172.16.1.1     YES manual up      up
Loopback2          172.16.2.1     YES manual up      up
Loopback3          172.16.3.1     YES manual up      up
Tunnel2            192.168.212.1 YES manual up      up
Tunnel5            192.168.215.1 YES manual up      up
r1#
```

Here is an example “show interface” command on a GRE tunnel:

```
r1#sh int tunnel2
Tunnel2 is up, line protocol is up
  Hardware is Tunnel
  Description: Tunnel to R2
  Internet address is 192.168.212.1/24
  MTU 1500 bytes, BW 9 Kbit, DLY 500000 usec, rely 255/255, load 1/255
  Encapsulation TUNNEL, loopback not set, keepalive set (10 sec)
  Tunnel source 192.168.1.1, destination 192.168.1.2
  Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
  ...
```

Rtr	Interface	IP Address	Tunnel Src	Tunnel Dest
r1	fddi0/0	192.168.1.1/24		
	loopback0	172.16.0.1/24		
	loopback1	172.16.1.1/24		
	loopback2	172.16.2.1/24		
	loopback3	172.16.3.1/24		
	tunnel2	192.168.212.1/24	192.168.1.1	192.168.1.2
	tunnel5	192.168.215.1/24	192.168.1.1	192.168.1.5
	ethernet2/0	192.168.10.1/24		
	ethernet2/1	192.168.20.1/24		
	ethernet2/2	192.168.30.1/24		
	ethernet2/3	192.168.40.1/24		
	ethernet2/4	192.168.50.1/24		
	ethernet2/5	192.168.60.1/24		
	r2	fddi0/0	192.168.1.2/24	
loopback0		172.16.32.2/24		
loopback1		172.16.33.2/24		
loopback2		172.16.34.2/24		
loopback3		172.16.35.2/24		
tunnel1		192.168.212.2/24	192.168.1.2	192.168.1.1
tunnel3		192.168.223.2/24	192.168.1.2	192.168.1.3
r3	fddi0/0	192.168.1.3/24		
	loopback0	172.16.64.3/24		
	loopback1	172.16.65.3/24		
	loopback2	172.16.66.3/24		
	loopback3	172.16.67.3/24		
	tunnel2	192.168.223.3/24	192.168.1.3	192.168.1.2
	tunnel4	192.168.234.3/24	192.168.1.3	192.168.1.4
serial1/6	192.168.36.3/24			
r4	fddi0/0	192.168.1.4/24		
	loopback0	172.16.96.4/24		
	loopback1	172.16.97.4/24		
	loopback2	172.16.98.4/24		
	loopback3	172.16.99.4/24		
	tunnel3	192.168.234.4/24	192.168.1.4	192.168.1.3
	tunnel5	192.168.245.4/24	192.168.1.4	192.168.1.5
r5	fddi0	192.168.1.5/24		
	loopback0	172.16.128.5/24		
	loopback1	172.16.129.5/24		
	loopback2	172.16.130.5/24		
	loopback3	172.16.131.5/24		
	tunnel1	192.168.215.5/24	192.168.1.5	192.168.1.1
	tunnel4	192.168.245.5/24	192.168.1.5	192.168.1.4
	fastethernet0	192.168.70.1/24		
ethernet0	192.168.80.1/24			
ethernet1	192.168.90.1/24			

PART3 – BGP Peering

On each router r1 through r5, establish a BGP peering session through each tunnel interface to your neighbor. You will be using exterior BGP or EBGP since each router is in a different ASN. On each router, you will need to advertise the networks on your loopback addresses. Instead of advertising these /24 blocks individually, you should advertise only a single prefix with a /19 network mask as documented in the diagram. When everything is working, each router r1 through r5 should have two BGP peering sessions. You should be receiving 3 network advertisements from each of your peers. We will be using the AS path length to determine the best BGP route. For example, on router r1, the BGP route to network 172.16.0.0/19 and 172.16.64.0/19 should be via Tunnel2, while the best route to networks 172.16.96.0/19 and 172.16.128.0/19 should be via Tunnel5.

The following commands may be helpful in debugging this section:

- show ip route
- show ip bgp sum
- show ip bgp neighbor w.x.y.z
- show ip bgp neighbor w.x.y.z routes
- show ip bgp neighbor w.x.y.z advertised-routes
- show ip bgp regexp .*

The following are some sample SHOW command executed from router r1 to give you an idea of what you can expect when everything is working. Note that there are two active BGP peering sessions:

```
r1#sh ip bgp sum
BGP table version is 26, main routing table version 26
5 network entries (7/15 paths) using 1092 bytes of memory
7 BGP path attribute entries using 800 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory

Neighbor          V    AS MsgRcvd MsgSent  TblVer  InQ  OutQ Up/Down  State
192.168.212.2     4     2    100    102     26   0    0 01:10:40
192.168.215.5     4     5    111    120     26   0    0 00:01:55
```

These are the BGP routes we are advertising to our BGP neighbors. The only internal route we are advertising is 172.16.0.0/19. Note that the other advertised routes are learned from our BGP peers and have ASPATH “2 3 4”, “5 4”, and “5” which all begin with one of our peer’s ASNs:

```

r1#sh ip bgp nei 192.168.212.2 advertised-routes
BGP table version is 26, local router ID is 172.16.3.1
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
*> 172.16.0.0/19	0.0.0.0	0		32768	i
* 172.16.96.0/19	192.168.212.2			0	2 3 4 i
*>	192.168.215.5			0	5 4 i
*> 172.16.128.0/19	192.168.215.5	0		0	5 i

Here are the routes we are receiving from our neighbor 192.168.212.2:

```

r1> sh ip bgp nei 192.168.212.2 routes
BGP table version is 26, local router ID is 172.16.3.1
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
*> 172.16.32.0/19	192.168.212.2	0		0	2 i
*> 172.16.64.0/19	192.168.212.2			0	2 3 i
* 172.16.96.0/19	192.168.212.2			0	2 3 4 i

Here is our routing table. The first character indicates which routing protocol inserted each route where B=BGP, C=connected, and S=static. You can see the /19 CIDR block advertisements learned from BGP only for the other routers.

```

r1#show ip route 172.16.0.0
Routing entry for 172.16.0.0/16, 8 known subnets
  Attached (4 connections)
  Variably subnetted with 2 masks

B       172.16.128.0/19 [20/0] via 192.168.215.5, 00:03:19
B       172.16.32.0/19 [20/0] via 192.168.212.2, 01:12:04
S       172.16.0.0/19 is directly connected, Null0
C       172.16.1.0/24 is directly connected, Loopback1
C       172.16.2.0/24 is directly connected, Loopback2
C       172.16.3.0/24 is directly connected, Loopback3
B       172.16.96.0/19 [20/0] via 192.168.215.5, 00:03:19
B       172.16.64.0/19 [20/0] via 192.168.212.2, 01:12:04

```

Here are our BGP routes to network 172.16.64.0/19. We have two routes, each with a different ASPATH, “2 3” and “5 4 3”. The former is selected as “best” because the ASPATH is shorter.

```

r1#sh ip bgp 172.16.64.0
BGP routing table entry for 172.16.64.0/19, version 4
Paths: (2 available, best #1, advertised over EBGP)
 2 3
   192.168.212.2 from 192.168.212.2 (172.16.35.2)
     Origin IGP, valid, external, best
 5 4 3
   192.168.215.5 from 192.168.215.5 (172.16.131.5)
     Origin IGP, valid, external

```

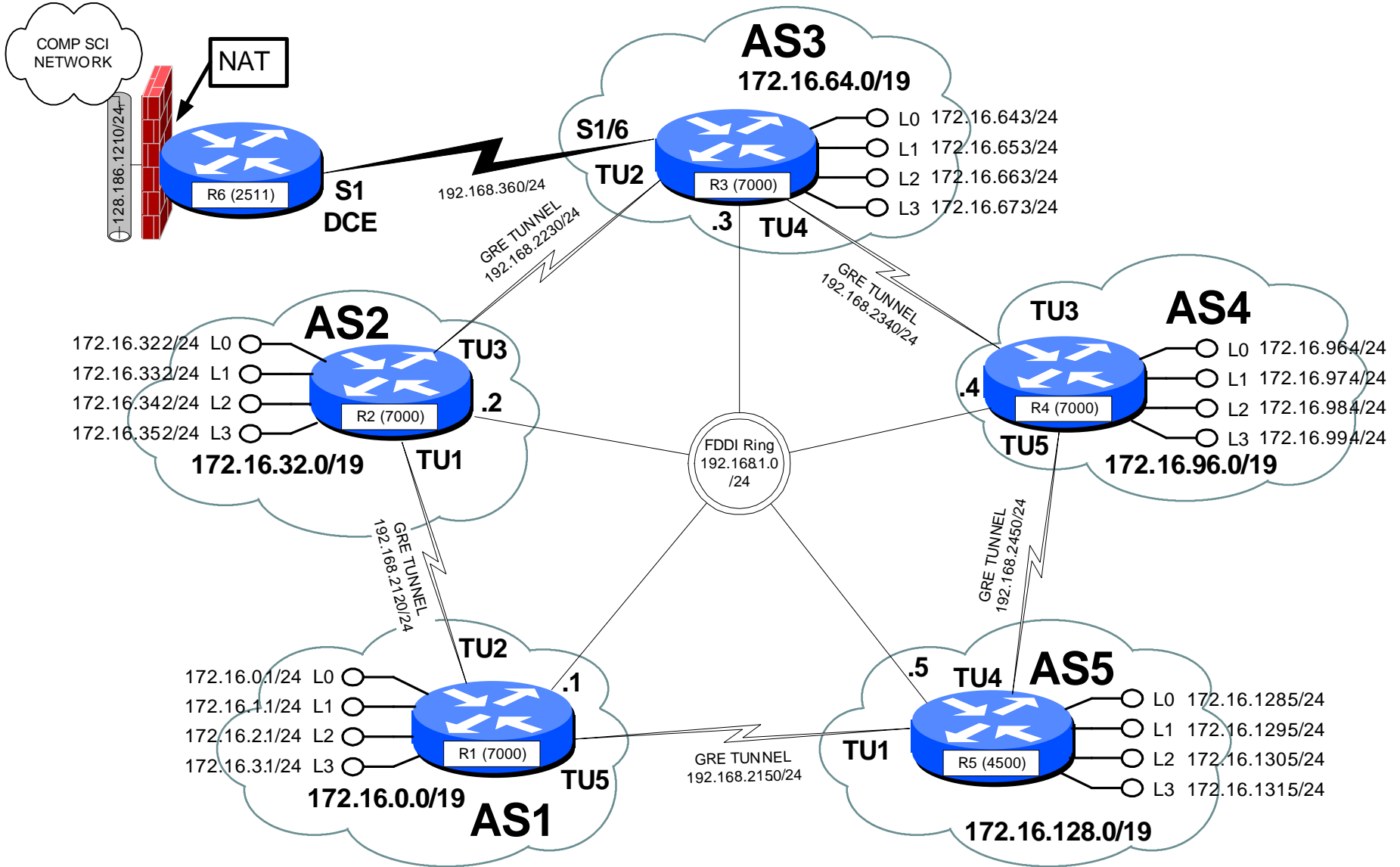
Here are all our known BGP routes including the ASPATH for each. The argument “.*” is a regular expression matching all ASPATHs.


```
rl#sh ip bgp regexp .*
BGP table version is 26, local router ID is 172.16.3.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

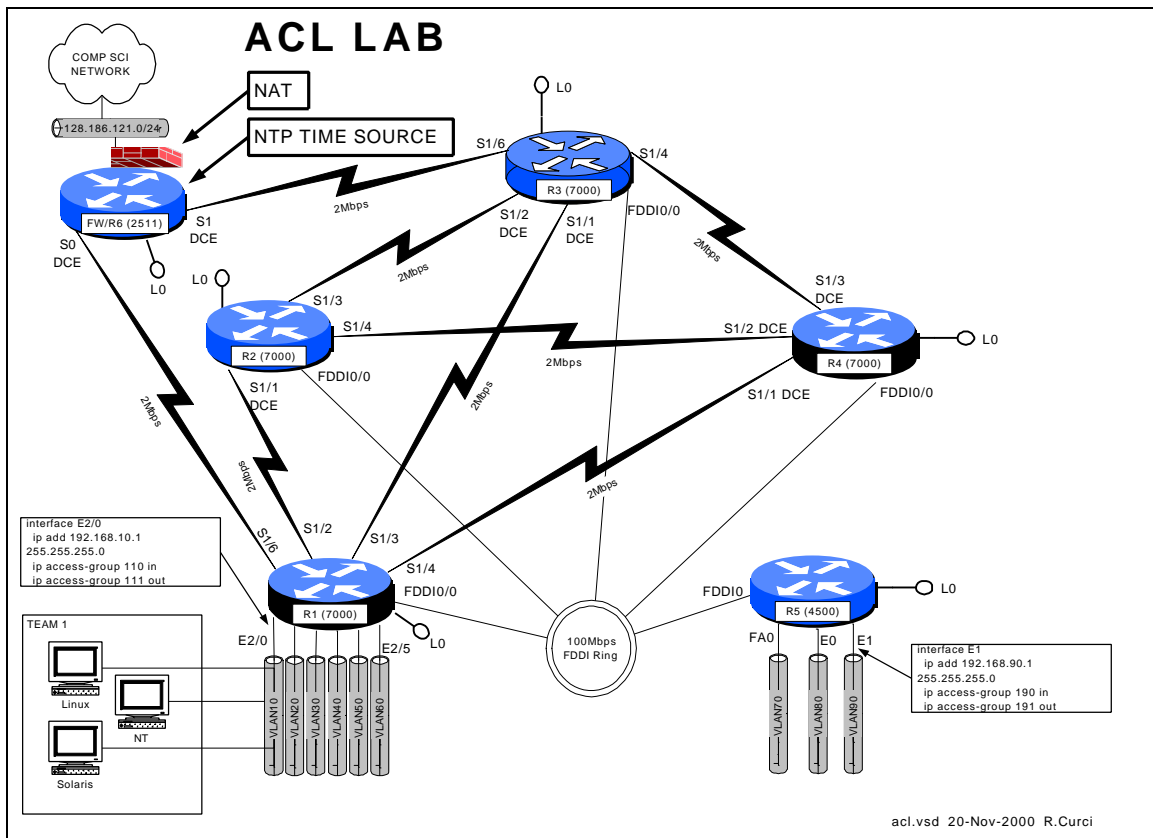
Network	Next Hop	Metric	LocPrf	Weight	Path
*> 172.16.0.0/19	0.0.0.0	0		32768	i
*> 172.16.32.0/19	192.168.212.2	0		0	2 i
*> 172.16.64.0/19	192.168.212.2			0	2 3 i
*	192.168.215.5			0	5 4 3 i
* 172.16.96.0/19	192.168.212.2			0	2 3 4 i
*>	192.168.215.5			0	5 4 i
*> 172.16.128.0/19	192.168.215.5	0		0	5 i

```
rl#
```

BGP LAB



INTERNET TEACHING LAB: ACL LAB



Overview

Access Control Lists (ACLs) can be used to selectively block IP traffic to provide a rudimentary firewall. In this lab, you will be using Cisco extended IP access lists to secure your network.

