CIS 192 - Lesson 3

Instructor: **Rich Simms**
Dial-in: **888-450-4821**
Passcode: **761867**


Solomon


Sean C.

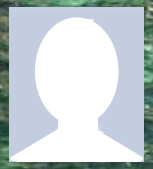

Chris


Corey


Bryan


Sean F.


Tony


David


Donna


Dave


Evan


Gabriel


Elia


Tajvia


Carlos


Adam


Ben


Laura

Email me (risimms@cabrillo.edu) a relatively current photo of your face for 3 points extra credit

Lesson Module Checklist

- Slides
- Whiteboard with 1st minute quiz

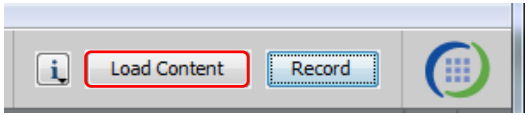
- Flashcards
- Web Calendar summary
- Web book pages
- Commands
- Howtos

- Lab tested
- Opus - lab template in depot
- Youtube Videos, if any, uploaded
- Whiteboard updated with random order quiz questions

- Backup slides, Confer links, handouts on flash drive
- 9V backup battery for microphone

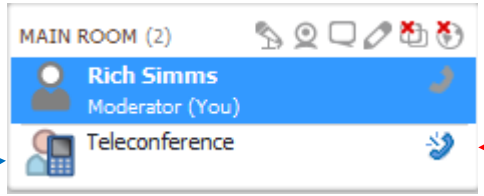
CCC Confer

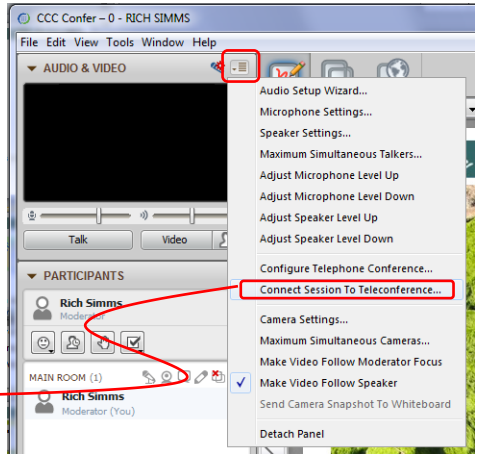
[ ] Preload White Board with *cis*lesson??*-WB*



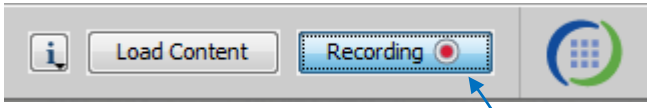
[ ] Connect session to Teleconference

Session now connected to teleconference





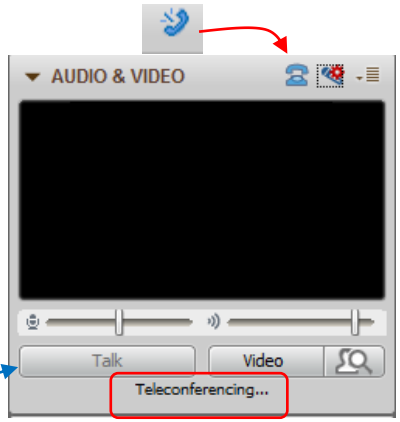
[ ] Is recording on?



Red dot means recording

[ ] Use teleconferencing, not mic

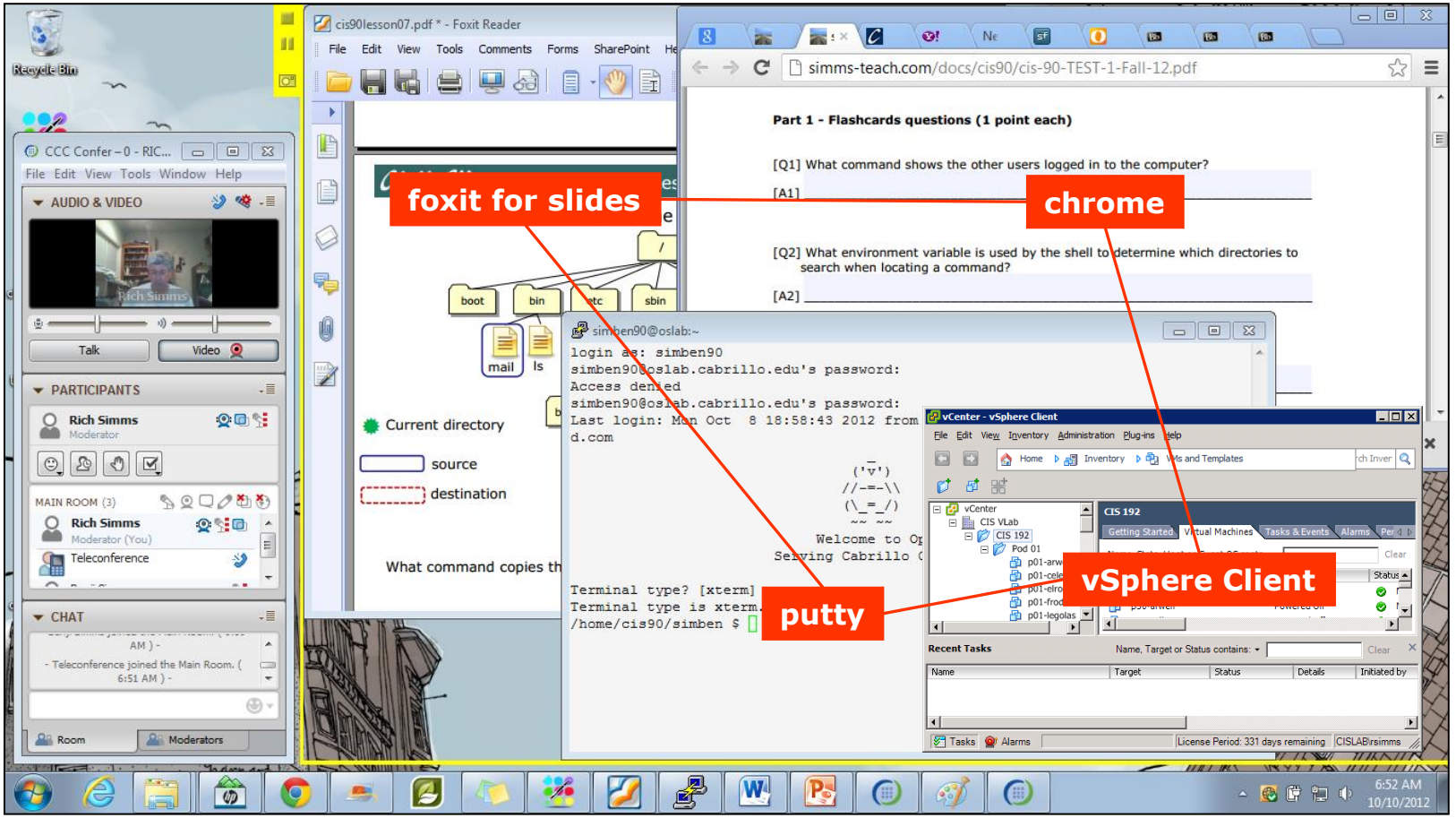
Should be greyed out



CCC Confer

Don't Forget

- [ ] **Video (webcam) optional**
- [ ] **layout and share apps**

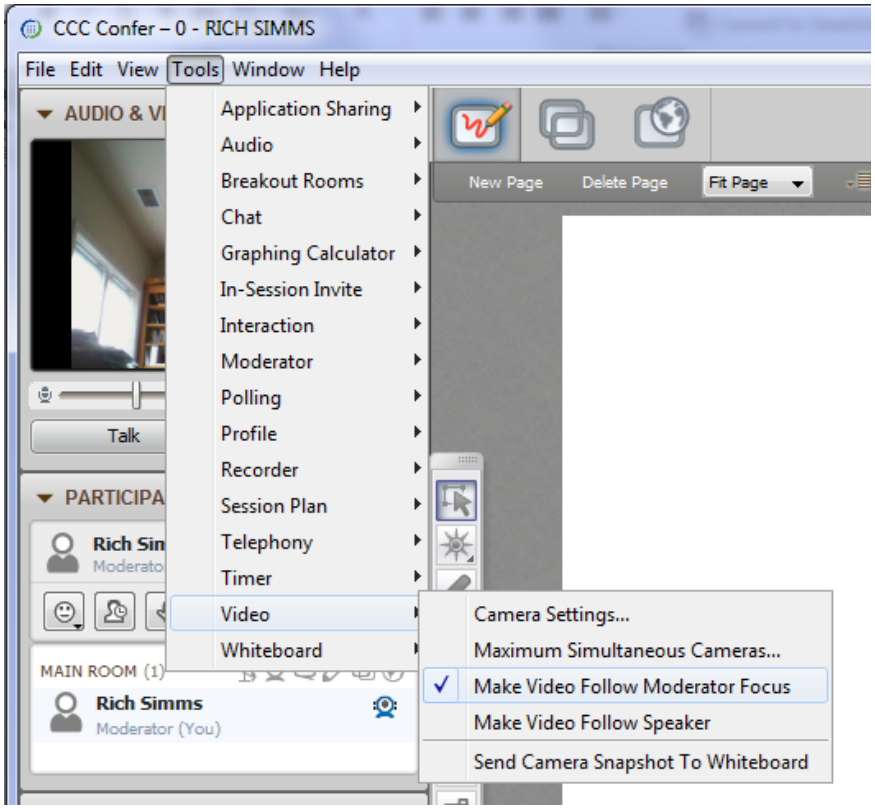


CCC Confer

[ ] Video (webcam) optional

[ ] Follow moderator

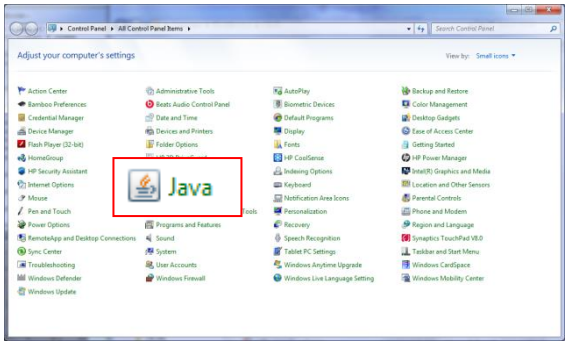
[ ] Double-click on postages stamps



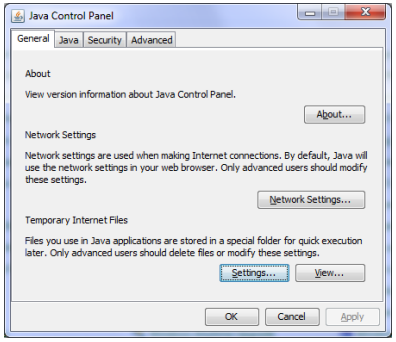
Universal Fix for CCC Confer:

1) Shrink (500 MB) and delete Java cache
2) Uninstall and reinstall latest Java runtime

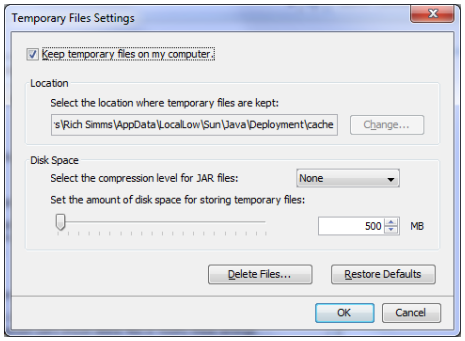
Control Panel (small icons)



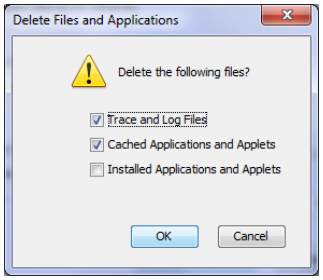
General Tab > Settings…



500MB cache size



Delete these



Google Java download

Course history and credits

Jim Griffin

- Jim created the original version of this course
- Jim's site: http://cabrillo.edu/~jgriffin/

Rick Graziani

- Thanks to Rick Graziani for the use of some of his great network slides
- Rick's site: http://cabrillo.edu/~rgraziani/

First Minute Quiz

Please answer these questions **in the order** shown:

Use CCC Confer White Board

For credit email answers to:
risimms@cabrillo.edu
within the first few minutes of class

Routing and Subnetting

Related Course Objectives	Agenda
• Configure appropriate IP addresses, network and subnet masks, and broadcast addresses based on the size and number of network segments required. • Connect multiple network segments together using Linux servers as routers and configuring the appropriate routing tables	• Quiz • Questions • Review and practice • Housekeeping • Routing • Packet forwarding • Static Routes • Routing table • Troubleshooting • Lab • Wrap

Questions

Questions

Lesson material?