PART1 – PC Setup

Linux and Solaris:

Configure your Linux system so that syslog messages received on facility “local7” should be logged to file `/var/log/cisco.log` at all severity levels including “debug”. You will need to create the log file, modify `/etc/syslog.conf`. By default, the syslog will not accept messages from the network which requires an optional flag when invoked. See the ‘man syslogd’ for more information. You will need to modify `/etc/rc.d/init.d/syslog` to include this flag when the daemon is invoked. You may find it useful to have a Linux window open to follow the log file with “`linux# tail -f /var/log/syslog.log`”.

Download and install NTP version 3 on your UNIX systems. Configure ntpd to use the R6 loopback0 port (192.168.66.6) as your time source. You can find the software at <http://www.eecis.udel.edu/~ntp/>.

Download and install Sendmail version 8 on your UNIX systems. Configure so that you can send e-mail between your two UNIX systems. You can find the latest software at <http://www.sendmail.org>.

Download and install the Apache web server. Configure a sample default web page. You can find the software at <http://www.apache.org>.

Download and install SSH client and server. You can find this at <http://SL.us.fsu.edu> or <http://www.ssh.com>.

NT 4.0 Server:

Install the Internet Information Server (IIS) version 4. If not already loaded, you will first need to install IIS version 2 from the NT 4.0 Server distribution CD-ROM. Afterwards, update the IIS server to version 4.0 using the Windows NT 4.0 Option Pack CD-ROM. Afterwards, be sure to reinstall the latest service pack (6a as of this writing). Create a sample default web page and verify you can access it from a web browser on another system.

Download and install an SSH client. You can find this at <http://SL.us.fsu.edu> or <http://www.ssh.com>.

PART2 – Baseline Configuration

Begin with the following baseline router configuration. You should be able to copy and paste the common configuration and router specific configuration into your router's configuration as appropriate.

```
COMMON:
service udp-small-servers
service tcp-small-servers
enable password cisco
no ip domain-lookup
no ip classless
logging buffered
snmp-server community public RO
line con 0
  exec-timeout 0 0
line aux 0
line vty 0 4
  password cisco
  login
no shutdown
interface Fddi0/0
  ip address 192.168.1.1 255.255.255.0
  no shutdown
interface Serial1/2
  description Link to R2 S1/1
  ip address 192.168.12.1 255.255.255.0
  bandwidth 2000
  no shutdown
interface Serial1/3
  description Link to R3 S1/1
  ip address 192.168.13.1 255.255.255.0
  bandwidth 2000
  no shutdown
interface Serial1/4
  description Link to R4 S1/1
  ip address 192.168.14.1 255.255.255.0
  bandwidth 2000
  no shutdown

R1:
hostname r1
interface Loopback0
  ip address 192.168.11.1 255.255.255.0
```

```

interface Serial1/6
  description Link to R6 S0
  ip address 192.168.16.1 255.255.255.0
  bandwidth 2000
  no shutdown
interface E2/0
  description Vlan 10 to cat1 FA0/1
  ip address 192.168.10.1 255.255.255.0
  no shutdown
interface E2/1
  description Vlan 20 to cat1 FA0/2
  ip address 192.168.20.1 255.255.255.0
  no shutdown
interface E2/2
  description Vlan 30 to cat1 FA0/3
  ip address 192.168.30.1 255.255.255.0
  no shutdown
interface E2/3
  description Vlan 40 to cat1 FA0/4
  ip address 192.168.40.1 255.255.255.0
  no shutdown
interface E2/4
  description Vlan 50 to cat1 FA0/5
  ip address 192.168.50.1 255.255.255.0
  no shutdown
interface E2/5
  description Vlan 60 to cat1 FA0/6
  ip address 192.168.60.1 255.255.255.0
  no shutdown
router rip
  network 192.168.11.0
  network 192.168.12.0
  network 192.168.13.0
  network 192.168.14.0
  network 192.168.16.0
  network 192.168.1.0
  network 192.168.10.0
  network 192.168.20.0
  network 192.168.30.0
  network 192.168.40.0
  network 192.168.50.0
  network 192.168.60.0

```

R2:

```

hostname r2
interface Loopback0
  ip address 192.168.22.2 255.255.255.0
  no shutdown
interface Fddi0/0
  ip address 192.168.1.2 255.255.255.0
  no shutdown
interface Serial1/1
  description Link to R1 S1/2
  ip address 192.168.12.2 255.255.255.0
  bandwidth 2000
  clockrate 2000000
  no shutdown
interface Serial1/3
  description Link to R3 S1/2
  ip address 192.168.23.2 255.255.255.0
  bandwidth 2000
  no shutdown
interface Serial1/4
  description Link to R4 S1/2
  ip address 192.168.24.2 255.255.255.0
  bandwidth 2000
  no shutdown
router rip
  network 192.168.12.0
  network 192.168.22.0

```

```

network 192.168.23.0
network 192.168.24.0
network 192.168.1.0

```

R3:

```

hostname r3
interface Loopback0
  ip address 192.168.33.3 255.255.255.0
  no shutdown
interface Fddi0/0
  ip address 192.168.1.3 255.255.255.0
  no shutdown
interface Serial1/0
  description Link to self
  no ip address
  bandwidth 2000
  no shutdown
interface Serial1/1
  description Link to R1 S1/3
  ip address 192.168.13.3 255.255.255.0
  bandwidth 2000
  clockrate 2000000
  no shutdown
interface Serial1/2
  description Link to R2 S1/3
  ip address 192.168.23.3 255.255.255.0
  bandwidth 2000
  clockrate 2000000
  no shutdown
interface Serial1/3
  description Link to self
  no ip address
  bandwidth 2000
  clockrate 2000000
  no shutdown
interface Serial1/4
  description Link to R4 S1/3
  ip address 192.168.34.3 255.255.255.0
  bandwidth 2000
  no shutdown
interface Serial1/6
  description Link to R6 S1
  ip address 192.168.36.3 255.255.255.0
  bandwidth 2000
  no shutdown
router rip
  network 192.168.33.0
  network 192.168.13.0
  network 192.168.23.0
  network 192.168.34.0
  network 192.168.36.0
  network 192.168.1.0

```

R4:

```

hostname r4
interface Loopback0
  ip address 192.168.44.4 255.255.255.0
  no shutdown
interface Fddi0/0
  description Link to R5 FDDI0
  ip address 192.168.1.4 255.255.255.0
  no shutdown
interface Serial1/1
  description Link to R1 S1/4
  ip address 192.168.14.4 255.255.255.0
  bandwidth 2000
  clockrate 2000000
  no shutdown
interface Serial1/2
  description Link to R2 S1/4
  ip address 192.168.24.4 255.255.255.0

```

```

bandwidth 2000
clockrate 2000000
no shutdown
interface Serial1/3
description Link to R3 S1/4
ip address 192.168.34.4 255.255.255.0
bandwidth 2000
clockrate 2000000
no shutdown
router rip
network 192.168.44.0
network 192.168.14.0
network 192.168.24.0
network 192.168.34.0
network 192.168.1.0

R5:
hostname r5
interface loopback0
ip address 192.168.55.5 255.255.255.0
no shutdown
interface FastEthernet0
description Vlan70 to cat1 FA0/7

ip address 192.168.70.1 255.255.255.0
media-type 100BaseX
no shutdown
interface Ethernet0
description Vlan80 to cat1 FA0/8
ip address 192.168.80.1 255.255.255.0
media-type 10BaseT
no shutdown
interface Ethernet1
description Vlan90 to cat1 FA0/9
ip address 192.168.90.1 255.255.255.0
media-type 10BaseT
no shutdown
interface Fddi0
description Link to R4 FDDI0/0
ip address 192.168.1.5 255.255.255.0
no keepalive
no shutdown
router rip
network 192.168.55.0
network 192.168.70.0
network 192.168.80.0
network 192.168.90.0
network 192.168.1.0

```

PART3 – NTP and SYSLOG

Configure your router to sync its clock using the network time protocol with the clock on router r6/fw. Use the r6 loopback0 address, 192.168.66.6. Use “show ntp association” and “show ntp status” to test. Configure your router for the appropriate timezone and daylight savings time with the “clock” configuration command. We are in the Eastern time zone which is –5 hours different than UTC/GMT and use EDT in the summer. Use the “show clock” command to verify you have it working correctly.

Now that you have an accurate clock, configure the router so that log messages and debug messages will prepend the local date, time, and timezone using the “service timestamp” configuration command.

Configure your router to generate SYSLOG messages to your Linux syslog server. Use the default “local7” facility and log all messages including those with severity level debug. You will need the “logging” and “logging trap” configuration commands. Verify your router settings with “show log”. Once you have it configured, turn on some debug messages such as “debug ntp packets” and verify you see the messages on your Linux syslog file /var/log/cisco.log. Remember to turn off debugging with “undebg all”.

PART4 – Access Control Lists

Extended IP access lists numbered between 100 through 199. Your team’s VLAN should connect to a router Ethernet or fast Ethernet port. Create two extended IP access lists and apply one to your ethernet port input and other to your ethernet port output as follows:

```
interface [ethernetX|fastethernetX]
  ip access-group XXX in
  ip access-group YYY out
```

Where XXX = (100 + 10 x TEAM) and YYY = (101 + 10 x TEAM):

TEAM	INPUT ACL	OUTPUT ACL
1	110	111
2	120	121
3	130	131
4	140	141
5	150	151
6	160	161
7	170	171
8	180	181
9	190	191

(The terms Input and Output are relative to your router's ethernet port. The terms "host" and "server" are synonymous in this context.)

Create two IP extended access lists for the input and output of your gateway router's ethernet interface to your team VLAN and apply to your ethernet or fast ethernet port with the following security policy:

Security Policy:

- Hosts on your VLAN should generally be able to access services outside your VLAN provided the services are not outside the FSU network. (FSU networks 128.186.0.0/16, 146.201.0.0/16, and 144.174.0.0/16 and RFC1918 private address space 192.168.0.0/16, 172.16.0.0/12, and 10.0.0.0/8 should be permitted).
- Do not allow any spoofed packets into your VLAN.
- Allow all NETBIOS over TCP/IP traffic.
- Allow all DNS, NTP, TFTP, SNMP, SYSLOG, and RIP v1 datagrams. (Do not worry about SNMP traps or DNS zone transfers).
- Allow TCP DISCARD and TTCP/IPERF packets for testing.
- Allow all ICMP packets for testing.
- Allow all shell (ssh), and web (www/http) access to hosts on your VLAN (Do not worry about secure http).
- Allow e-mail access (smtp,pop3,imap) to only your Linux server.
- Allow TELNET access to your servers if sourced from a trusted group's VLAN. All even groups only trust each other. All odd groups only trust each other.
- Disallow any other TELNET access from unauthorized IP addresses
- Deny everything else.

- All disallowed traffic must be logged to your Linux host using syslog on file /var/log/cisco.log

You can find out TCP/IP port number assignments from the Internet Assigned Numbers Authority, <http://www.isi.edu/in-notes/iana/assignments/port-numbers>. The relevant assignments are also included in the table below.

service	protocol	port	description
discard	tcp	9	Bit Bucket/Discard Protocol for Testing
ssh	tcp	22	SSH Remote Login Protocol
telnet	tcp	23	Telnet
smtp	tcp	25	Simple Mail Transfer Protocol
dns	udp	53	Domain Name Server
tftp	udp	69	Trivial File Transfer Protocol
http/www	tcp	80	HyperText Transport Protocol (WWW)
pop3	tcp	110	Post Office Protocol version 3
ntp	udp	123	Network Time Protocol
netbios-ns	tcp	137	NETBIOS Name Service
netbios-ns	udp	137	NETBIOS Name Service
netbios-dgm	tcp	138	NETBIOS Datagram Service
netbios-dgm	udp	138	NETBIOS Datagram Service
netbios-ssn	tcp	139	NETBIOS Session Service
netbios-ssn	udp	139	NETBIOS Session Service
imap4	tcp	143	Internet Message Access Protocol
snmp	udp	161	Simple Network Management Protocol
syslog	udp	514	System Log Messages
rip	udp	520	Routing Information Protocol
ttcp/iperf	tcp	5001	Test TCP / IPERF Testing Protocol

Example of how to apply an access list to an ethernet interface and converting the policy into a detailed intermediate form before coding the access lists:

```
interface ethernet0
  ip address 192.168.10.1 255.255.255.0
  ip access-group 110 in
  ip access-group 111 out
```

Input access list 110:

1. Allow all traffic, provided the destination is in RFC1918 private address space or one of FSU's three class B addresses:
 - a. 192.168.0.0/16
 - b. 172.16.0.0/12
 - c. 10.0.0.0/8
 - d. 128.186.0.0/16

- e. 146.201.0.0/16
- f. 144.174.0.0/16
- 2. Deny everything else and log it.