Labs?

How this course works?

Chinese Proverb	他問一個問題，五分鐘是個傻子，他不問一個問題仍然是一個傻瓜永遠。 *He who asks a question is a fool for five minutes; he who does not ask a question remains a fool forever.*

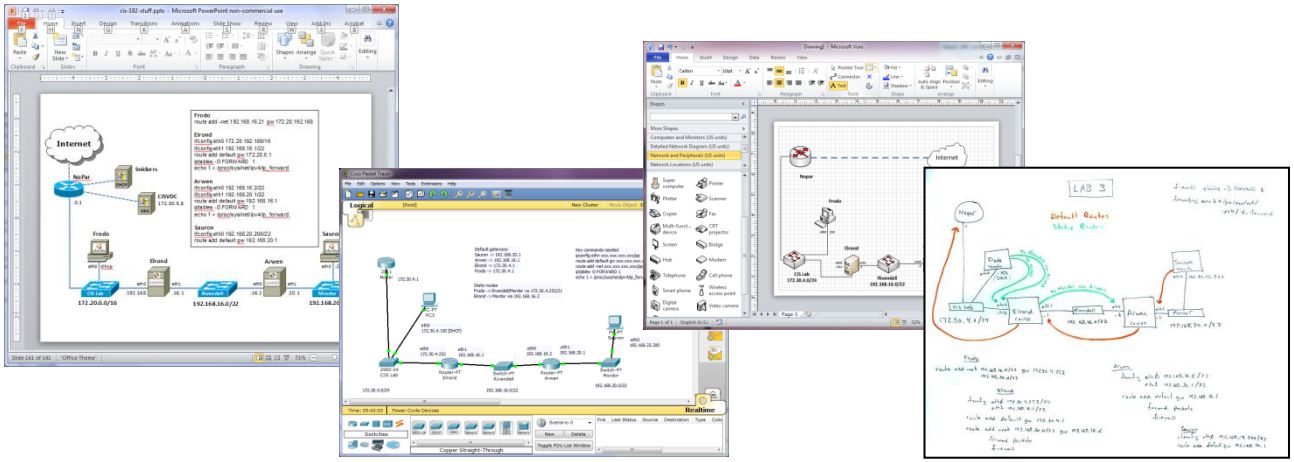
Network Topology Tools

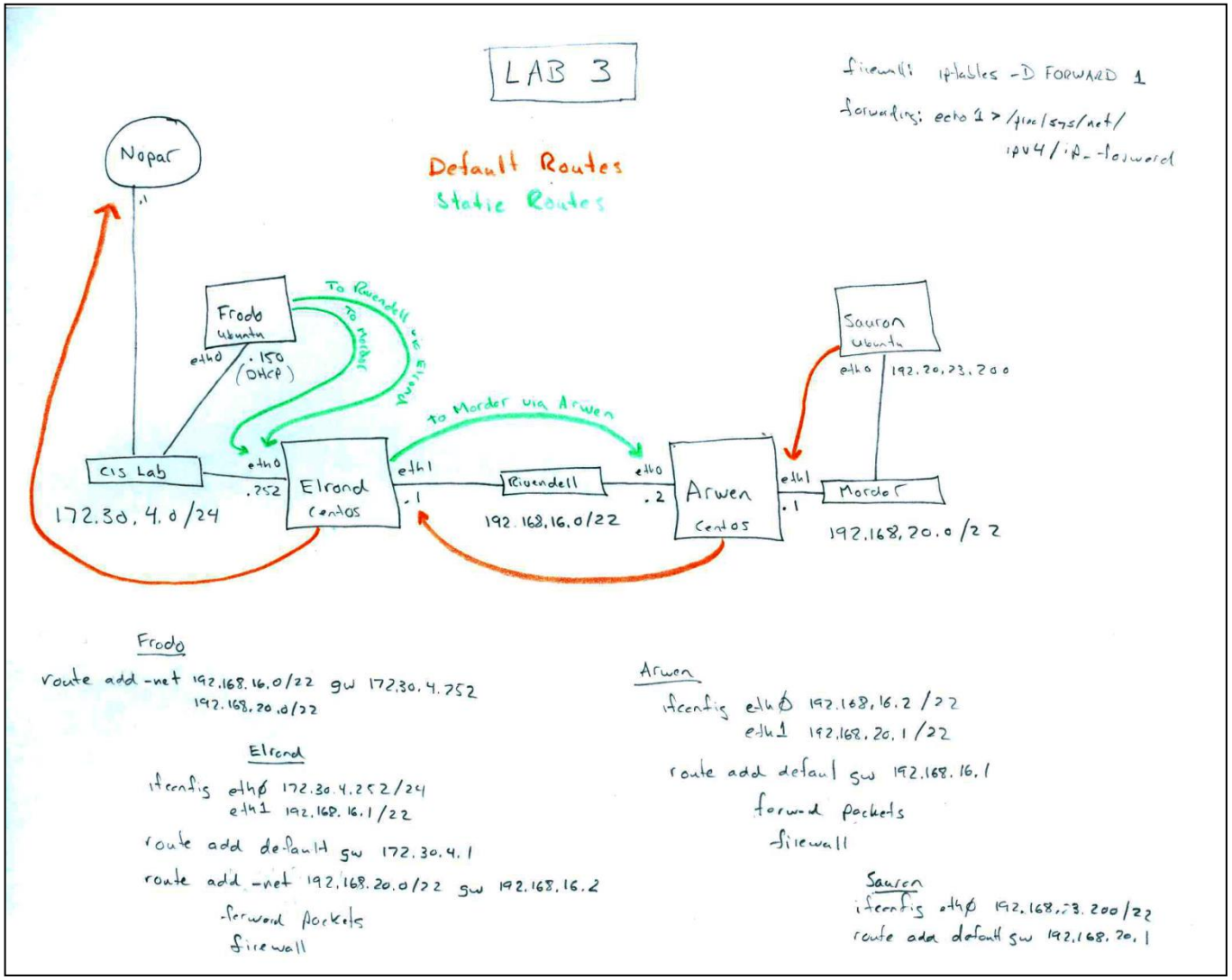
Network Topology Diagrams

Plan > Implement > Troubleshoot/Test

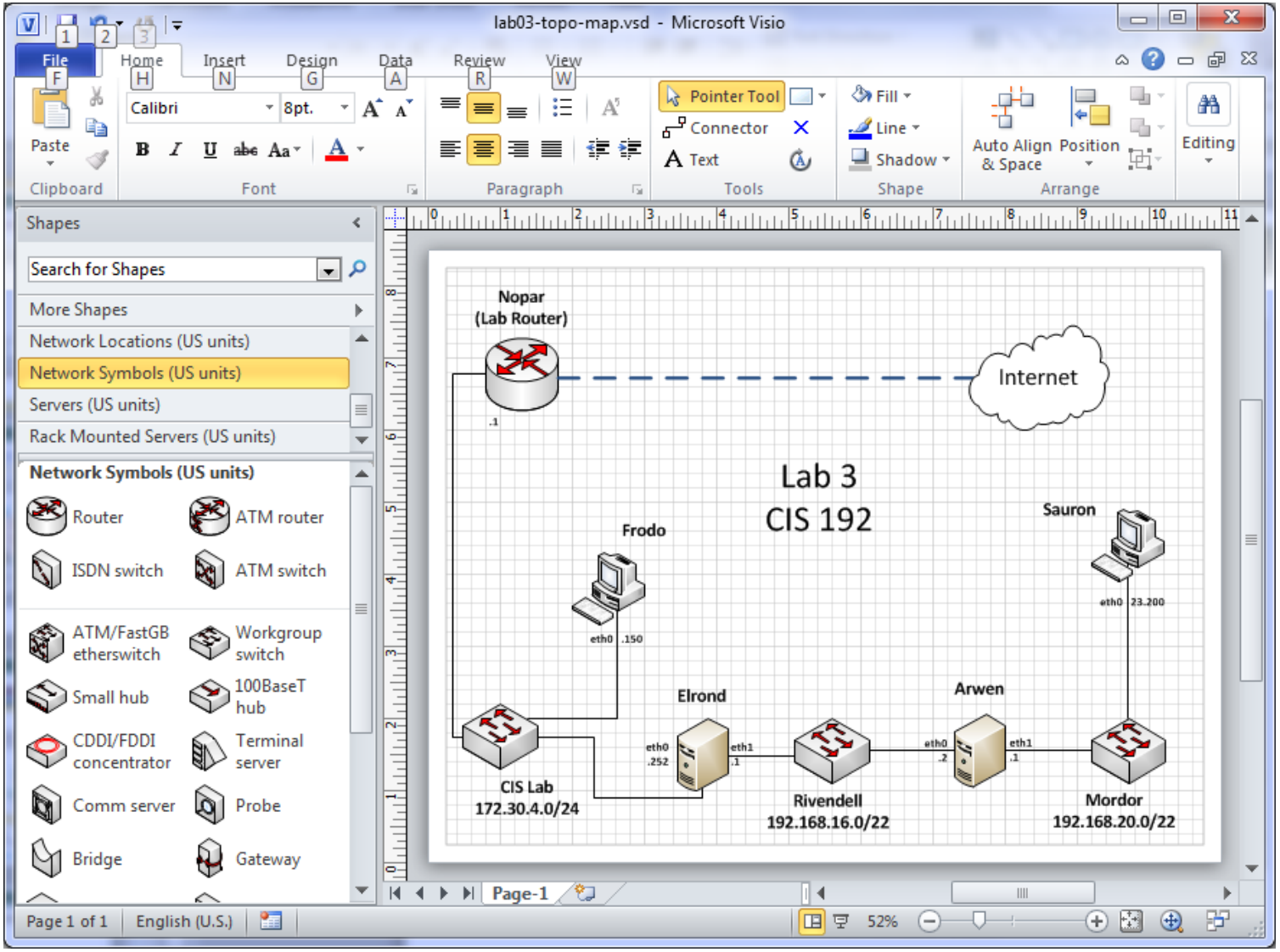
Spending a little time here

... will save you a "Ton" of time here when doing lab assignments

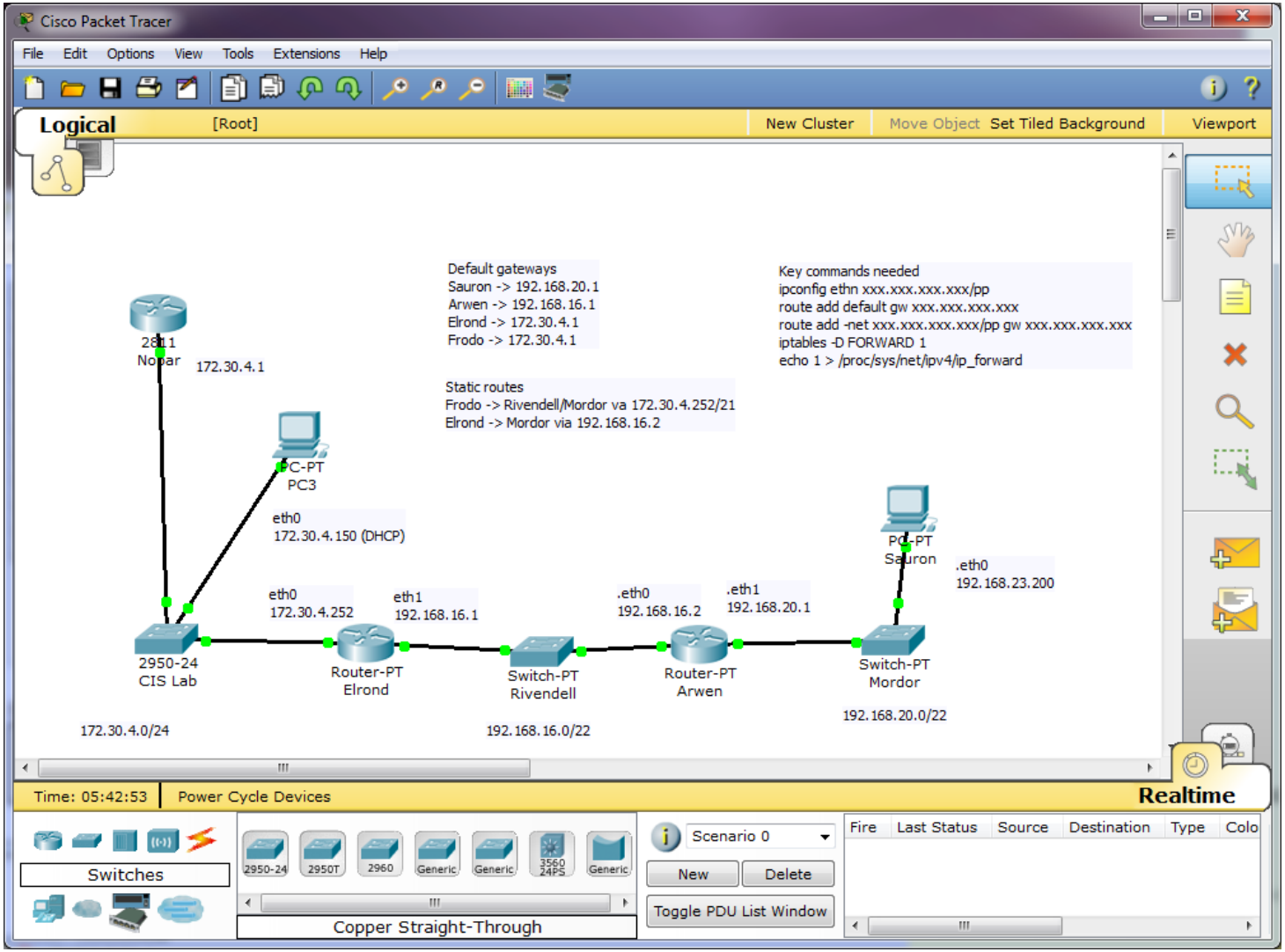




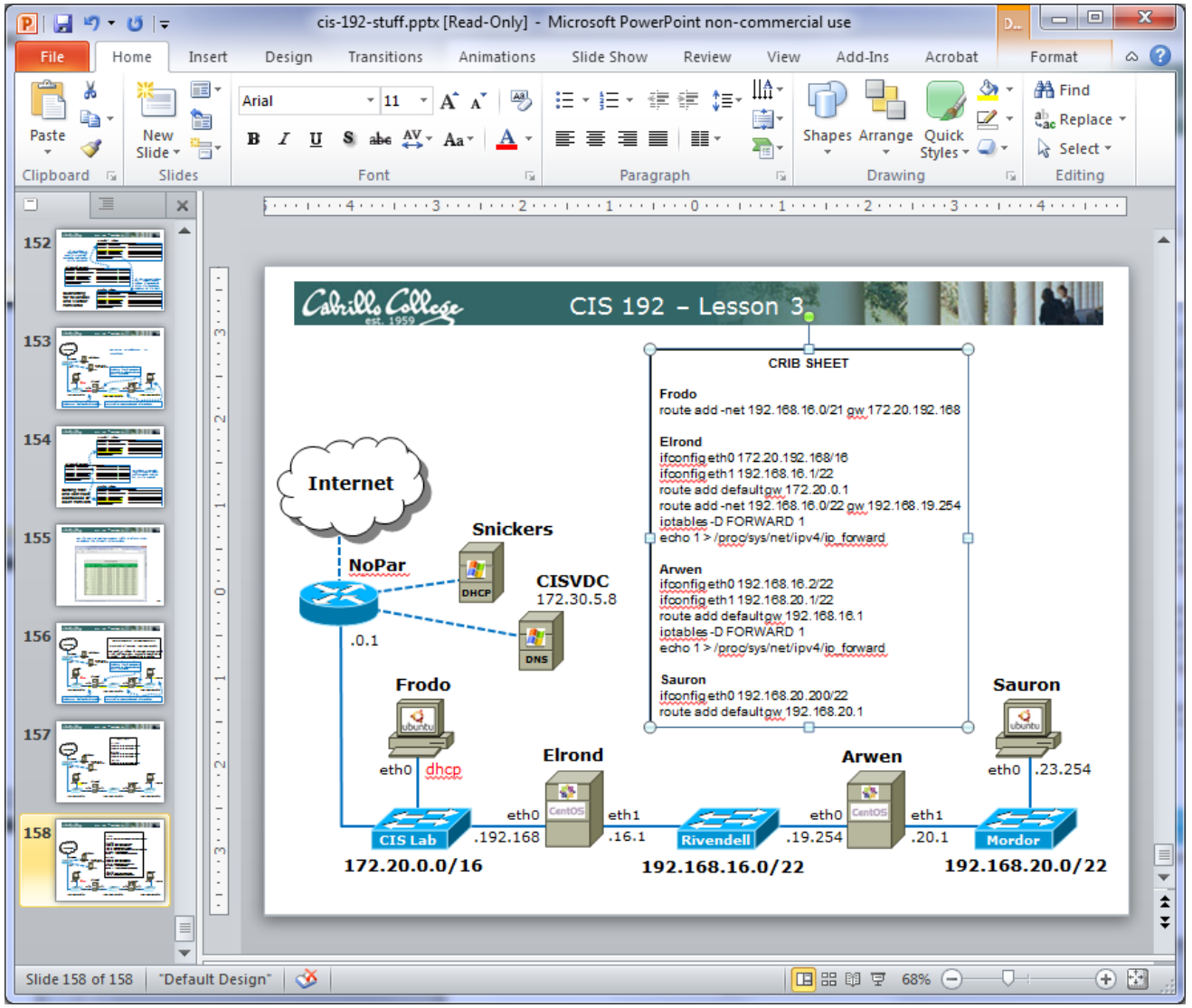
Some use pencil or whiteboard



Some use Visio

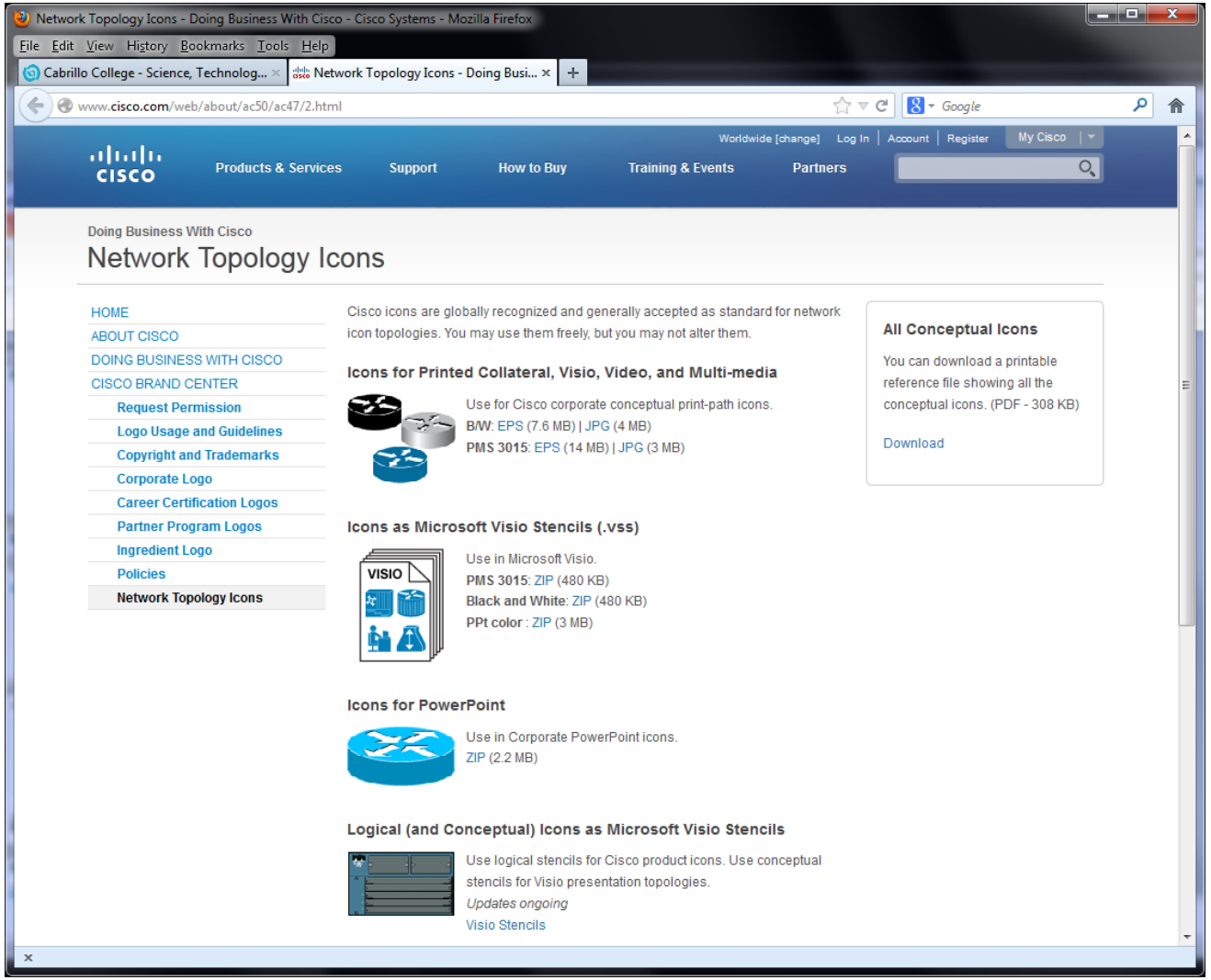


Some use Cisco Packet Tracer



Some use PowerPoint

http://www.cisco.com/web/about/ac50/ac47/2.html



Cisco provides network icons

Network Topology Drawing Tools

Many more drawing tools ...