Output access list 111:

- 1. Allow all established TCP connections
- 2. Deny forged packets with IP source address on your VLAN and log it.
- 3. Allow all Microsoft NetBIOS name, datagram, and session traffic (137/udp, 138/udp, 139/udp, 137/tcp, 138/tcp, 139/tcp).
- 4. Allow all DNS, NTP, TFTP, SNMP, SYSLOG, and RIP datagrams (53/udp, 123/udp, 69/udp, 161/udp, 514/udp, 520/udp).
- 5. Allow TCP DISCARD and TFTP/IPERF packets (9/tcp, 5001/tcp).
- 6. Allow all ICMP packets.
- 7. Allow all TCP SSH and WWW to our VLAN. (22/tcp, 80/tcp)
- 8. Allow SMTP, POP3, and IMAP only to our Linux server (25/tcp, 110/tcp, 143/tcp).
- 9. Allow all TELNET (23/tcp) access from trusted VLAN IP addresses.
- 10. Deny all other (23/tcp) TELNET and log it.
- 11. Deny everything else and log it.

PART5 – Verification

Verify that your access lists are working. The following are some examples of tests that can be performed on the routers and Linux PC for partly testing out your access lists.

PING packets use ICMP protocol and should work from your PC to an FSU destination, but fail to an outside destination:

```
[curci@s1 curci]$ ping www.cnn.com.
PING cnn.com (207.25.71.24) from 192.168.10.2 : 56(84) bytes of data.
From 192.168.10.1: Packet filtered
From 192.168.10.1: Packet filtered
. . .
--- cnn.com ping statistics ---
5 packets transmitted, 0 packets received, +5 errors, 100% packet loss

[curci@s1 curci]$ ping nu.cs.fsu.edu
PING nu.cs.fsu.edu (128.186.121.10) from 192.168.10.2 : 56(84) bytes of
data.
64 bytes from nu (128.186.121.10):icmp_seq=0 ttl=253 time=4.6 ms
64 bytes from nu (128.186.121.10):icmp_seq=1 ttl=253 time=4.3 ms
64 bytes from nu (128.186.121.10): icmp_seq=2 ttl=253 time=4.2 ms
--- nu.cs.fsu.edu ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 4.2/4.3/4.6 ms
[curci@s1 curci]$
```

Ping should also work from outside your Vlan from r6 to your Linux server:

```
fw/r6#ping 192.168.10.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/7/12 ms
fw/r6#
```

Test NTP protocol by syncing Linux server clock to ntp server on r6 loopback address 192.168.66.6 using the ntpdate utility:

```
[root@s1 curci]# /usr/sbin/ntpdate -v 192.168.66.6
22 Nov 23:23:33 ntpdate[1826]: ntpdate 3-5.93e Fri Feb 18
                               18:55:19 EST 2000 (1)
22 Nov 23:23:33 ntpdate[1826]: adjust time server 192.168.66.6
                               offset 0.001193 sec
```

Test SNMP protocol by fetching the system.sysName.0 MIB variable from r6:

```
[root@s1 curci]# snmpget -v 1 192.168.66.6 public system.sysName.0
system.sysName.0 = fw/r6
```

Test DNS datagram traffic by fetching the SOA record for domain cs.fsu.edu from nu.cs.fsu.edu:

```
[root@s1 curci]# nslookup
> lserver nu.cs.fsu.edu.
Default Server:  nu.cs.fsu.edu
Address:  128.186.121.10

> set type=SOA
> cs.fsu.edu.

fsu.edu
      origin = dns1.fsu.edu
      mail addr = hostmaster.acns.fsu.edu
      serial = 2000112203
      refresh = 3600 (1H)
      retry = 1200 (20M)
      expire = 604800 (1W)
      minimum ttl = 86400 (1D)
>
```

From Linux PC, test iperf client using discard TCP port 9 on r6:

```
[root@s1 curci]# iperf -c 192.168.66.6 -p 9
-----
Client connecting to 192.168.66.6, TCP port 9
TCP window size: 64.0 KByte (default)
-----
[  3] local 192.168.10.2 port 2690 connected with 192.168.66.6 port 9
[ ID] Interval      Transfer      Bandwidth
[  3] 0.0-10.3 sec    1.5 MBytes    1.1 Mbits/sec
[root@s1 curci]#
```

From the Linux PC, test access to an outside FSU web page
<http://www.cs.fsu.edu/~curci>:

```

[root@s1 curci]# telnet www.cs.fsu.edu 80
Trying 128.186.121.41...
Connected to xi.cs.fsu.edu.
Escape character is '^]'.
GET /~curci/

<html>
<head><title>Ray Curci Home Page</title></head>
<body>Ray Curci Home Page 16-Nov-2000</p>
I am presently working on an MS degree in the FSU Computer
Network and Systems Administration track.
</body></html>
Connection closed by foreign host.
[root@s1 curci]#

```

Your team VLAN should connect to an ethernet port on either r1 or r5. If you go to r1 or r5, whichever does not connect to your VLAN, you can execute TELNET sourced from a trusted and untrusted group to verify the access list. For example, I am on team 1 served from router r1 interface ethernet 2/0, and my Linux server is at IP address 192.168.10.2. (Vlan10). If try to telnet to my Linux PC from r5 and source from team 8's untrusted ethernet port Ethernet0 it should fail, but work if sourced from team 9's trusted ethernet port Ethernet1, it should work and I will see the login prompt:

```

(Sourced from r5 Ethernet0, ip address 192.168.80.1 (untrusted))
r5#telnet 192.168.10.2 /source-interface Ethernet0
Trying 192.168.10.2 ...
% Destination unreachable; gateway or host down

(Sourced from r5 Ethernet1, ip address 192.168.90.1 (trusted))
r5#telnet 192.168.10.2 /source-interface Ethernet1
Trying 192.168.10.2 ... Open

Red Hat Linux release 6.2 (Zoot)
Kernel 2.2.14-5.0 on an i586
login:

```

My my Linux syslog server in logfile /var/log/cisco.log, the denied telnet attempt from 192.168.80.1 appears. There are four fields in this message (1) time/date stamp prepended by the Linux syslogd program, (2) IP address of device that sent the message, r1's ethernet 2/0 port, prepended by Linux syslogd, (3) time/date stamp prepended by router r1, and (4) the log message itself indicating a denied TCP packet from 192.168.80.1 port 11000 to 192.168.10.2 port 23 (telnet port):

```

Nov 22 23:43:54 192.168.10.1 63: Nov 22 23:43:53 EST:
%SEC-6-IPACCESSLOGP: list 111 denied
tcp 192.168.80.1(11000) -> 192.168.10.2(23), 1 packet

```

From outside, I should be able to access the WWW server on my Linux system (192.168.10.2) or NT system at 192.168.10.3:

```

fw/r6#telnet 192.168.10.2 80
Trying 192.168.10.2, 80 ... Open
GET /
<html><head><title>S1 Sample WWW Page</title></head><body>

```

```
<h1>S1 Sample WWW Page</h1>
<hr>This is a test WWW page on server S1 Linux Redhat 6.2 Server
<hr></body></html>
[Connection to 192.168.10.2 closed by foreign host]

fw/r6#telnet 192.168.10.3 80
Trying 192.168.10.3, 80 ... Open
GET /
<html><head><title>S2 Sample WWW Page</title></head>
<body><h1>S2 Sample WWW Page</h1><hr>
This is a test WWW page on server S2 Windows NT 4.0 Server
<hr></body></html>
[Connection to 192.168.10.3 closed by foreign host]
fw/r6#
```

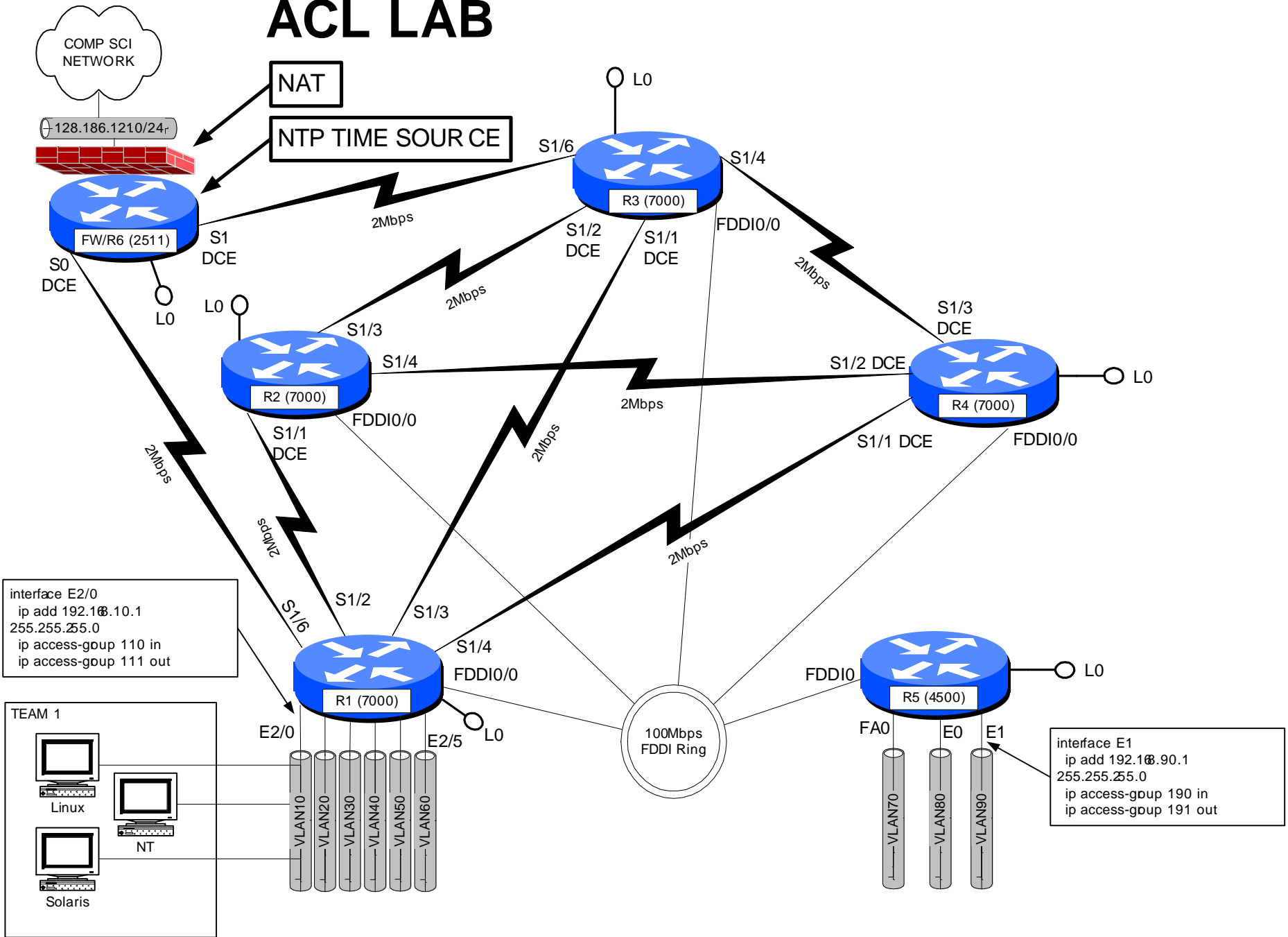
From outside on r6, I should be able to access my Linux system 192.168.10.2 with SMTP e-mail:

```
fw/r6#telnet 192.168.10.2 25
Trying 192.168.10.2, 25 ... Open
220 s1.egghead.net ESMTTP Sendmail 8.9.3/8.9.3; Wed, 22 Nov 2000 23:50:05
-0500
quit
221 s1.egghead.net closing connection
[Connection to 192.168.10.2 closed by foreign host]
```

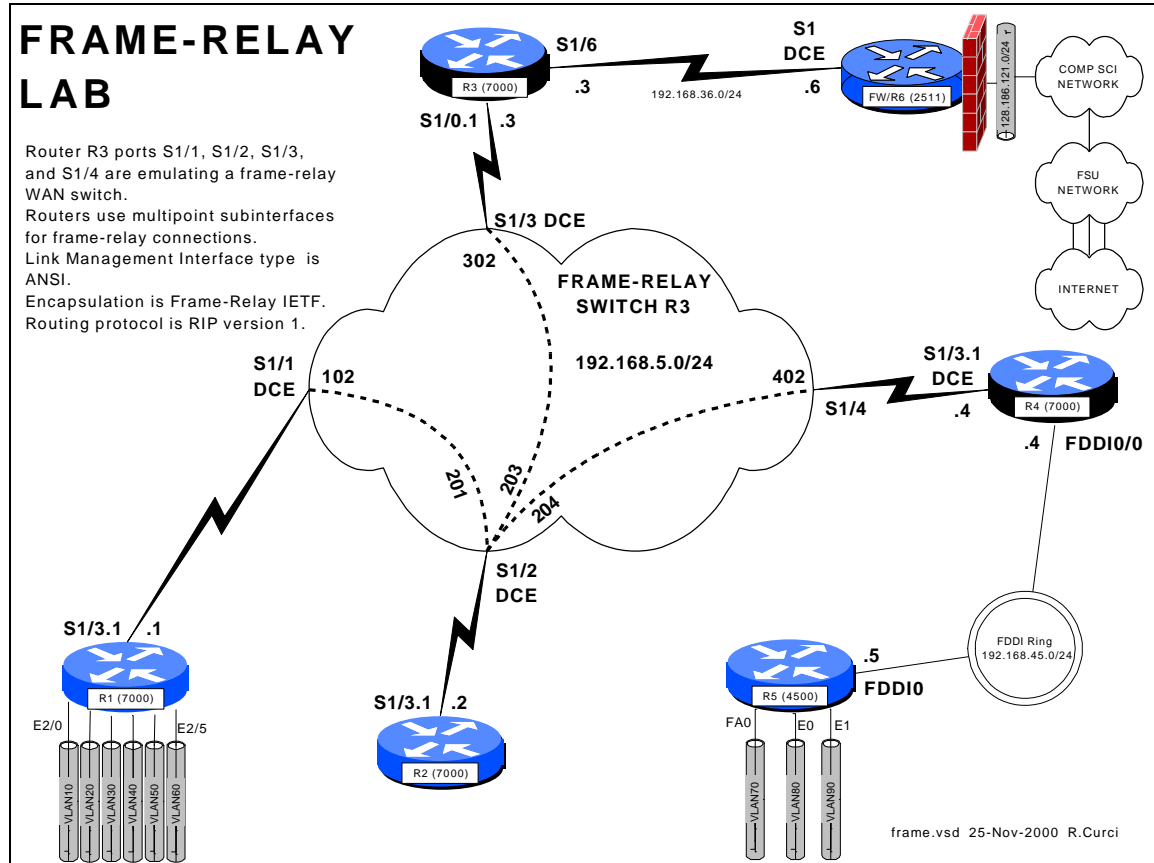
Here is an excerpt from “show access-list 111”. Note that some lines have been matched and the number of matches are displayed:

```
r1# show access-list 111
. . .
  permit udp any eq domain any (79 matches)
  permit udp any any eq ntp (8 matches)
. . .
```

ACL LAB



INTERNET TEACHING LAB: FRAME-RELAY LAB



Overview

In this lab, we will explore the frame-relay data link protocol. Frame-relay is widely deployed by phone companies in wide area networks (WANs) and related to the X.25 and ATM protocols. Routers or frame-relay access devices (FRADs) have a physical serial connection to a service provider's nearest frame-relay switch typically across a T1 or digital data service (DDS) circuit. Usually, the service provider will have several interconnected frame-relay switches depicted in diagrams as a cloud. A state-wide service provider in Florida, for example, would typically have a frame-relay switch in each of Florida's ten LATAs. Since an end user data circuit to the nearest frame-relay switch would be intralata (will not cross a LATA boundary), the cost for the "local loop" is greatly reduced. Within the frame network, permanent virtual circuits (PVCs) are created. The PVC endpoints are identified by data link channel identifiers (DLCIs) represented by integers in the range [16..1007]. Although possible to build a full mesh of PVCs in the frame network, this is rarely done because there is usually a recurring cost associated with each PVC and with N nodes, the number of PVCs required, $N(N-1)/2$ becomes large quickly. A more common configuration is a logical "hub-and-spoke" topology. In this lab, r2 will be the hub, while r1, r3, and r4 will be spokes. (Router r5 will not have a frame-relay connections because it has no serial WAN interfaces.)

Frame-relay switches also use a control protocol called the link management interface (LMI) used to inform routers what DLCIs are defined and their status.

ASSIGNMENT:

In this lab, you will be given a partially broken router configuration with 3 problems that need to be identified and solved:

1. The frame-relay DLCIs by default are associated with the router physical interfaces but in this exercise need to be associated with the subinterfaces. For example, on r4, the DLCI 402 should be associated with the multipoint subinterface Serial1/3.1 instead of physical interface Serial1/3.
2. Routers r1, r2, r3, and r4 all have their frame-relay interfaces addressed on the same 192.168.5.0/24 network, yet only some will be able to PING each other. A protocol called “inverse arp” can automatically map frame-relay DLCI numbers to IP addresses, but the mapping will be incomplete because there is not a full mesh of PVCs. You will find that R2 can PING the R1, R3, and R4 and they can PING R2, but that R1, R3, and R4 cannot PING each other.
3. Distance vector routing protocols like RIP normally do not advertise routes out an interface on which the route was learned. This behavior is called “split horizon”.

Commands that may be helpful to debug this assignment:

- show frame-relay pvc
- show frame-relay lmi
- debug frame-relay events
- debug frame-relay packets
- show ip route
- show ip protocol
- show ip interface
- show frame-relay route (useful only on R2)

Hints:

Read up on the following commands in the Cisco manuals:

- frame-relay interface-dlci
- frame-relay map ip
- ip split-horizon

Even with the partially broken configuration given, you should see LMI or Link Management Interface messages on your router. These are status messages where the frame-relay switch informs your router which DLCIs are defined and their status. You can use the “show frame-relay lmi” command. If set up correctly, you should see the number of status enquire messages sent incrementing, with an equal number of status messages received as shown below.

Good Luck!

```
r4#show frame-relay lmi

LMI Statistics for interface Serial1/3 (Frame Relay DTE) LMI TYPE = ANSI
  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 94818       Num Status msgs Rcvd 94818
  Num Update Status Rcvd 0          Num Status Timeouts 0
```

r

Router	Interface	IP Address
r1	Loopback0	192.168.11.1/24
	ethernet2/0	192.168.10.1/24
	ethernet2/1	192.168.20.1/24
	ethernet2/2	192.168.30.1/24
	ethernet2/3	192.168.40.1/24
	ethernet2/4	192.168.50.1/24
	ethernet2/5	192.168.60.1/24
	serial1/3.1	192.168.5.1/24
r2	Loopback0	192.168.22.2/24
	serial1/3.1	192.168.5.2/24
r3	Loopback0	192.168.33.3/24
	serial1/0.1	192.168.5.3/24
	serial1/6	192.168.36.3/24
r4	Loopback0	192.168.44.4/24
	fddi0/0	192.168.45.4/24
	serial1/3.1	192.168.5.4/24
r5	Loopback0	192.168.55.5/24
	fastethernet0	192.168.70.1/24
	ethernet0	192.168.80.1/24
	ethernet1	192.168.90.1/24
	fddi0	192.168.45.5/24

BROKEN ROUTER CONFIGURATION:

COMMON:

```
service udp-small-servers
service tcp-small-servers
enable password cisco
no ip domain-lookup
ip classless
ip subnet-zero
logging buffered
service timestamps debug datetime
localtime
service timestamps log datetime localtime
clock timezone EST -5
clock summer-time EDT recurring
ntp server 192.168.66.6
snmp-server community public RO
line con 0
  exec-timeout 0 0
line aux 0
line vty 0 4
  password cisco
  login
```

R1:

```
hostname r1
interface E2/0
  description Vlan 10 to cat1 FA0/1
  ip address 192.168.10.1 255.255.255.0
  no shutdown
interface E2/1
  description Vlan 20 to cat1 FA0/2
  ip address 192.168.20.1 255.255.255.0
  no shutdown
interface E2/2
  description Vlan 30 to cat1 FA0/3
  ip address 192.168.30.1 255.255.255.0
  no shutdown
interface E2/3
  description Vlan 40 to cat1 FA0/4
  ip address 192.168.40.1 255.255.255.0
  no shutdown
interface E2/4
  description Vlan 50 to cat1 FA0/5
  ip address 192.168.50.1 255.255.255.0
  no shutdown
interface E2/5
  description Vlan 60 to cat1 FA0/6
  ip address 192.168.60.1 255.255.255.0
  no shutdown
interface loopback0
  ip address 192.168.11.1 255.255.255.0
  no shutdown
interface Serial1/3
  description Frame-Relay WAN
  encapsulation frame-relay IETF
  frame-relay lmi-type ansi
  no shutdown
interface Serial1/3.1 multipoint
  ip address 192.168.5.1 255.255.255.0
  no shutdown
router rip
  network 192.168.11.0
  network 192.168.10.0
  network 192.168.20.0
  network 192.168.30.0
  network 192.168.40.0
  network 192.168.50.0
  network 192.168.60.0
  network 192.168.5.0
```

R2:

```
hostname r2
interface loopback0
  ip address 192.168.22.2 255.255.255.0
  no shutdown
interface Serial1/3
  description Frame-Relay WAN
  encapsulation frame-relay IETF
  frame-relay lmi-type ansi
  no shutdown
interface Serial1/3.1 multipoint
  ip address 192.168.5.2 255.255.255.0
  no shutdown
router rip
  network 192.168.22.0
  network 192.168.5.0
```

R3:

```
hostname r3
frame-relay switching
interface loopback0
  ip address 192.168.33.3 255.255.255.0
  no shutdown
interface Serial1/0
  description Frame-Relay WAN
  encapsulation frame-relay IETF
  frame-relay lmi-type ansi
  no shutdown
interface Serial1/0.1 multipoint
  ip address 192.168.5.3 255.255.255.0
  no shutdown
interface Serial1/1
  description Frame-Relay port to R1 S1/3
  no ip address
  encapsulation frame-relay IETF
  clockrate 2000000
  frame-relay lmi-type ansi
  frame-relay intf-type dce
  frame-relay route 102 interface
  Serial1/2 201
  no shutdown
interface Serial1/2
  description Frame-Relay port to R2 S1/3
  no ip address
  encapsulation frame-relay IETF
  clockrate 2000000
  frame-relay lmi-type ansi
  frame-relay intf-type dce
  frame-relay route 201 interface
  Serial1/1 102
  frame-relay route 203 interface
  Serial1/3 302
  frame-relay route 204 interface
  Serial1/4 402
  no shutdown
interface Serial1/3
  description Frame-Relay port to R3 S1/0
  no ip address
  encapsulation frame-relay IETF
  clockrate 2000000
  frame-relay lmi-type ansi
  frame-relay intf-type dce
  frame-relay route 302 interface
  Serial1/2 203
  no shutdown
interface Serial1/4
  description Frame-Relay port to R4 S1/3
```

```

no ip address
encapsulation frame-relay IETF
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 402 interface
Serial1/2 204
no shutdown
!
interface serial1/6
descr Serial link to R6 S1 toward
Internet
ip address 192.168.36.3 255.255.255.0
no shutdown
router rip
network 192.168.36.0
network 192.168.33.0
network 192.168.5.0

```

R4:

```

hostname r4
interface loopback0
ip address 192.168.44.4 255.255.255.0
no shutdown
interface fddi0/0
descr Link to R5 FDDI0
ip address 192.168.45.4 255.255.255.0
no shutdown
interface Serial1/3
description Frame-Relay WAN
encapsulation frame-relay IETF
frame-relay lmi-type ansi
clock rate 2000000
no shutdown
interface Serial1/3.1 multipoint
ip address 192.168.5.4 255.255.255.0
no shutdown

```

```

router rip
network 192.168.44.0
network 192.168.45.0
network 192.168.5.0

```

R5:

```

hostname r5
interface FastEthernet0
description Vlan70 to cat1 FA0/7
ip address 192.168.70.1 255.255.255.0
media-type 100BaseX
no shutdown
interface Ethernet0
description Vlan80 to cat1 FA0/8
ip address 192.168.80.1 255.255.255.0
media-type 10BaseT
no shutdown
interface Ethernet1
description Vlan90 to cat1 FA0/9
ip address 192.168.90.1 255.255.255.0
media-type 10BaseT
no shutdown
interface Fddi0
description Link to R4 FDDI0/0
ip address 192.168.45.5 255.255.255.0
no shutdown
interface loopback0
ip address 192.168.55.5 255.255.255.0
no shutdown
router rip
network 192.168.70.0
network 192.168.80.0
network 192.168.90.0
network 192.168.45.0
network 192.168.55.0

```

FRAME-RELAY LAB

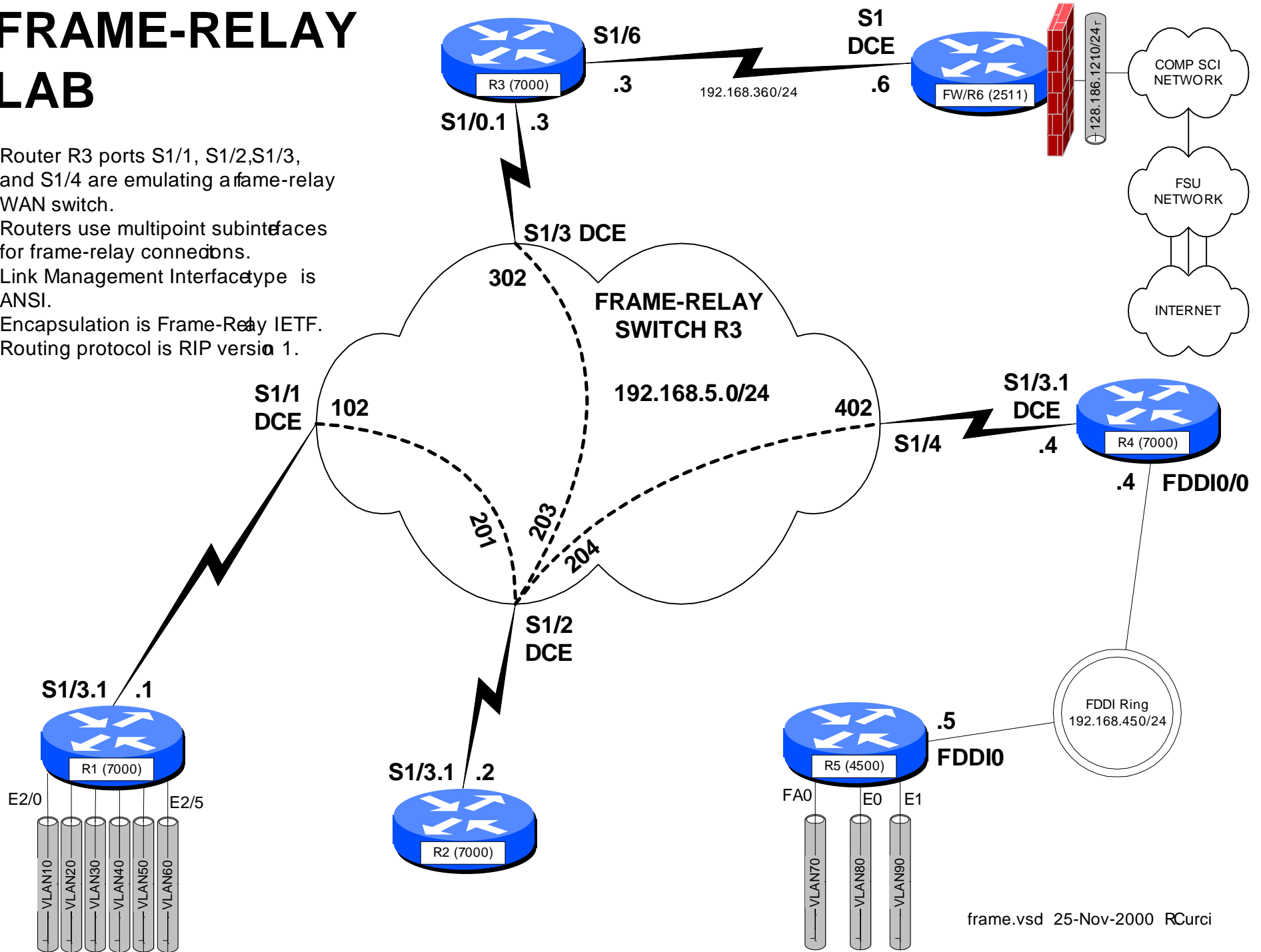
Router R3 ports S1/1, S1/2, S1/3, and S1/4 are emulating a frame-relay WAN switch.