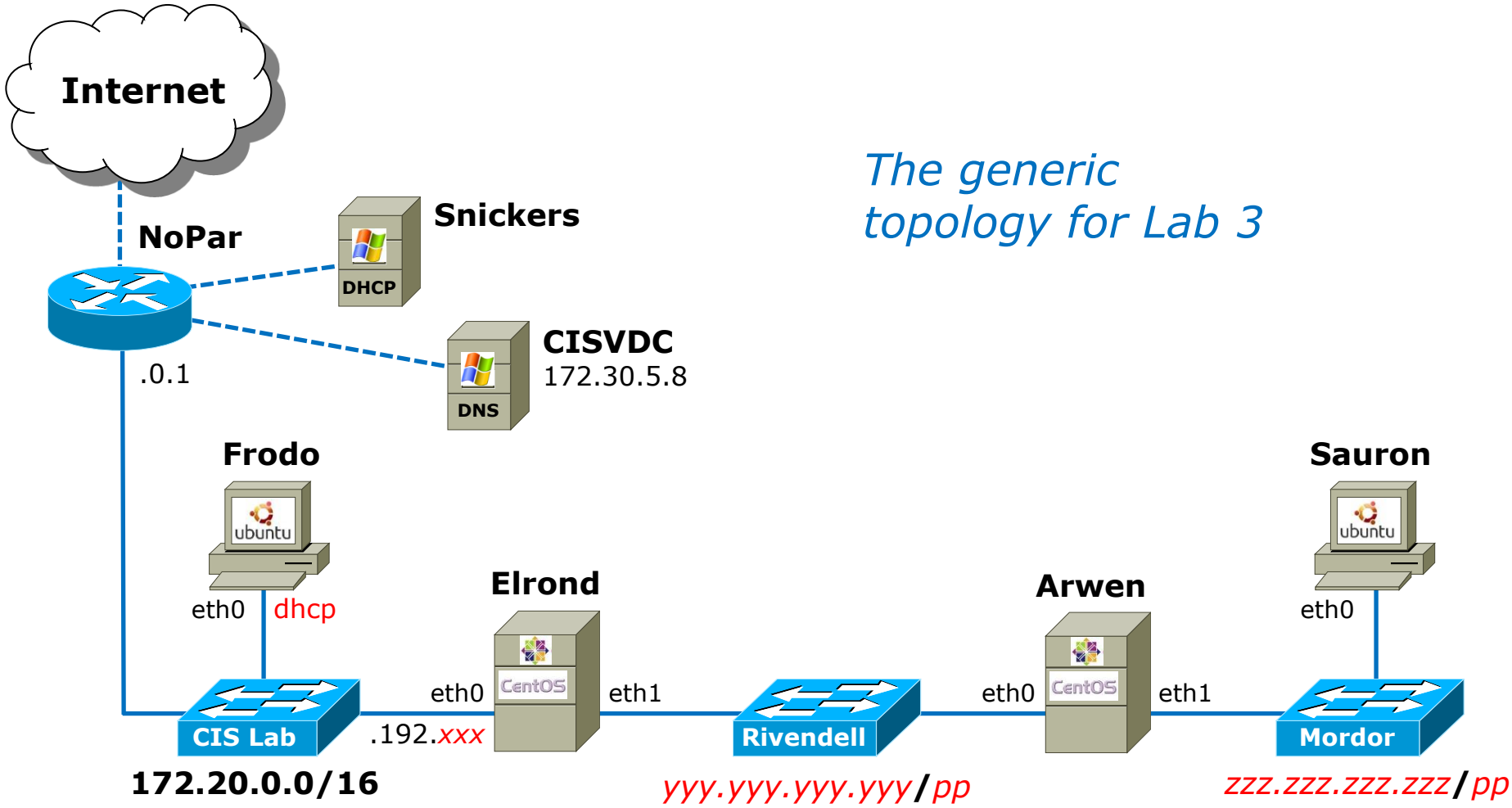
CCC Confer Whiteboard Activity

(Use the common symbols in the Clip Art

Instructions:

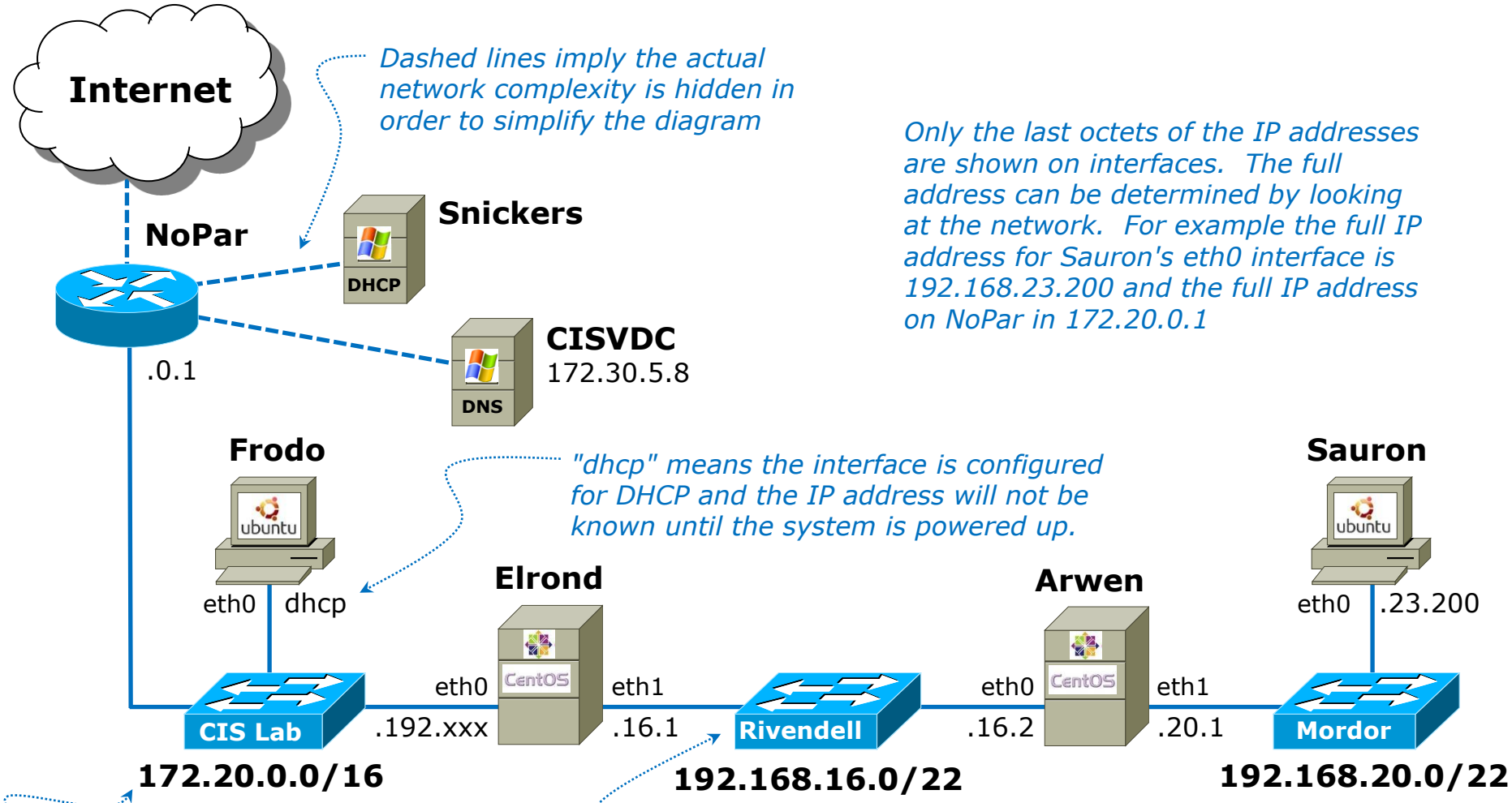
- If not on the white board already add name of your favorite drawing tool
- Add a  to drawing tools you have used
- Add a  next to your favorite tool(s)

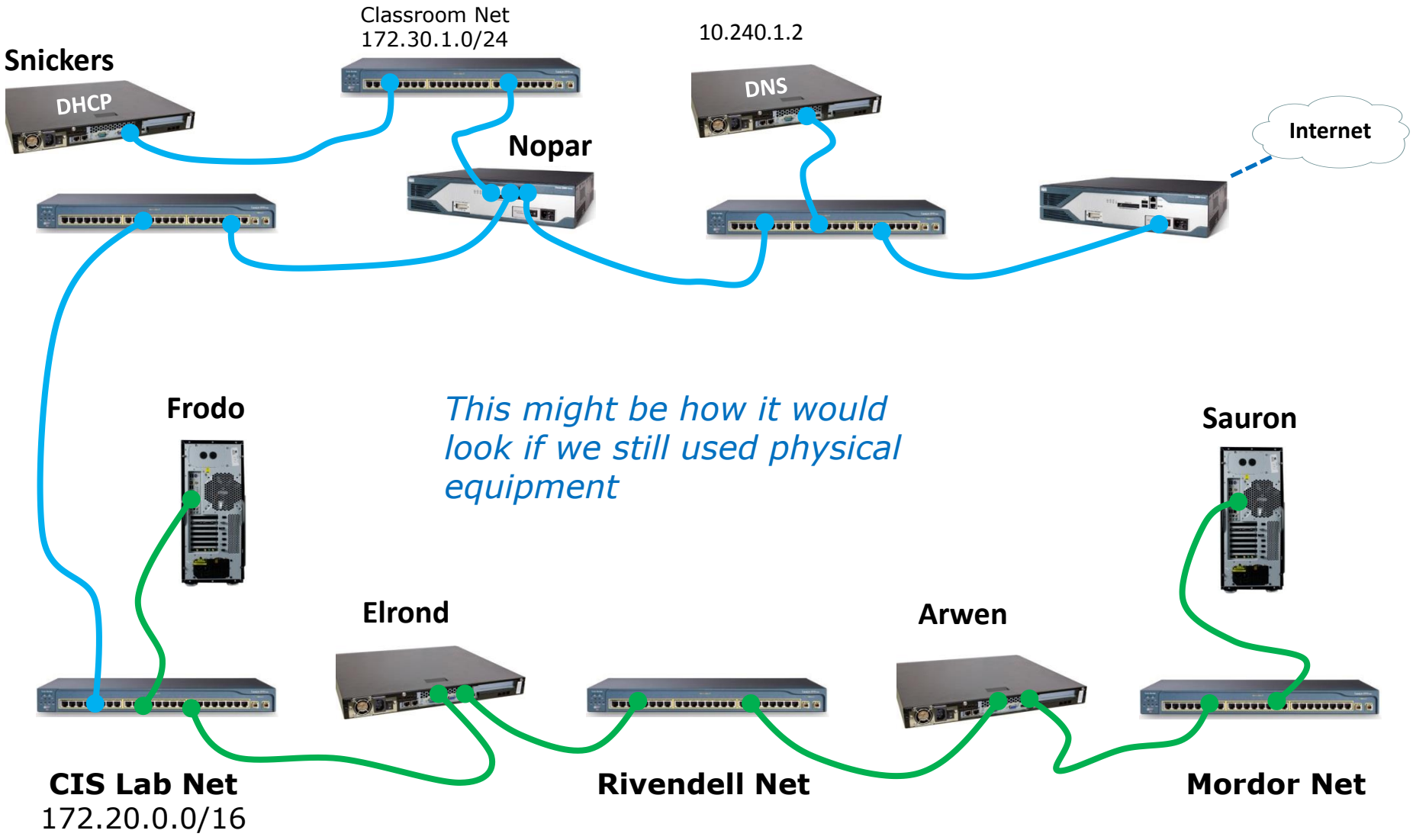
Generic Topology Lab 3



The generic topology for Lab 3

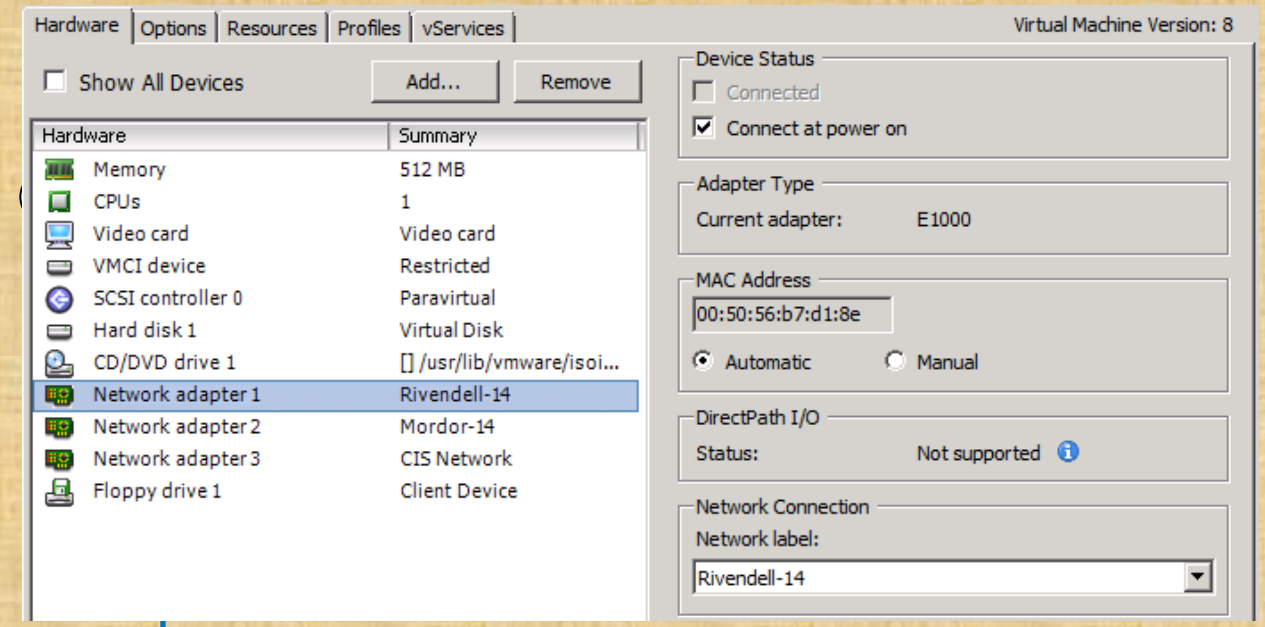
How to interpret





The green cables indicates the cabling you do in VLab

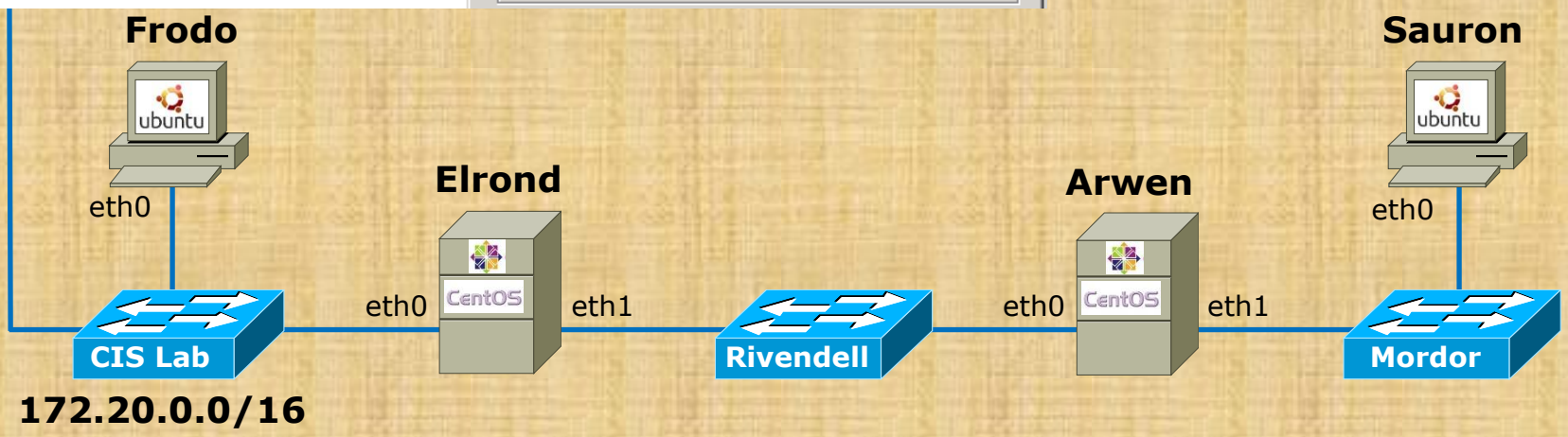
Activity



1) Which system is this?

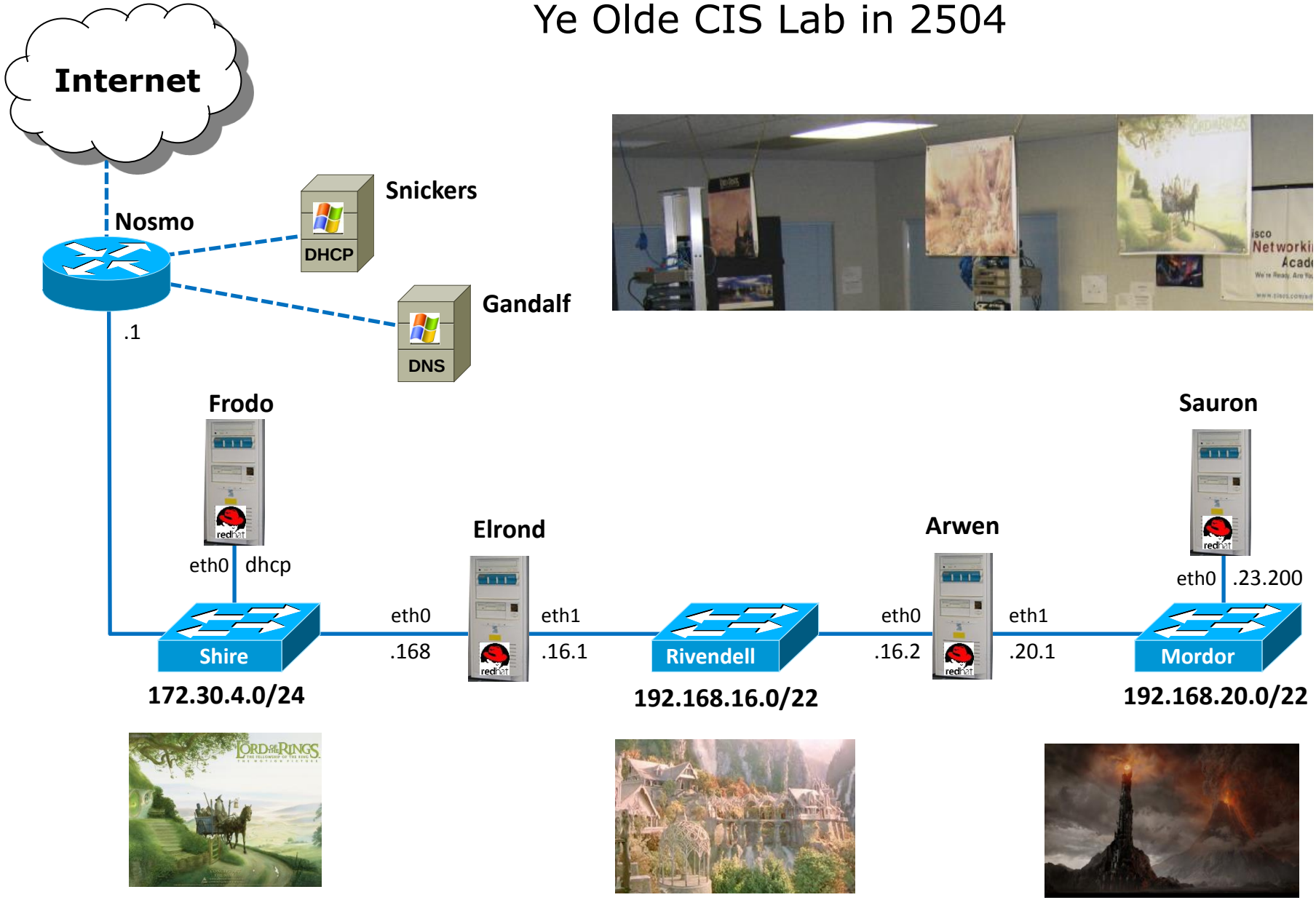
2) Who does it belong to?

Write your answers in the chat window



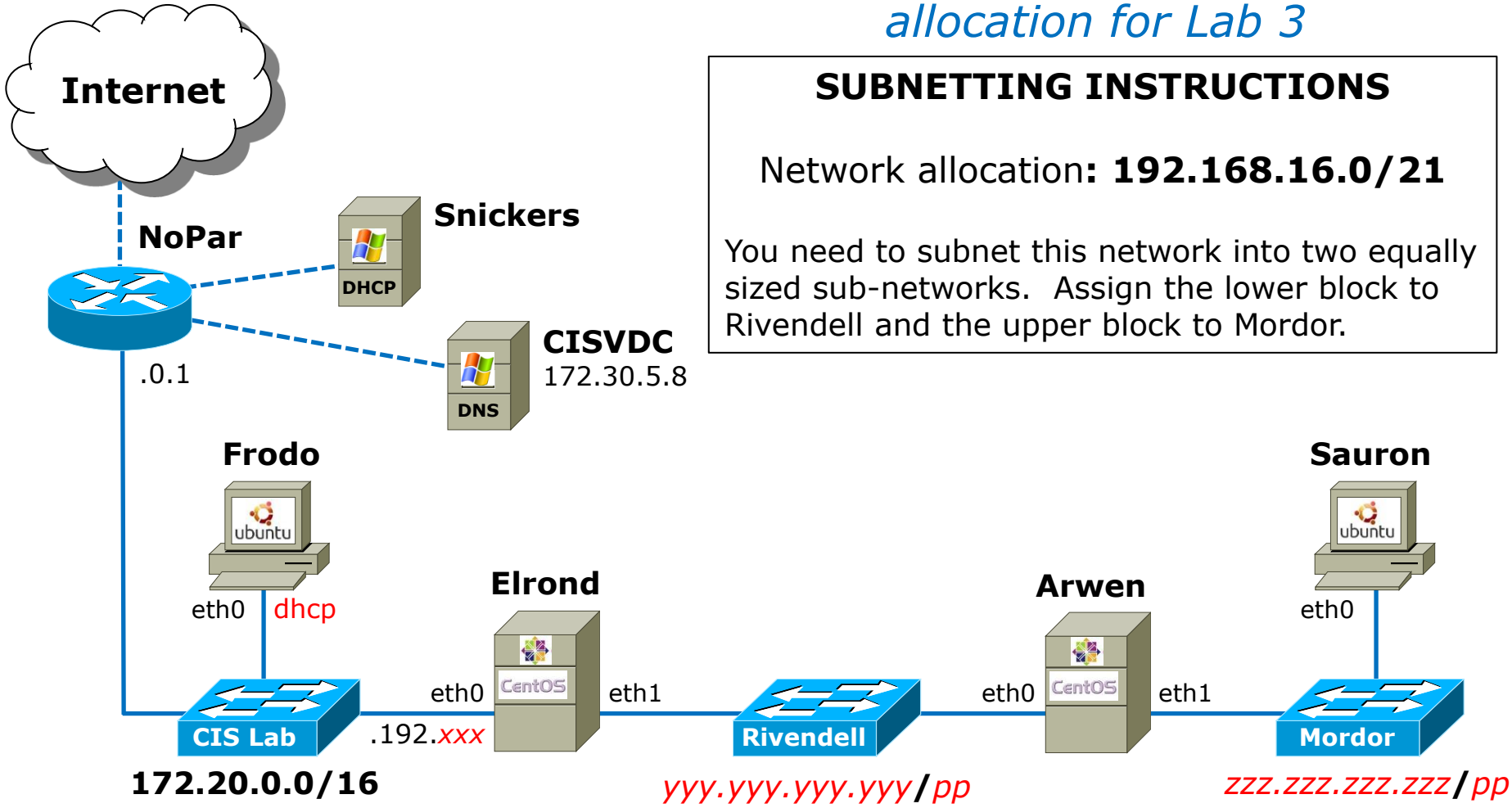
The LOR networks

Ye Olde CIS Lab in 2504



Lab 3 Prep

An example network allocation for Lab 3



SUBNETTING INSTRUCTIONS

Network allocation**: 192.168.16.0/21**

You need to subnet this network into two equally sized sub-networks. Assign the lower block to Rivendell and the upper block to Mordor.