Routers use multipoint subinterfaces for frame-relay connections.

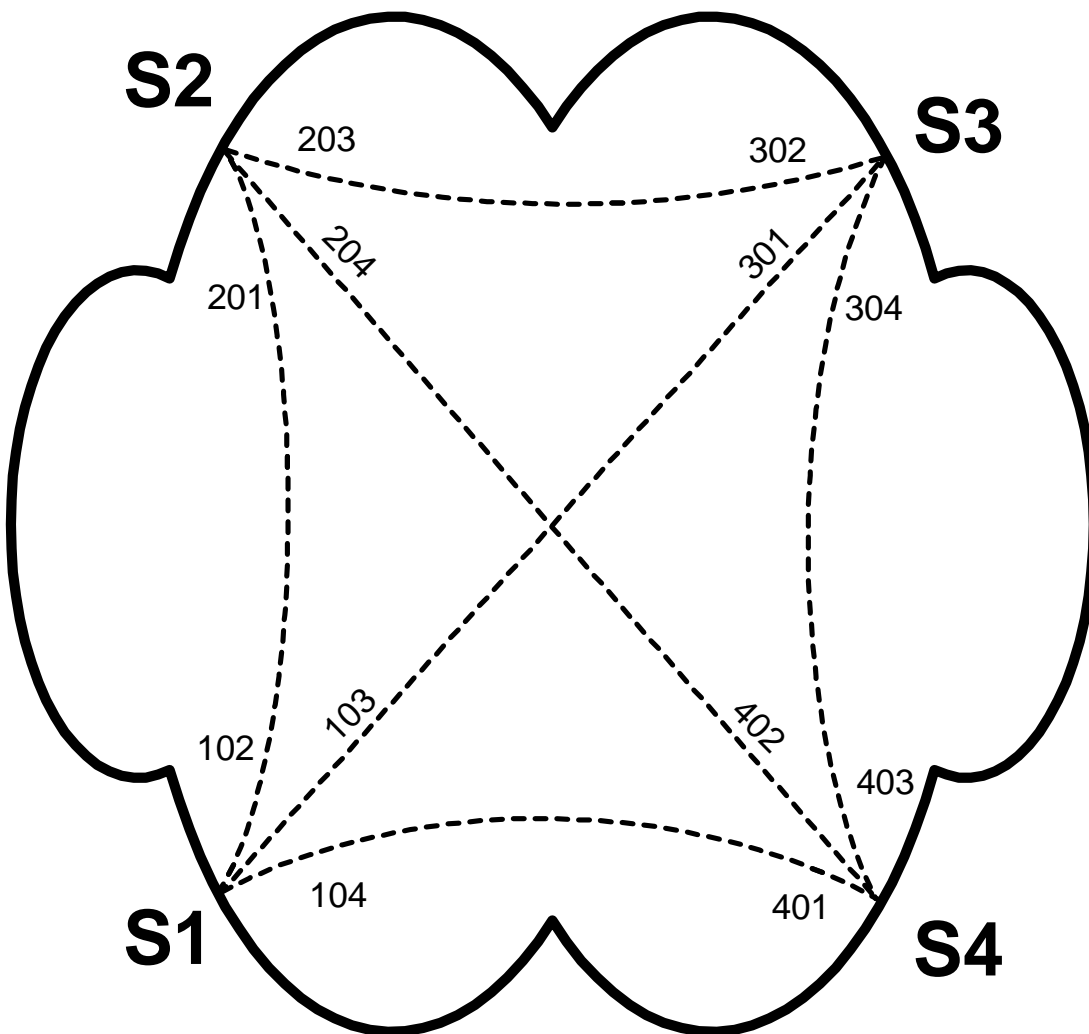
Link Management Interface type is ANSI.

Encapsulation is Frame-Relay IETF.

Routing protocol is RIP version 1.



FRAME-RELAY PVCs



```

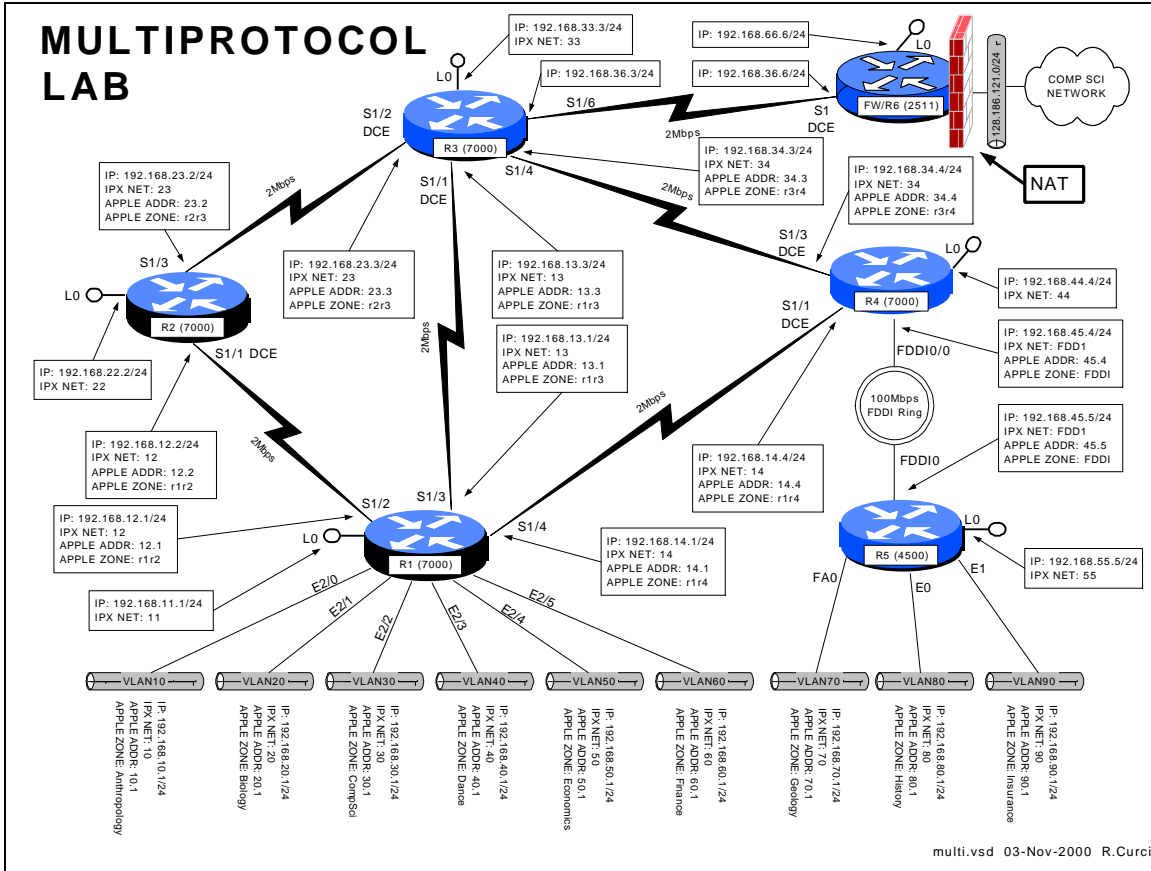
! Cisco Router Config to simulate a
! fully meshed Frame-Relay WAN
!
frame-relay switching
!
interface Serial1
description Frame-Relay port to R1
no ip address
encapsulation frame-relay IETF
clockrate 5000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 102 interface Serial2 201
frame-relay route 103 interface Serial3 301
frame-relay route 104 interface Serial4 401
!
interface Serial2
description Frame-Relay port to R2
no ip address
encapsulation frame-relay IETF
clockrate 5000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 201 interface Serial1 102
frame-relay route 203 interface Serial3 302
frame-relay route 204 interface Serial4 402
!
interface Serial3
description Frame-Relay port to R3
no ip address
encapsulation frame-relay IETF
clockrate 5000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 301 interface Serial1 103
frame-relay route 302 interface Serial2 203
frame-relay route 304 interface Serial4 403
!
interface Serial4
description Frame-Relay port to R4
no ip address
encapsulation frame-relay IETF
clockrate 5000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 401 interface Serial1 104
frame-relay route 402 interface Serial2 204
frame-relay route 403 interface Serial3 304
!

```

FSU Internet Teaching Lab

framepvc.vsd 25-Nov-2000R.Curci

INTERNET TEACHING LAB: MULTIPROTOCOL LAB



Overview

This lab explores two popular non-IP protocols: Novell's IPX and Apple's Appletalk. IPX is a modified version of the Xerox XNS protocol adapted for use on Novell file servers. Today, this protocol is also supported under Windows and Linux. Appletalk was designed for use on the Apple Macintosh computer and Apple LaserWriter printers. It is supported under Linux and partly supported under Windows NT. For example, Windows NT has support to act as a native IPX file server or a native Appletalk File server (called AppleShare in Apple terminology). This allows IPX and Apple devices to access the server without the need for additional software.

Part 1 – IPX

IPX network addresses are composed of a 32-bit network address and a 48-bit host address. The syntax is often abbreviated N.H.H.H and written in hexadecimal. Addresses can be entered in the form "NNNNNNNN.HHHH.HHHH.HHHH" but leading zeros can be omitted. IPX routing is turned on with the global router command

“ipx routing HHHH.HHHH.HHHH” where HHHH.HHHH.HHHH is a host identifier for your router. If present, this address will be used on interfaces that do not have any MAC address like serial lines. If omitted, the router will make up an address or use one from an active ethernet port. Embed your router name in the address to make things like routing table listings a little easier to read. For example, if you are programming router r4, turn on ipx with the global command “ipx routing 4.4.4”. Once the IPX routing process is running on the router, you will need to add an IPX network address to each interface you want to speak IPX. Use the interface command “ipx network NNNNNNNN” in hexadecimal. Follow the diagram above carefully to add IPX routing. The Cisco routers have an IPX PING command that is helpful to verify connectivity.

```
r5#ping
Protocol [ip]: ipx
Target IPX address: 11.1.1.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Verbose [n]:
Type escape sequence to abort.
Sending 5, 100-byte IPXcisco Echoes to 11.0001.0001.0001, timeout is 2
seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms
```

Use the following commands to help with debugging your configuration.

- show ipx route
- show ipx interface
- show ipx interface brief
- show ipx traffic
- show ipx server
- show ipx server detailed

Part 2 – Appletalk

Appletalk Phase II addresses of a 16-bit network address and 8-bit host address. Host addresses 0, 254, and 255 are reserved, so you can only use host addresses 1 through 253 in your network. In order to provision networks supporting more than 253 hosts, you can specify a range of consecutive network addresses (called “cable-range” in Apple terminology), but it will not be necessary in this lab. Appletalk normally will dynamically select an unused host number, however, we will be specifying it manually so it will be easier to test the network with tools like PING. Appletalk also uses the concept of a “zone” to logically name the networks. A single zone name may belong to multiple network segments, and a single network segment may have multiple zones, but only a single default zone. Zone names can include whitespace and non alphanumeric characters and are case sensitive, so type them carefully.

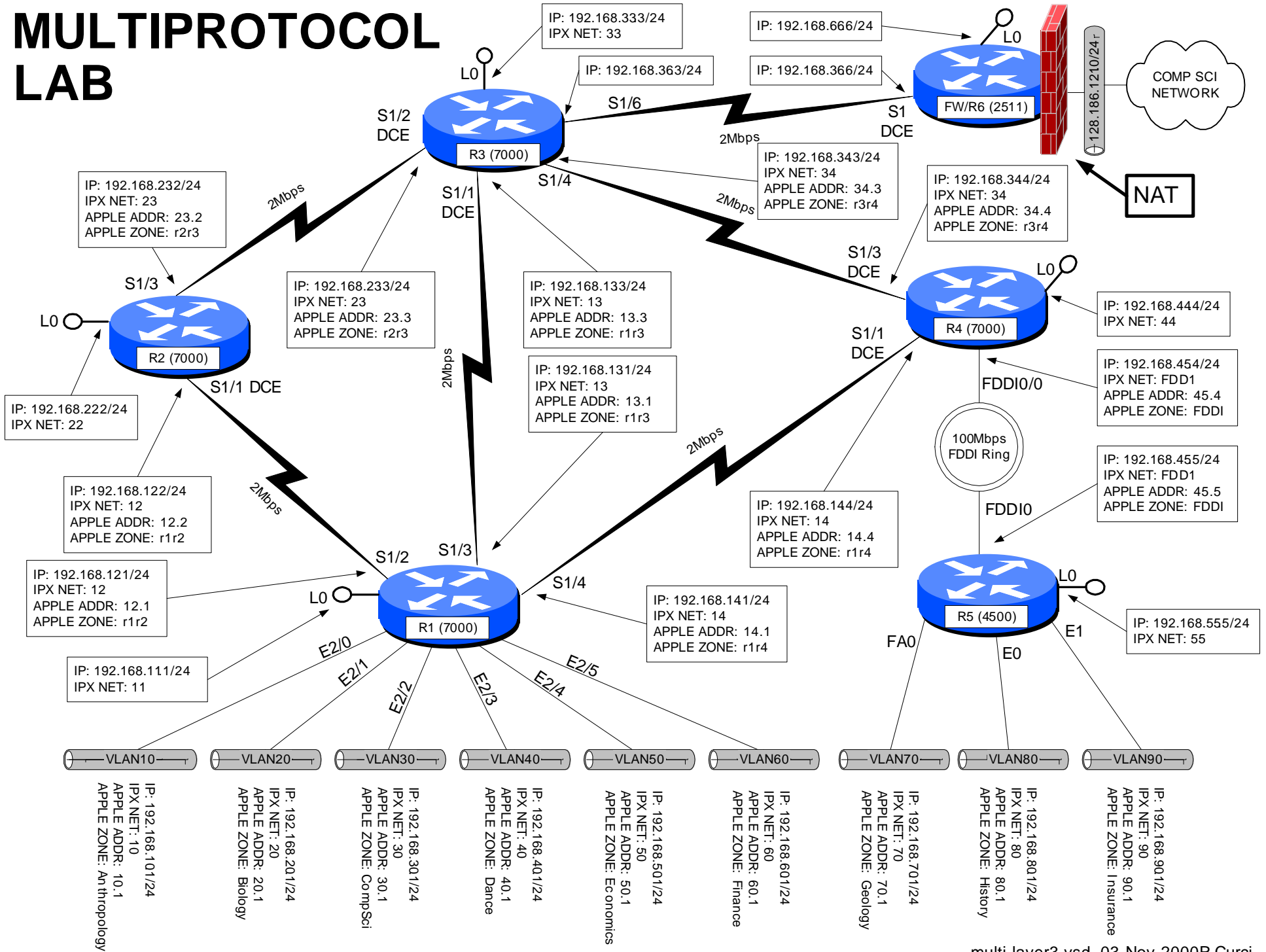
Enable appletalk routing with global command “appletalk routing” which uses the RTMP routing protocol by default. To enable appletalk on an interface and assign a network address, enter the interface command “appletalk cable-range N-N N.H” where N is your network number and H is your host number. The cisco router also has a built-in appletalk PING command that can be used for testing as follows.

```
r5#ping
Protocol [ip]: apple
Target AppleTalk address: 10.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Verbose [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte AppleTalk Echos to 10.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/7/12 ms
```