Subnetting the networks

1) To subnet a network into two equally sized sub-networks add one bit to the netmask

Rivendell subnet

```
root@p14-frodo:~# ipcalc 192.168.16.0/22
Address:   192.168.16.0          11000000.10101000.000100 00.00000000
Netmask:   255.255.252.0 = 22    11111111.11111111.111111 00.00000000

Network:   192.168.16.0/22       11000000.10101000.000100 00.00000000
HostMin:   192.168.16.1          11000000.10101000.000100 00.00000001
HostMax:   192.168.19.254        11000000.10101000.000100 11.11111110
Broadcast: 192.168.19.255        11000000.10101000.000100 11.11111111
Hosts/Net: 1022
```

Allocated network

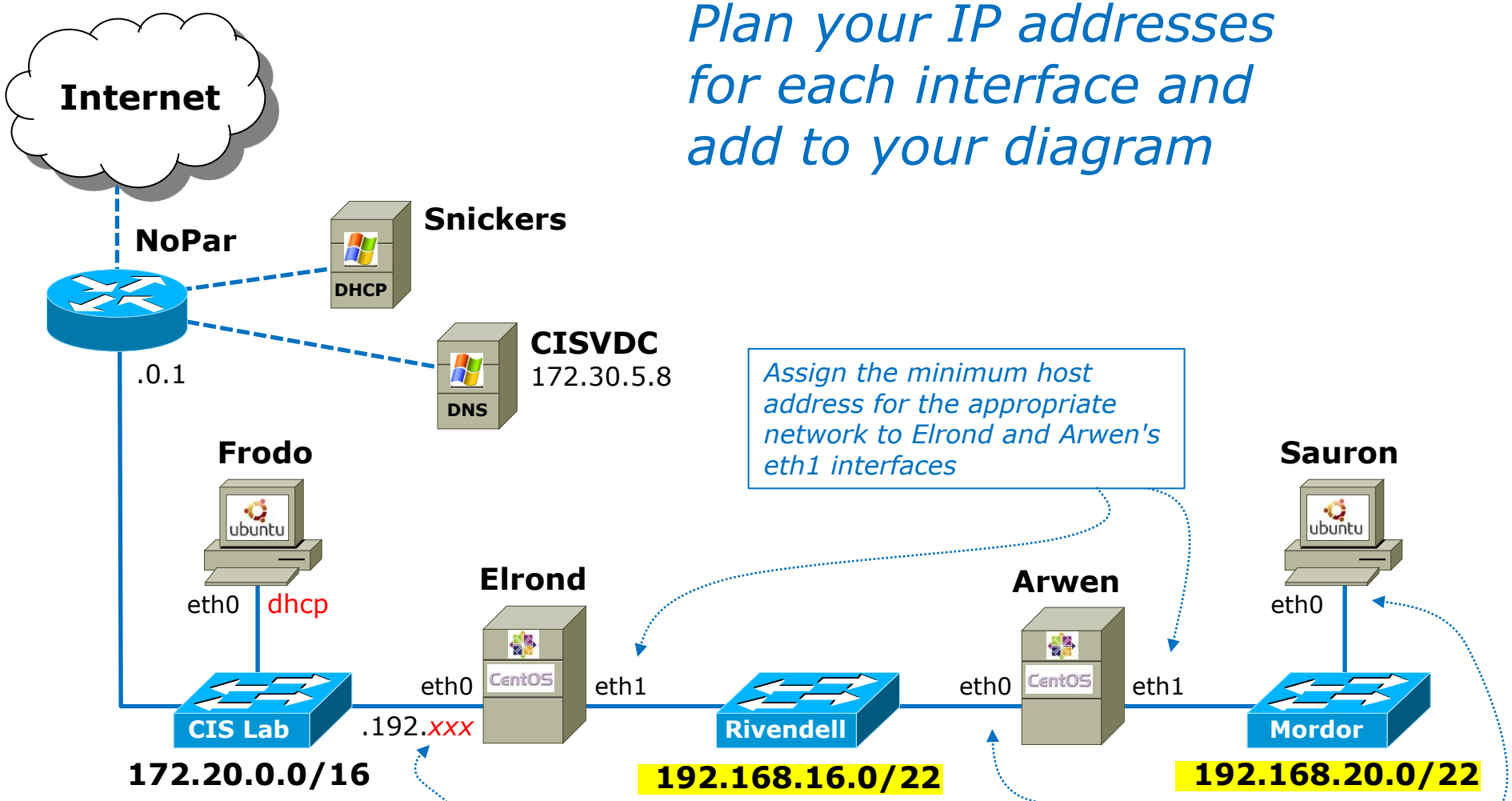
```
root@p14-frodo:~# ipcalc 192.168.16.0/21
Address:   192.168.16.0          11000000.10101000.00010 000.00000000
Netmask:   255.255.248.0 = 21    11111111.11111111.11111 000.00000000

Network:   192.168.16.0/21       11000000.10101000.00010 000.00000000
HostMin:   192.168.16.1          11000000.10101000.00010 000.00000001
HostMax:   192.168.23.254        11000000.10101000.00010 111.11111110
Broadcast: 192.168.23.255        11000000.10101000.00010 111.11111111
Hosts/Net: 2046
```

2) The second subnet will follow the first subnet (immediately after the broadcast address of the first)

Mordor subnet

```
root@p14-frodo:~# ipcalc 192.168.20.0/22
Address:   192.168.20.0          11000000.10101000.000101 00.00000000
Netmask:   255.255.252.0 = 22    11111111.11111111.111111 00.00000000

Network:   192.168.20.0/22       11000000.10101000.000101 00.00000000
HostMin:   192.168.20.1          11000000.10101000.000101 00.00000001
HostMax:   192.168.23.254        11000000.10101000.000101 11.11111110
Broadcast: 192.168.23.255        11000000.10101000.000101 11.11111111
Hosts/Net: 1022
```

Plan your IP addresses for each interface and add to your diagram



Getting first and last host addresses on each network

Rivendell subnet

```
root@p14-frodo:~# ipcalc 192.168.16.0/22
Address:   192.168.16.0          11000000.10101000.000100 00.00000000
Netmask:   255.255.252.0 = 22    11111111.11111111.111111 00.00000000

Network:   192.168.16.0/22       11000000.10101000.000100 00.00000000
HostMin:   192.168.16.1          11000000.10101000.000100 00.00000001
HostMax:   192.168.19.254        11000000.10101000.000100 11.11111110
Broadcast: 192.168.19.255        11000000.10101000.000100 11.11111111
Hosts/Net: 1022
```

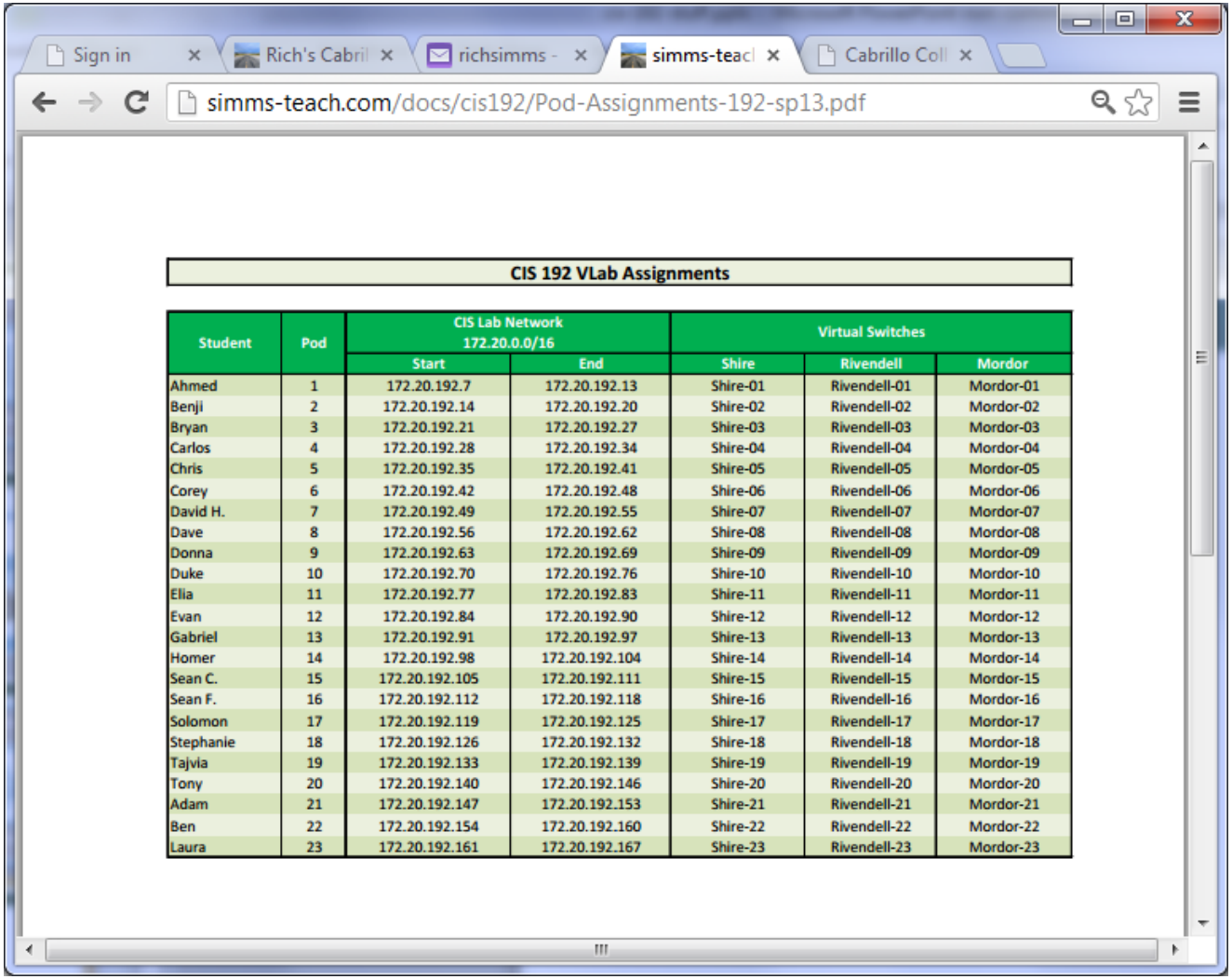
Allocated network

```
root@p14-frodo:~# ipcalc 192.168.16.0/21
Address:   192.168.16.0          11000000.10101000.00010 000.00000000
Netmask:   255.255.248.0 = 21    11111111.11111111.11111 000.00000000

Network:   192.168.16.0/21       11000000.10101000.00010 000.00000000
HostMin:   192.168.16.1          11000000.10101000.00010 000.00000001
HostMax:   192.168.23.254        11000000.10101000.00010 111.11111110
Broadcast: 192.168.23.255        11000000.10101000.00010 111.11111111
Hosts/Net: 2046
```

Mordor subnet

```
root@p14-frodo:~# ipcalc 192.168.20.0/22
Address:   192.168.20.0          11000000.10101000.000101 00.00000000
Netmask:   255.255.252.0 = 22    11111111.11111111.111111 00.00000000

Network:   192.168.20.0/22       11000000.10101000.000101 00.00000000
HostMin:   192.168.20.1          11000000.10101000.000101 00.00000001
HostMax:   192.168.23.254        11000000.10101000.000101 11.11111110
Broadcast: 192.168.23.255        11000000.10101000.000101 11.11111111
Hosts/Net: 1022
```

Use the CIS 192 Pod Assignments table to select a static IP address for Elrond's eth0 interface



CIS 192 VLab Assignments

Student	Pod	CIS Lab Network 172.20.0.0/16		Virtual Switches		
		Start	End	Shire	Rivendell	Mordor
Ahmed	1	172.20.192.7	172.20.192.13	Shire-01	Rivendell-01	Mordor-01
Benji	2	172.20.192.14	172.20.192.20	Shire-02	Rivendell-02	Mordor-02
Bryan	3	172.20.192.21	172.20.192.27	Shire-03	Rivendell-03	Mordor-03
Carlos	4	172.20.192.28	172.20.192.34	Shire-04	Rivendell-04	Mordor-04
Chris	5	172.20.192.35	172.20.192.41	Shire-05	Rivendell-05	Mordor-05
Corey	6	172.20.192.42	172.20.192.48	Shire-06	Rivendell-06	Mordor-06
David H.	7	172.20.192.49	172.20.192.55	Shire-07	Rivendell-07	Mordor-07
Dave	8	172.20.192.56	172.20.192.62	Shire-08	Rivendell-08	Mordor-08
Donna	9	172.20.192.63	172.20.192.69	Shire-09	Rivendell-09	Mordor-09
Duke	10	172.20.192.70	172.20.192.76	Shire-10	Rivendell-10	Mordor-10
Elia	11	172.20.192.77	172.20.192.83	Shire-11	Rivendell-11	Mordor-11
Evan	12	172.20.192.84	172.20.192.90	Shire-12	Rivendell-12	Mordor-12
Gabriel	13	172.20.192.91	172.20.192.97	Shire-13	Rivendell-13	Mordor-13
Homer	14	172.20.192.98	172.20.192.104	Shire-14	Rivendell-14	Mordor-14
Sean C.	15	172.20.192.105	172.20.192.111	Shire-15	Rivendell-15	Mordor-15
Sean F.	16	172.20.192.112	172.20.192.118	Shire-16	Rivendell-16	Mordor-16
Solomon	17	172.20.192.119	172.20.192.125	Shire-17	Rivendell-17	Mordor-17
Stephanie	18	172.20.192.126	172.20.192.132	Shire-18	Rivendell-18	Mordor-18
Tajvia	19	172.20.192.133	172.20.192.139	Shire-19	Rivendell-19	Mordor-19
Tony	20	172.20.192.140	172.20.192.146	Shire-20	Rivendell-20	Mordor-20
Adam	21	172.20.192.147	172.20.192.153	Shire-21	Rivendell-21	Mordor-21
Ben	22	172.20.192.154	172.20.192.160	Shire-22	Rivendell-22	Mordor-22
Laura	23	172.20.192.161	172.20.192.167	Shire-23	Rivendell-23	Mordor-23

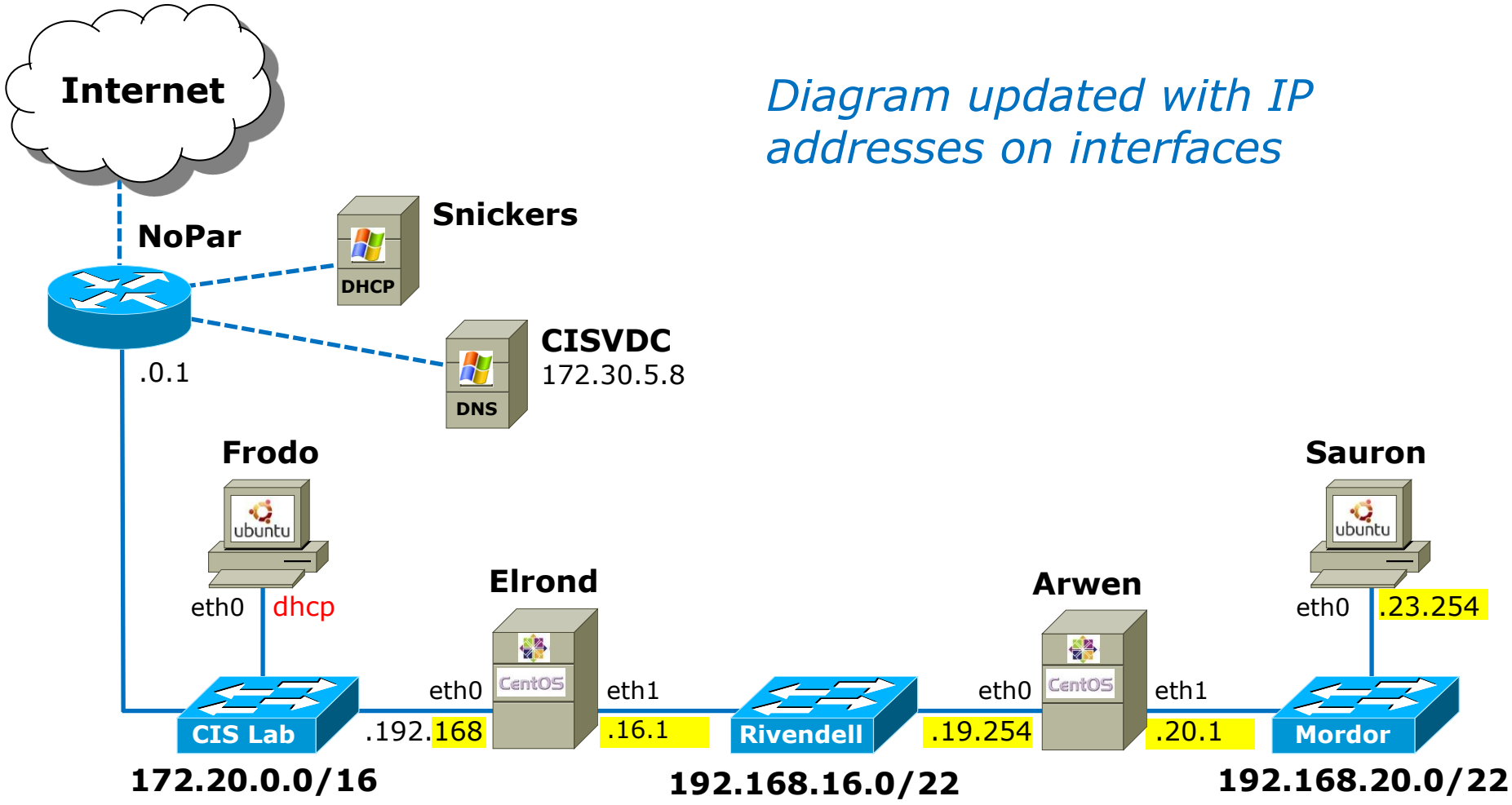


Diagram updated with IP addresses on interfaces

Activity

You have been given the **63.54.0.0/18** network

You are going to subnet it for three organizations, where the first organization (Org 1) gets half of the addresses (the lower block) and the other two (Org 2 & Org 3) split the remainder.

What are the three subnets and how many hosts are allowed on each?

```
root@p24-frodo:~# ipcalc 63.54.0.0/18
Address:   63.54.0.0            00111111.00110110.00 000000.00000000
Netmask:   255.255.192.0 = 18   11111111.11111111.11 000000.00000000
Network:   63.54.0.0/18         00111111.00110110.00 000000.00000000
HostMin:   63.54.0.1            00111111.00110110.00 000000.00000001
HostMax:   63.54.63.254         00111111.00110110.00 111111.11111110
Broadcast: 63.54.63.255         00111111.00110110.00 111111.11111111
Hosts/Net: 16382
```

Add one bit to the mask to cut the allocated network in half for the first subnet for Org 1

Org 1 subnet

```
root@p24-frodo:~# ipcalc 63.54.0.0/19
Address:   63.54.0.0            00111111.00110110.000 00000.00000000
Netmask:   255.255.224.0 = 19   11111111.11111111.111 00000.00000000
Wildcard:  0.0.31.255           00000000.00000000.000 11111.11111111
Network:   63.54.0.0/19         00111111.00110110.000 00000.00000000
HostMin:   63.54.0.1            00111111.00110110.000 00000.00000001
HostMax:   63.54.31.254         00111111.00110110.000 11111.11111110
Broadcast: 63.54.31.255         00111111.00110110.000 11111.11111111
Hosts/Net: 8190
```

Starting after the last broadcast address of Org 1's subnet and adding another bit to the mask to cut the next network in half for Org 2

Org 2 subnet

```
root@p24-frodo:~# ipcalc 63.54.32.0/20
Address:   63.54.32.0           00111111.00110110.0010 0000.00000000
Netmask:   255.255.240.0 = 20   11111111.11111111.1111 0000.00000000
Wildcard:  0.0.15.255           00000000.00000000.0000 1111.11111111
Network:   63.54.32.0/20        00111111.00110110.0010 0000.00000000
HostMin:   63.54.32.1           00111111.00110110.0010 0000.00000001
HostMax:   63.54.47.254         00111111.00110110.0010 1111.11111110
Broadcast: 63.54.47.255         00111111.00110110.0010 1111.11111111
Hosts/Net: 4094
```

Starting after the last broadcast address of the Org 2's subnet and using the same mask calculate the last subnet for Org 3

Org 3 subnet

```
root@p24-frodo:~# ipcalc 63.54.48.0/20
Address:   63.54.48.0           00111111.00110110.0011 0000.00000000
Netmask:   255.255.240.0 = 20   11111111.11111111.1111 0000.00000000
Wildcard:  0.0.15.255           00000000.00000000.0000 1111.11111111
Network:   63.54.48.0/20        00111111.00110110.0011 0000.00000000
HostMin:   63.54.48.1           00111111.00110110.0011 0000.00000001
HostMax:   63.54.63.254         00111111.00110110.0011 1111.11111110
Broadcast: 63.54.63.255         00111111.00110110.0011 1111.11111111
Hosts/Net: 4094
```

Crib Sheets

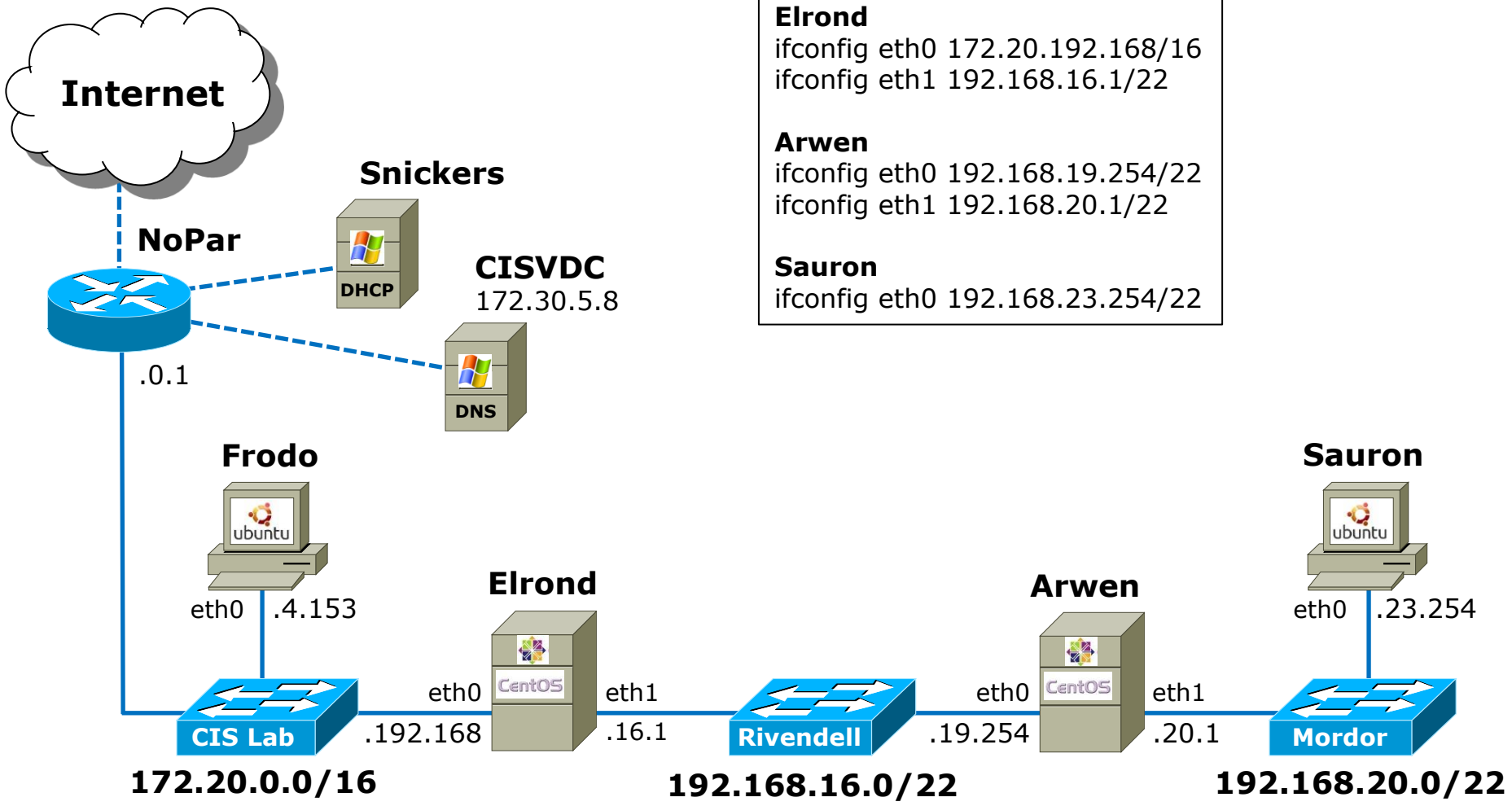
Start a crib sheet of the commands needed to implement your networks