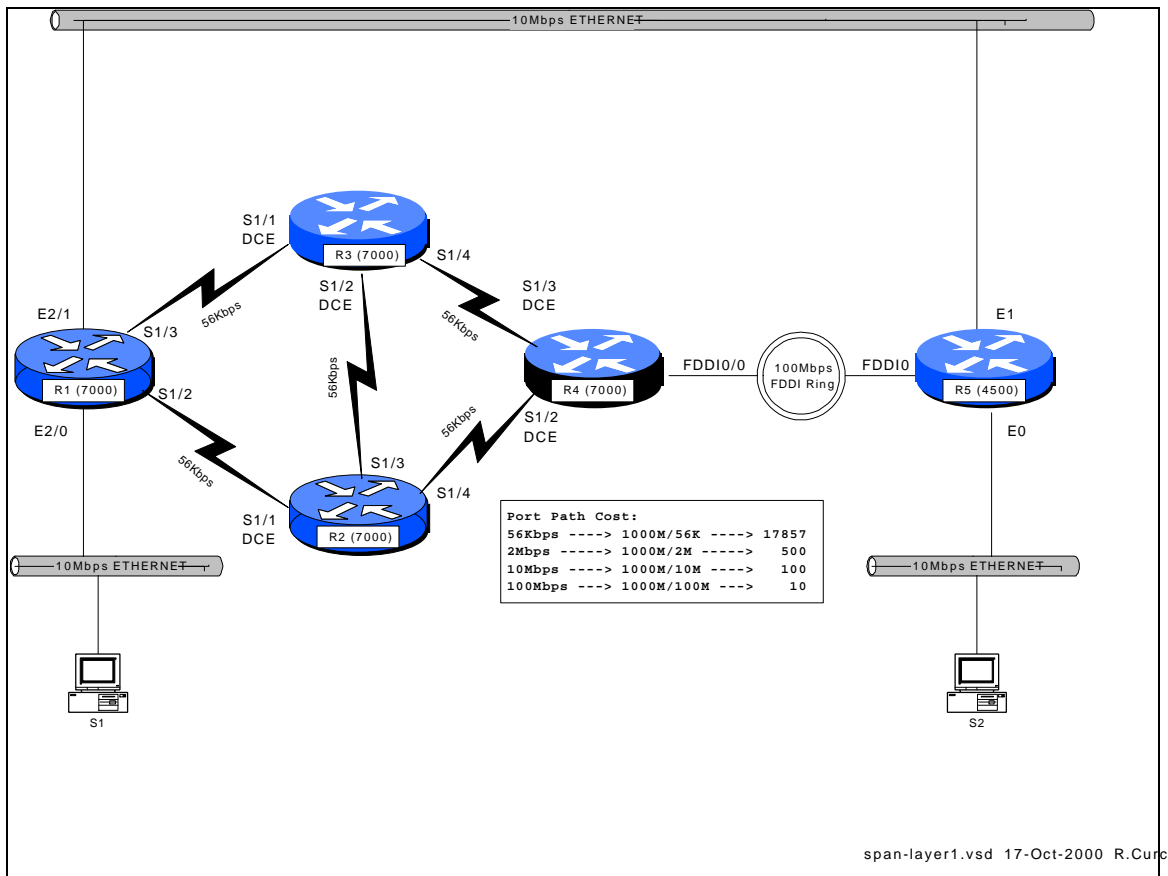
The following commands may be helpful if debugging your configuration.

- show appletalk route
- show appletalk zone
- show appletalk interface
- show appletalk interface brief
- show appletalk adjacent-routers
- show appletalk globals
- show appletalk neighbors
- show appletalk traffic

MULTIPROTOCOL LAB



INTERNET TEACHING LAB: SPANNING TREE PROTOCOL



Overview

The Spanning Tree Protocol, also known as the Djijkstra's Algorithm, is documented in the IEEE 802.1D standard. It is implemented in many current routers, bridges, and switches to provide a loop-free network topology. It is popular to build layer2 networks with redundant network connections to improve reliability, but the redundancy can lead to broadcast storms. Spanning Tree Protocol provides a mechanism for network devices to learn the network topology, elect a root bridge, and selectively block ports to form a loop-free spanning tree. We will explore some of the capabilities of this protocol, advantages, and limitations. The IEEE spanning tree protocol was first implemented in the DEC LAN Bridge 100 in the mid 1980s by Dr. Radia Perlman whose text book, *Interconnections*, now in the second edition, is the definitive reference.

Configuration

We will explore the Cisco Router implementation of 802.1D. Set up the physical cabling as specified in diagram above. The initial configuration for all five routers is listed at the end of this document also on text file [span-config.txt](#). Log into each of the five routers R1, R2, R3, R4, and R5, go into router configuration mode, and paste the

appropriate configuration commands. Verify that all appropriate interfaces are up and that everything is cabled to the correct routers and ports. Use the commands “show ip interface”, “show ip interface brief”, and “show cdp neighbors” for verification.

Setup PCs

Configure PCs S1 and S2 with IP addresses in the same IP network. Verify that you can PING between the two PCs. (Hint: If this does not work you can test the PCs by temporarily connecting them to the same physical Ethernet segment or by using a 10baseT Ethernet crossover cable. You may have difficulty if your router interface accidentally has an IP address on one of the bridge interface in which case it may be routing IP protocol and bridging non -IP traffic. You can verify that the router is bridging IP traffic on the appropriate interfaces with the command “show interface crb”)

Try sending a series of PINGs from S1 → S2 using both small 64-byte packets and large 1500-byte packets and note the average round-trip time. Repeat this test while S1 and S2 are temporarily directly connected. Compare the numbers and if substantially different, explain why.

There are redundant connections in your network and we want to determine the physical path between S1 and S2 used by the PING packets. First, determine the Ethernet MAC addresses for the NIC cards in S1 and S2. (Hint: If two devices on the same IP network have recently communicated, you will find each other’s Ethernet MAC address inside their respective ARP caches which can be displayed with the command “arp-a”)
Use the command “show bridge 1” on each router to display the bridge forwarding table and find the S1 and S2 entries. Record the forwarding path on your network diagram.

Bridge IDs and Port Path Cost

Using the command “show span 1”, determine which router is the root bridge and indicate it on your network diagram. This implementation of 802.1D computes the port path cost by dividing 1,000,000,000 by the bandwidth of the port in bits/second. This gives us the following port costs for the connections in your network:

INTERFACE TYPE	BANDWIDTH	PORT PATH COST
56K SERIAL	56,000 bits/sec	17857
10M ETHERNET	10,000,000 bits/sec	100
FDDI	100,000,000 bits/sec	10

Given your diagram, knowledge of the root bridge, and above table, manually compute the spanning tree algorithm. For each bridge port, indicate the port state (F=forwarding,

B=blocking) as well as the port type (RP=root port, DP=designated port, NDP=non-designated port).

Verify your calculations by comparing them with the output of the command “show spanning-tree 1” on each router.

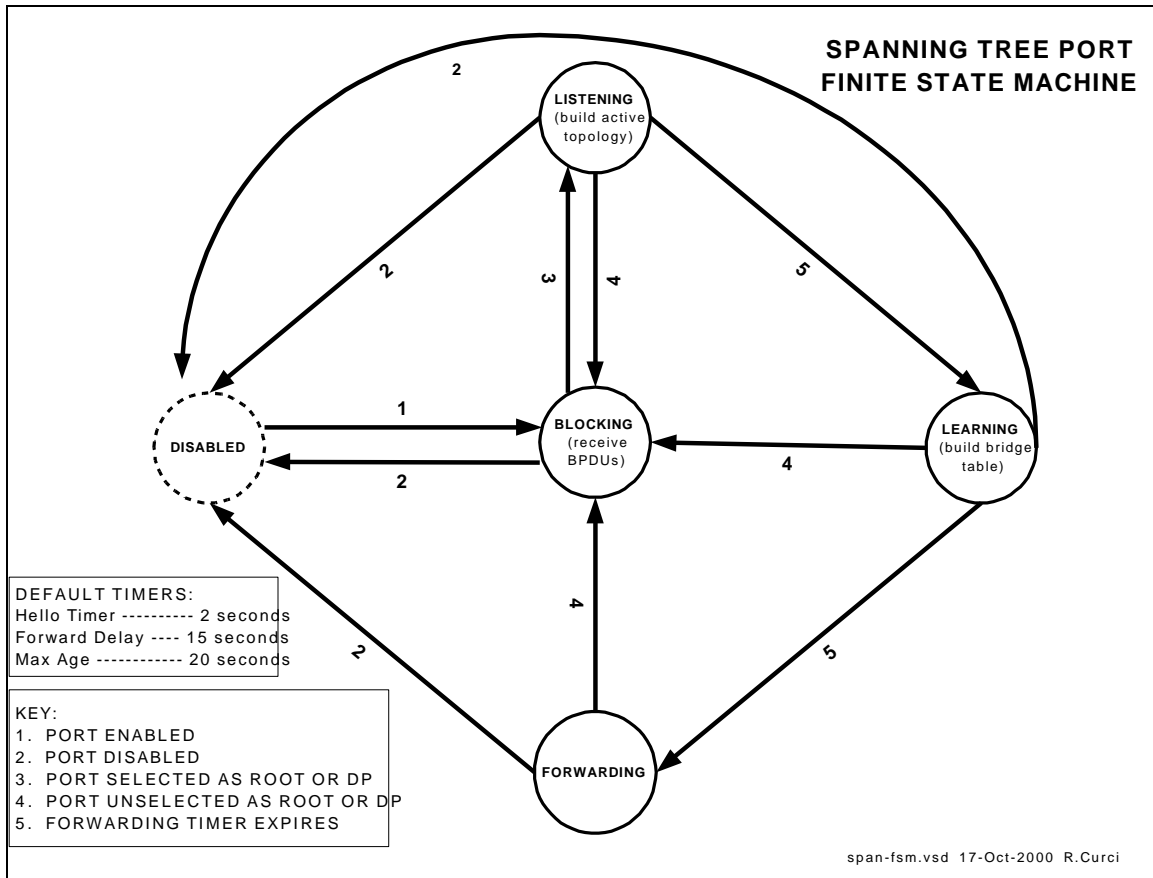
Bridge Protocol Data Units

On one of your routers with a blocked bridge port, issue the command “show interface xxx” where xxx is the name of the blocked interface/port. Note the input and output packet counters. Are they incrementing? If so, why are they incrementing? Instead of doing the arithmetic, you may find it easier to “clear counters” to zero the counters before you start.

The Cisco router has a number of debug modes used to diagnose network problems. Although sometimes dangerous to use on a production network, they are very good tools in a lab environment. The command “term monitor” will enable debug messages to be displayed on your router session and disabled with “term no monitor”. Try turning on the spanning tree topology change debug with “debug spanning tree” until you collect a few messages, then turn it off with “undebug all”. You should see some bridge protocol data unit packets represented in hexadecimal. You should be able to spot the MAC address of your root bridge embedded in the packet. Using the following table, decode the root bridge ID (priority and MAC address), sending bridge ID (priority and MAC address), root path cost, and timers.

FIELD	OCTETS	FUNCTION
Protocol ID	2	future (always zero)
Version	1	future (always zero)
Type	1	BPDU Type (0=config BPDU)
Flags	1	LSB (topolgy chg flash), MSB (Topology chg ACK)
Root BID	8	Bridge ID of root (16bit priority + 48bit MAC)
Root Path Cost	4	Cumulative cost to root bridge
Sending BID	8	Bridge ID of sender (16bit priority + 48bit MAC)
Port ID	2	Port ID that sent this BPDU
Message Age	2	Age of root BPDU
Max Age	2	Max age to save BPDU info (default = 20s)
Hello Time	2	Time between sending consecutive BPDUs (default = 2s)
Forward Delay	2	Time spent in listening and learning states (default = 15s)

Finite State Machine



Bridge ports can be in one of five states: disabled, blocking, listening, learning, and forwarding. See the diagram *span-fsm.pdf* to see what events cause transitions between different states. Log into one of your routers and identify a bridge interface in the forwarding state. Turn on spanning tree topology events debugging with “debug spanning events” and shut down the interface with “interface xyz” and “shutdown”. Wait a minute, then turn it back on with “no shutdown”. Note the state changes as it transitions from the disabled to the forwarding state including intermediate states. Record how much time was spent in each state. Turn off debugging with “undebug all”.

TEST TCP

Locate the program *TTCP* by searching the Internet. At the time of this writing, it was available for anonymous/ftp download at <ftp://FTP.ARL.MIL/pub/ttcp>. It is a TCP/IP benchmarking program. There are both C-language versions, usually named *ttcp.c*, and java implementations that work on Windows systems. You basically start this program on one system in receive mode, then start the other copy in transmit mode and supply the IP address of the receiver. The utility sends several blocks of data (you specify how many blocks and how many bytes per block) then displays statistics in Bytes/Second and Bits/Second on speed of the transfer. Use this tool to measure the network performance from S1 → S2 traversing your network. How many bits per second did you achieve? Study your network diagram paying particular attention to your router link speeds and

which interfaces are blocked. As packets traverse your network, your throughput is affected factors such as the speed of the links traversed, congestion, router CPU load and switching method, errors, etc. If you focus on the link speeds, is there a better (faster) path through your network that is not used? Determine which bridge should be made the root bridge in order to maximize the S1 → S2 throughput and change your configuration to make it so. Is there an optimal solution or more than one equally good solution? Repeat your S1 → S2 test and compare results with the first time. (Hint: The bridge with lowest bridge ID is elected the root. BIDs are 64-bit numbers by concatenating the bridge priority with the bridge MAC address. Although you normally cannot change the MAC address, you can change the bridge priority.) What is the slowest link traversed in the new network configuration? Was your throughput significantly less than your slowest link speed? Why? (Hint: read up on CSMA/CD)

INITIAL ROUTER CONFIGURATION:

COMMON:

```
service timestamps debug uptime
enable password cisco
no ip domain-lookup
ip classless
line con 0
  exec-timeout 0 0
line vty 0 4
  password cisco
  login
```

R1:

```
hostname r1
interface Serial1/2
  description Link to R2 S1/1
  no ip address
  bandwidth 56
  bridge-group 1
  no shutdown
interface Serial1/3
  description Link to R3 S1/1
  no ip address
  bandwidth 56
  bridge-group 1
  no shutdown
interface Ethernet2/0
  description Link to S1
  ip address 192.168.10.1
  255.255.255.0
  bridge-group 1
  no shutdown
interface Ethernet2/1
  description Link to R5 E1
  no ip address
  bridge-group 1
  no shutdown
bridge crb
bridge 1 protocol ieee
bridge 1 route ip
```

R2:

```
hostname r2
interface Serial1/1
  description Link to R1 S1/2
  no ip address
  bandwidth 56
  clockrate 56000
  bridge-group 1
  no shutdown
interface Serial1/3
  description Link to R3 S1/2
  no ip address
  bandwidth 56
  clockrate 56000
  bridge-group 1
  no shutdown
interface Serial1/4
  description Link to R4 S1/2
  no ip address
  bandwidth 56
```

```
bridge-group 1
  no shutdown
bridge crb
bridge 1 protocol ieee
bridge 1 priority 100
```

R3:

```
hostname r3
interface Serial1/1
  description Link to R1 S1/3
  no ip address
  bandwidth 56
  clockrate 56000
  bridge-group 1
  no shutdown
interface Serial1/2
  description Link to R2 S1/3
  no ip address
  bandwidth 56
  clockrate 56000
  bridge-group 1
  no shutdown
interface Serial1/4
  description Link to R4 S1/3
  no ip address
  bandwidth 56
  bridge-group 1
  no shutdown
bridge crb
bridge 1 protocol ieee
```

R4:

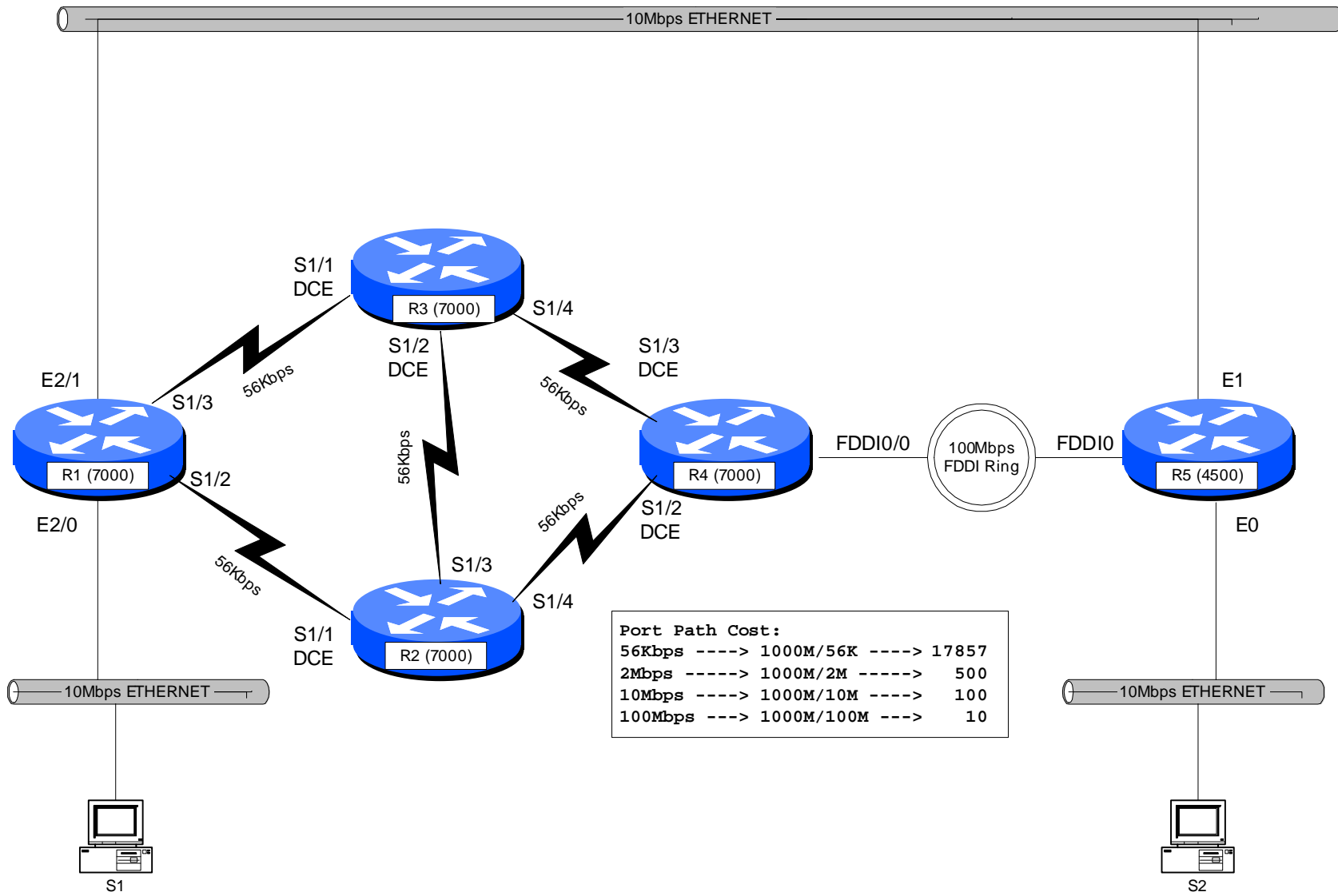
```
hostname r4
interface Fddi0/0
  description Link to R5 FDDI0
  no ip address
  bridge-group 1
  no shutdown
interface Serial1/2
  description LINK to R2 S1/0
  no ip address
  bandwidth 56
  clockrate 56000
  bridge-group 1
  no shutdown
interface Serial1/3
  description LINK to R3 S1/0
  no ip address
  bandwidth 56
  clockrate 56000
  bridge-group 1
  no shutdown
bridge crb
bridge 1 protocol ieee
bridge 1 route ip
```

R5:

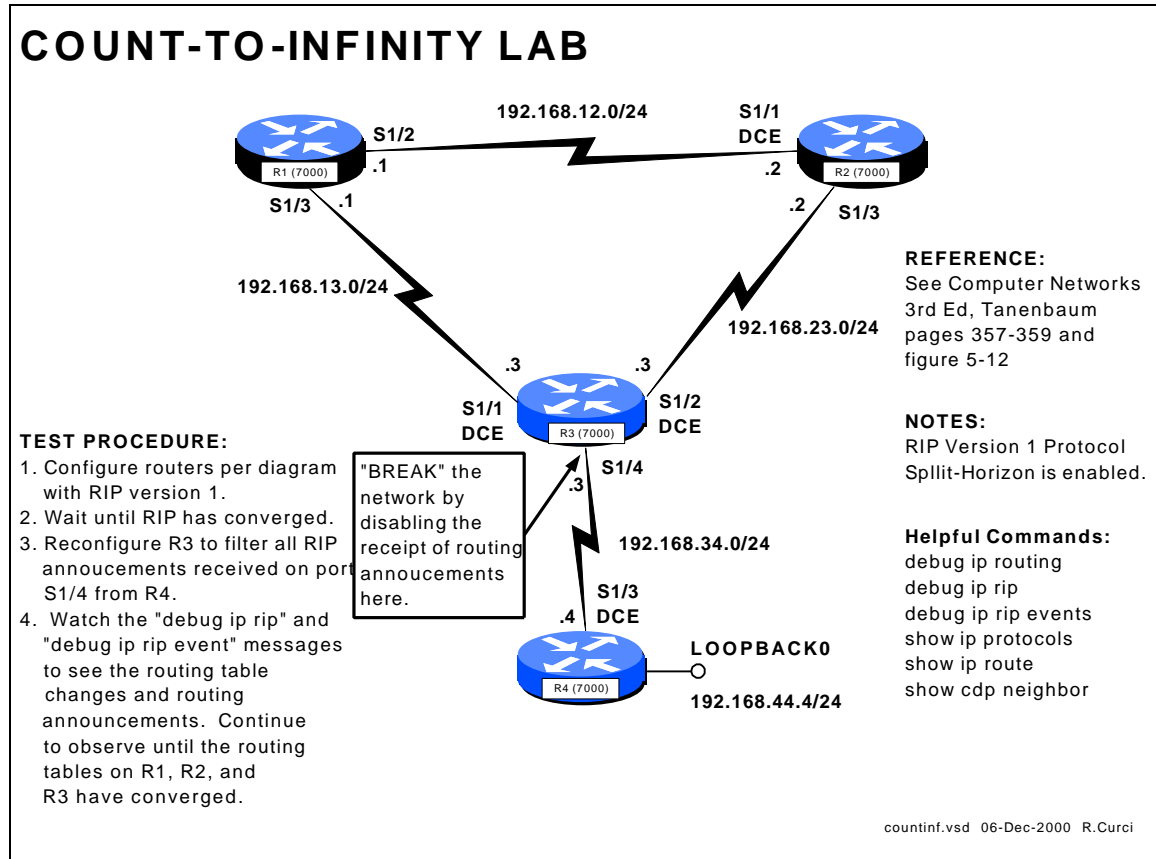
```
hostname r5
interface Ethernet0
  description Link to S2
```

```
no ip address
bridge-group 1
no shutdown
interface Ethernet1
description Link to R1 E2/0
no ip address
media-type 10BaseT
bridge-group 1
```

```
no shutdown
interface Fddi0
no ip address
bridge-group 1
no shutdown
bridge crb
bridge 1 protocol ieee
bridge 1 route ip
```



INTERNET TEACHING LAB: COUNT TO INFINITY LAB



OVERVIEW

In this lab, we will explore the “count to infinity” problem of distance vector routing protocols such as RIP version 1. (For background information, read Tanenbaum’s Computer Networks 3rd Edition pages 357 through 359.) Normally, routers with distance vector routing protocols implement the *split horizon* algorithm where they will not advertise a network route out an interface to a neighbor from whom the route was learned. This can help reduce the *convergence time*, the time it takes the routing tables in each router to reach a steady state. We will configure the lab network on routers R1, R2, R3, and R4 as shown on the diagram above. By configuring routers R1, R2, and R3 in a cycle, we will attempt to defeat the *split horizon* hack and will try to demonstrate the count to infinity problem, the problem where distance vector routing protocols can take a very long time to reach convergence.

Routers R1, R2, and R3 are connected with serial links in the shape of a triangle. R3 also has a serial link to R4. R4 has a loopback interface to network 192.168.44.0/24 which we will simply call “network 44”. This lab network contains five IP networks that will be abbreviated as shown in the following table.

IP NETWORK	ABBREVIATION
192.168.12.0/24	12
192.168.13.0/24	13
192.168.23.0/24	23
192.168.34.0/24	34
192.168.44.0/24	44

[IP Networks in this Lab]

We will perform the following experiment:

1. Configure the network as in the above diagram with RIP version 1 protocol and wait for RIP to converge to a steady state.
2. Examine the routing tables and verify that each router has a route for networks 12, 13, 23, 34, and 44. We are especially interested in network 44 on the loopback interface of R4.
3. “Break” the connection between R3 and R4 by installing an access list on R3’s Serial1/4 interface that blocks RIP traffic received R4.
4. Examine the routing announcements on R1, R2, and R3 and watch how their routing tables change the R3---R4 connection is “broken.” Pay particular attention to network 44 which is no longer reachable but this will not be immediately known to router R3. We expect the routing metric on routers R1, R2, and R3 for network 44 to gradually increase, by one hop at a time, until a hop count of 16 or RIP infinity is reached.

BACKGROUND

The RIP protocol uses four adjustable timers to control its operation. There is a single UPDATE timer and an instance of the INVALID, HOLDDOWN, and FLUSH timers for each entry in the routing table.

- **UPDATE**
This timer controls how frequently a router announces routes to its neighbors. By default, this occurs every 30 seconds.
- **INVALID**
This controls how long after not hearing an update for a route that the route will be declared invalid. By default, this timer is set to 180 seconds or 3 minutes which represents 6 RIP update cycles. It is restarted whenever a route is received.
- **HOLDDOWN**
This controls how long after a route has been invalidated a router will wait before accepting a new route of a higher metric. This helps reduce the count-to-infinity problem. By default, this timer is set to 180 seconds or 3 minutes.

- **FLUSH**
This timer controls when a routing table entry is removed. It restarts every time a route is received and runs concurrently with the INVALID and HOLDDOWN timers. When the FLUSH timer has expired for a route, the route is removed from the routing table. The FLUSH timer expires before the HOLDDOWN timer, so HOLDDOWN never runs for its complete cycle.

The “show ip protocols” router command displays the current values for the RIP timers, as well as a list of routers from whom RIP announcements have been received:

```
r3#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 6 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface        Send  Recv   Key-chain
    Serial1/1         1     1 2
    Serial1/2         1     1 2
    Serial1/4         1     1 2
  Routing for Networks:
    192.168.13.0
    192.168.23.0
    192.168.34.0
  Routing Information Sources:
    Gateway          Distance    Last Update
    192.168.34.4      120         00:00:03
    192.168.13.1      120         00:00:16
    192.168.23.2      120         00:00:04
  Distance: (default is 120)
r3#
```

STEP1 – Configure the Network:

For this exercise, we will only need to use routers R1, R2, R3, and R4. Configure these routers by erasing their configurations and pasting the following configuration information into the routers. Note that the “COMMON” section should be applied to all 4 routers, and the other sections as appropriate. For more information on router configuration basics, see the “Basic Router Configuration” lab.

INITIAL ROUTER CONFIGURATION:

COMMON:

```
service timestamp debug uptime
enable password cisco
no ip domain-lookup
ip classless
line con 0
  exec-timeout 0 0
line vty 0 4
  password cisco
  login
```

R1:

```
hostname r1
interface Serial1/2
  description Link to R2 S1/1
  ip address 192.168.12.1
255.255.255.0
  no shutdown
interface Serial1/3
  description Link to R3 S1/1
  ip address 192.168.13.1
255.255.255.0
  no shutdown
router rip
  network 192.168.12.0
  network 192.168.13.0
```

R2:

```
hostname r2
interface Serial1/1
  description Link to R1 S1/2
  ip address 192.168.12.2
255.255.255.0
  clockrate 2000000
  no shutdown
interface Serial1/3
  description Link to R3 S1/2
  ip address 192.168.23.2
255.255.255.0
  no shutdown
router rip
  network 192.168.12.0
```

```
network 192.168.23.0
```

R3:

```
hostname r3
interface Serial1/1
  description Link to R1 S1/3
  ip address 192.168.13.3
255.255.255.0
  clockrate 2000000
  no shutdown
interface Serial1/2
  description Link to R2 S1/3
  ip address 192.168.23.3
255.255.255.0
  clockrate 2000000
  no shutdown
interface Serial1/4
  description Link to R4 S1/3
  ip address 192.168.34.3
255.255.255.0
  no shutdown
router rip
  network 192.168.13.0
  network 192.168.23.0
  network 192.168.34.0
```

R4:

```
hostname r4
interface Loopback0
  ip address 192.168.44.4
255.255.255.0
  no shutdown
interface Serial1/3
  description Link to R3 S1/4
  ip address 192.168.34.4
255.255.255.0
  clockrate 2000000
  no shutdown
router rip
  network 192.168.44.0
  network 192.168.34.0
```

STEP2 – Examine Routing Tables:

Output from the “show ip route” command on each of the four routers is shown below. Note that routes for the same 5 networks appear on each router. For each router, networks that are directly connected prefixed with “C” for Connected while those learned through RIP are prefixed with “R”. Note that for the RIP entries in the square brackets are the administrative distance (120 for RIP) and the RIP hop count metric which are boldfaced. You will also notice sometimes where there are more than one entry for the same network. For example, notice that router R1 has two entries for network 23 both with metric 1. This is because there are two equal cost paths from R1 to network 23, one via interface Serial1/2 and the other via interface Serial1/3.

```

r1#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
R   192.168.44.0/24 [120/2] via 192.168.13.3, 00:00:20, Serial1/3
R   192.168.34.0/24 [120/1] via 192.168.13.3, 00:00:20, Serial1/3
C   192.168.12.0/24 is directly connected, Serial1/2
C   192.168.13.0/24 is directly connected, Serial1/3
R   192.168.23.0/24 [120/1] via 192.168.13.3, 00:00:20, Serial1/3
      [120/1] via 192.168.12.2, 00:00:07, Serial1/2

r2#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
R   192.168.44.0/24 [120/2] via 192.168.23.3, 00:00:06, Serial1/3
R   192.168.34.0/24 [120/1] via 192.168.23.3, 00:00:06, Serial1/3
C   192.168.12.0/24 is directly connected, Serial1/1
R   192.168.13.0/24 [120/1] via 192.168.12.1, 00:00:19, Serial1/1
      [120/1] via 192.168.23.3, 00:00:07, Serial1/3
C   192.168.23.0/24 is directly connected, Serial1/3

r3#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
R   192.168.44.0/24 [120/1] via 192.168.34.4, 00:00:04, Serial1/4
C   192.168.34.0/24 is directly connected, Serial1/4
R   192.168.12.0/24 [120/1] via 192.168.13.1, 00:00:27, Serial1/1
      [120/1] via 192.168.23.2, 00:00:27, Serial1/2
C   192.168.13.0/24 is directly connected, Serial1/1
C   192.168.23.0/24 is directly connected, Serial1/2

r4#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
C   192.168.44.0/24 is directly connected, Loopback0
C   192.168.34.0/24 is directly connected, Serial1/3
R   192.168.12.0/24 [120/2] via 192.168.34.3, 00:00:20, Serial1/3
R   192.168.13.0/24 [120/1] via 192.168.34.3, 00:00:21, Serial1/3
R   192.168.23.0/24 [120/1] via 192.168.34.3, 00:00:21, Serial1/3

```

STEP3 – “BREAK” the R3—R4 Connection:

We will now break the connection between R3 and R4 such that R3 will no longer hear advertisements for network 44. Instead of unplugging the cable where R3 would immediately notice the that connection went down, we will be sneaky and instead install an access list on R3’s interface Serial1/4 input to prevent it from hearing any RIP advertisements. From router R3’s RIP process perspective, it will not have any indication of any problems except that it will no longer hear advertisements for network 44.

```

! First turn on debugging so we can see what is happening:
r3# debug ip rip
r3# debug ip rip events
r3# term monitor
! Now create an access list and apply to deny traffic from R4:
r3# config term
r3(config)# no access-list 1
r3(config)# access-list 1 deny any
r3(config)# interface Serial1/4
r3(config-if)# ip access-group 1 in

```

STEP4 – Examine Routing Table and Announcement Changes:

Router R3 was reconfigured to filter out all RIP updates from R4 at 23:11:00. Here are the messages from “debug ip rip” and “debug ip rip events” on R3:

```
(R3 continues to advertise network 44 with metric 2 for about 3 minutes)
23:11:00:    network 192.168.44.0, metric 2
23:11:27:    network 192.168.44.0, metric 2
23:11:56:    network 192.168.44.0, metric 2
23:12:25:    network 192.168.44.0, metric 2
23:12:51:    network 192.168.44.0, metric 2
23:13:18:    network 192.168.44.0, metric 2
23:13:46:    network 192.168.44.0, metric 2
23:14:16:    network 192.168.44.0, metric 16 (advertising unreachable)
23:14:16: RT: flushed route to 192.168.44.0 via 192.168.34.4 (Serial1/4)
23:14:16: RT: no routes to 192.168.44.0, entering holddown
23:15:13: RT: garbage collecting entry for 192.168.44.0
23:15:13: RIP: sending v1 update to 255.255.255.255 via Serial1/1
23:15:13:    (First update without any route to network 192.168.44.0)
23:15:13:    network 192.168.34.0, metric 1
23:15:13:    network 192.168.23.0, metric 1
23:15:13: RIP: Update contains 2 routes
23:15:13: RIP: Update queued
23:15:14: RIP: Update sent via Serial1/1
```

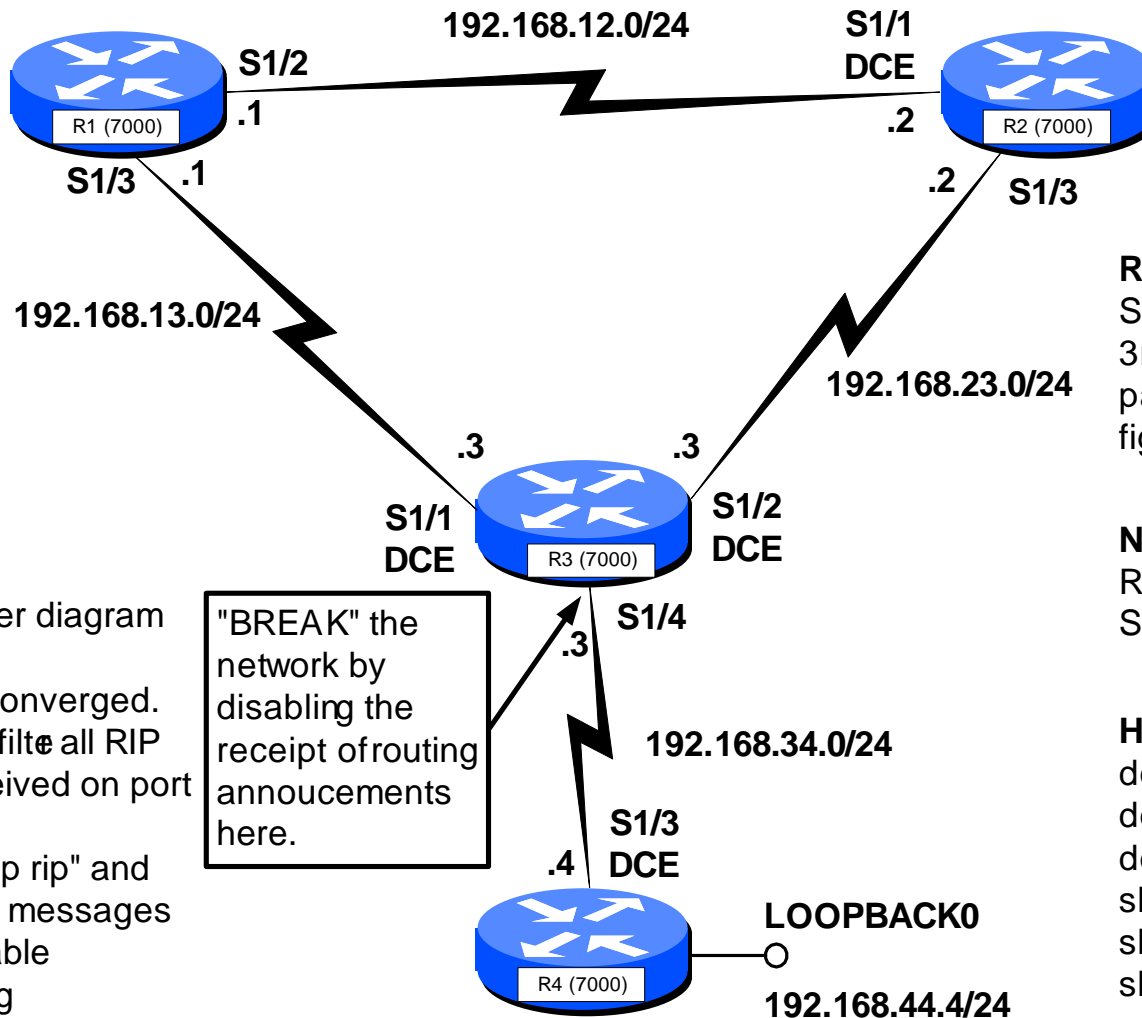
Here are the debug messages captured on router R1:

```
23:14:16: RIP: received v1 update from 192.168.13.3 on Serial1/3
23:14:16:    192.168.44.0 in 16 hops (inaccessible)
23:14:16: RT: delete route to 192.168.44.0 via
           192.168.13.3, rip metric [120/2]
23:14:16: RT: no routes to 192.168.44.0, entering holddown
23:17:22: RT: 192.168.44.0 came out of holddown
23:17:56: RT: garbage collecting entry for 192.168.44.0
```

After “breaking” the R3—R4 connection, R3 continues to advertise network 44 to its neighbors with metric 44 every 30 seconds. About 3 minutes after the “break”, the INVALID timer expires and R3’s entry for network 44 is marked as INVALID. It will still use this route, but will not advertise it as reachable to its neighbors. R3 network 44 advertisements now have metric 16 or unreachable. Since R3’s route for network 44 is now in HOLDDOWN, it will not accept any advertisements for this network with a metric greater than 2 preventing it from learning an incorrect route from R1 or R2. After approximately 4 minutes after the “break”, the FLUSH timer expires and the route indicates “gabbage collecting entry for 192.168.44.0” and the entry to network 44 is completely removed.

In this example, routers R1, R2, and R3 marked their routes to network 44 with metric 16 or unreachable after just over 3 minutes after the “break” and converged to a consistent state. This is much faster than we would have predicted from Tanenbaum. The CISCO use of the HOLDDOWN timer when a router will not accept routes with a higher metric and the use of a technique called “poison reverse” where a router advertises a network with metric 16 or unreachable help the routing tables converge more quickly than predicted.

COUNT-TO-INFINITY LAB



TEST PROCEDURE:

1. Configure routers per diagram with RIP version 1.
2. Wait until RIP has converged.
3. Reconfigure R3 to filter all RIP announcements received on port S1/4 from R4.
4. Watch the "debug ip rip" and "debug ip rip event" messages to see the routing table changes and routing announcements. Continue to observe until the routing tables on R1, R2, and R3 have converged.

"BREAK" the network by disabling the receipt of routing announcements here.

REFERENCE:

See Computer Networks 3rd Ed, Tanenbaum pages 357-359 and figure 5-12

NOTES:

RIP Version 1 Protocol Split-Horizon is enabled.

Helpful Commands:

```
debug ip routing
debug ip rip
debug ip rip events
show ip protocols
show ip route
show cdp neighbor
```