CRIB SHEET

Elrond
ifconfig eth0 172.20.192.168/16
ifconfig eth1 192.168.16.1/22

Arwen
ifconfig eth0 192.168.19.254/22
ifconfig eth1 192.168.20.1/22

Sauron
ifconfig eth0 192.168.23.254/22



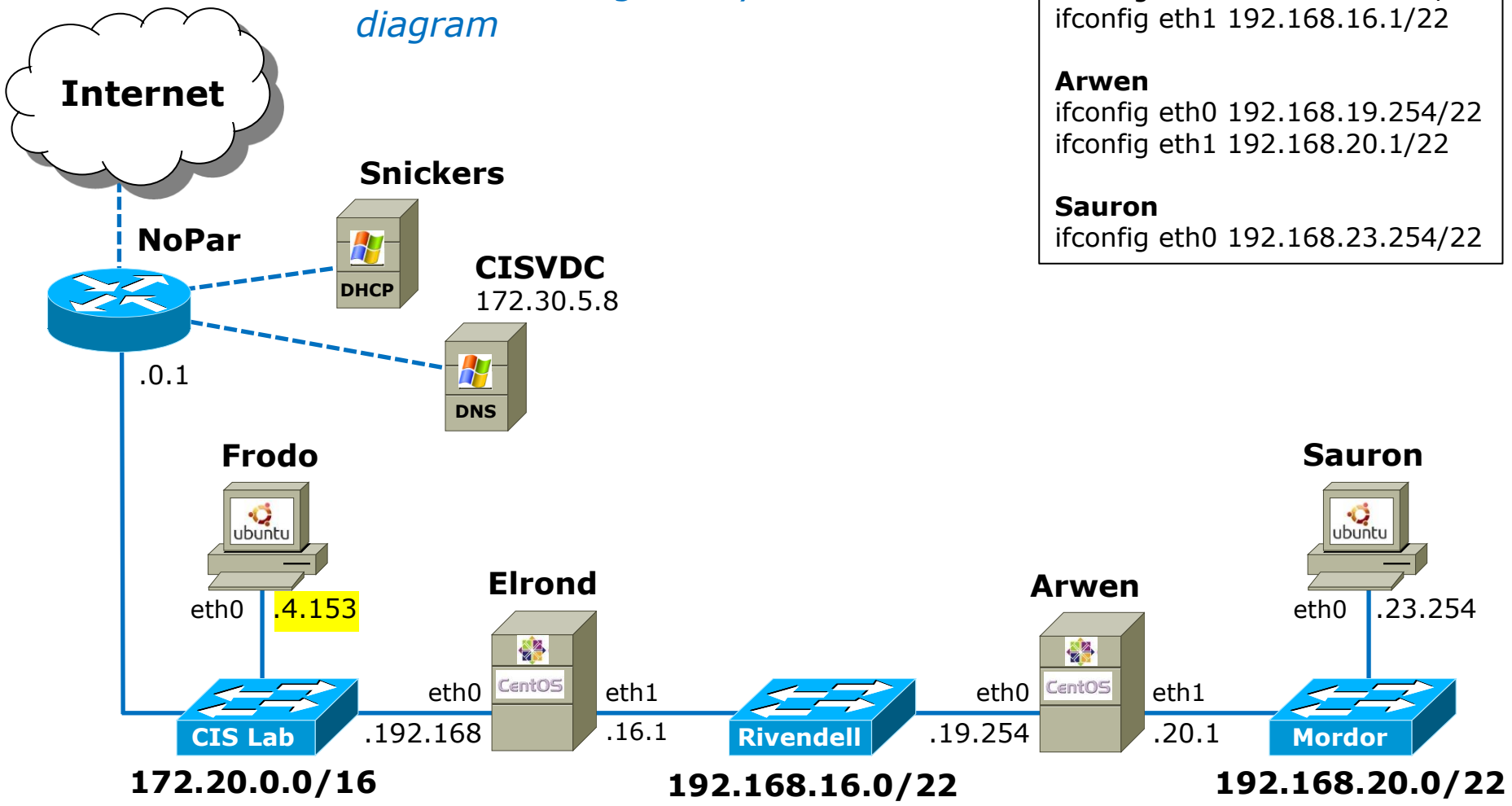
After powering on and configuring interfaces you can record the IP address Frodo got on your diagram

CRIB SHEET

Elrond
ifconfig eth0 172.20.192.168/16
ifconfig eth1 192.168.16.1/22

Arwen
ifconfig eth0 192.168.19.254/22
ifconfig eth1 192.168.20.1/22

Sauron
ifconfig eth0 192.168.23.254/22



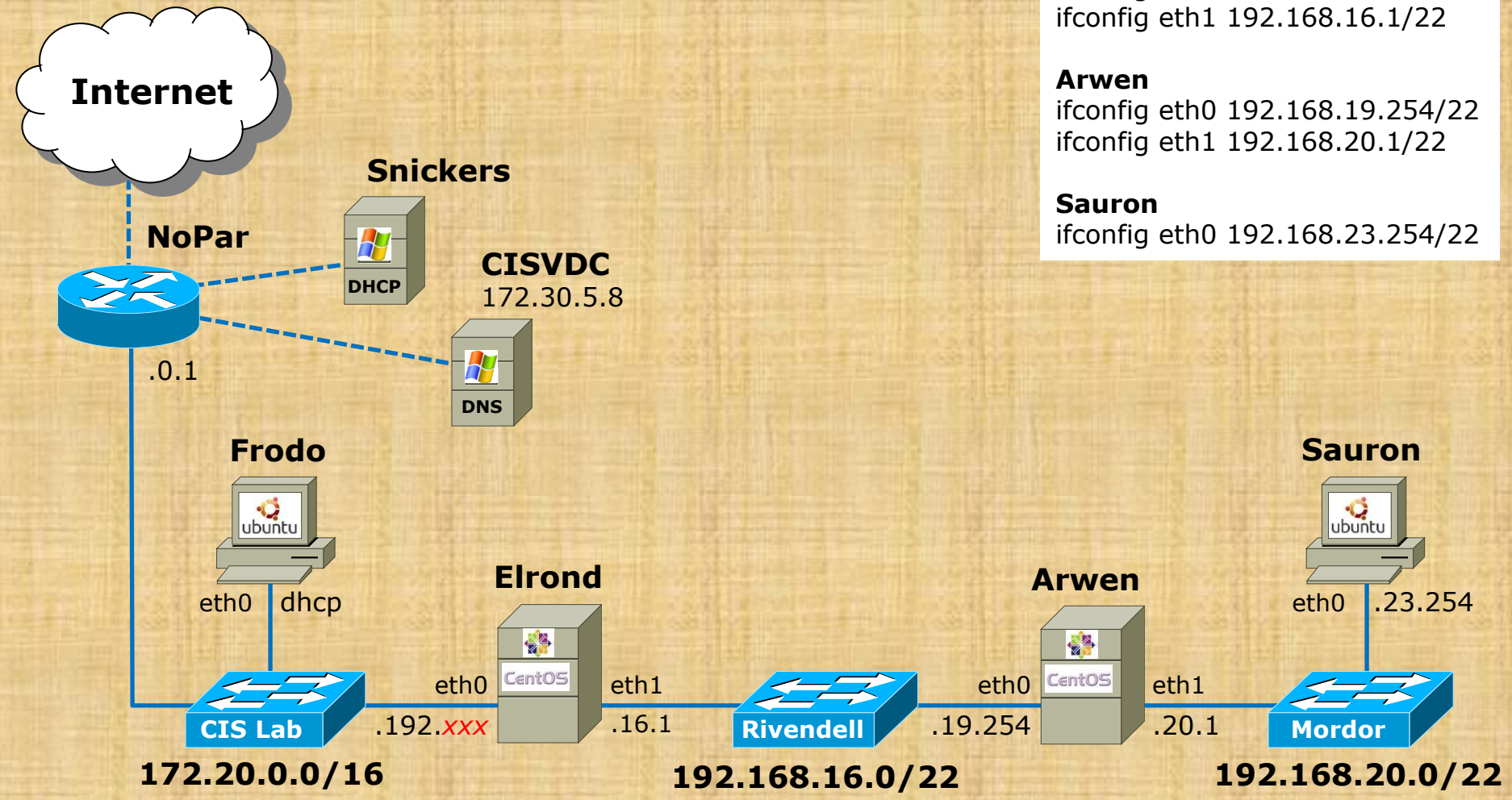
Your turn: Build this in your pod

CRIB SHEET

Elrond
ifconfig eth0 172.20.192.*xxx*/16
ifconfig eth1 192.168.16.1/22

Arwen
ifconfig eth0 192.168.19.254/22
ifconfig eth1 192.168.20.1/22

Sauron
ifconfig eth0 192.168.23.254/22



Housekeeping

Help with labs

Like some help with labs?

I'm in the CIS Lab Monday afternoons
- See schedule at http://webhawks.org/~cislab/

or see me during office hours

or contact me to arrange another time online

Turning in work and getting graded work back

Benji (simben192) submits his lab2 for grading

```
[simben192@opus ~]$ cp lab02 /home/rsimms/turnin/cis192/lab02.$LOGNAME
```

Benji verifies his lab was submitted

```
[simben192@opus ~]$ ls /home/rsimms/turnin/cis192
lab01  lab02.simben192  lab02.visgab192  lab02.wildan192  lab02.winjas192
```

Benji's lab1 has been graded and placed in his directory

```
[simben192@opus ~]$ ls -l
total 52
-rw-r--r-- 1 simben192 cis192  610 Nov  7 08:51 capture
-rw-r--r-- 1 simben192 cis192  360 Nov  1 09:26 lab01
-r-------- 1 simben192 staff  4170 Nov  2 16:05 lab01.graded
-rw------- 1 simben192 cis192 3702 Nov  7 08:49 lab02
-rw-r--r-- 1 simben192 cis192 1350 Oct 31 19:05 labnotes
-rw-r--r-- 1 simben192 cis192 1400 Nov  1 13:15 notes
[simben192@opus ~]$
```

Grades Web Page

http://simms-teach.com/cis192grades.php

Code Name	Grading Choice	Quizzes & Tests													Forum				Labs										Final	Extra Credit	Total	Grade
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	T1	T2	T3	F1	F2	F3	F4	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10				
Max Points		3	3	3	3	3	3	3	3	3	3	30	30	30	20	20	20	20	30	30	30	30	30	30	30	30	30	30	60	90	560	
Aragorn	Grade	2																	30											3		
Bilbo	Grade	3																	29											3		
Denethor	Grade	3																	8													
Dwalin	Grade																															
Elrohir	Grade	3																	30											5		
Elrond	Grade	3																	30											3		
Eomer	Grade																															
Faramir	Grade	3																	30													
Frodo	Grade	3																	29													
Gwaihir	Grade																		30													
Ioreth	Grade	3																	30													
Legolas	Grade	3																	30													
Nazgul	Grade	3																	30											3		
Pippin	Grade	3																	30											5		
Samwise	Grade	3																	30											3		
Saruman	Grade	3																	30											3		
Strider	Grade	3																	29											3		
Theoden	Grade	3																	30											3		
Treebeard	Grade																															

Please check your:

- Grading Choice
- Quiz #1 points
- Lab #1 points
- Extra Credit points

Don't know you secret LOR code name?

… then email me your student survey to get it!

Grades Web Page

How your grade is determined:

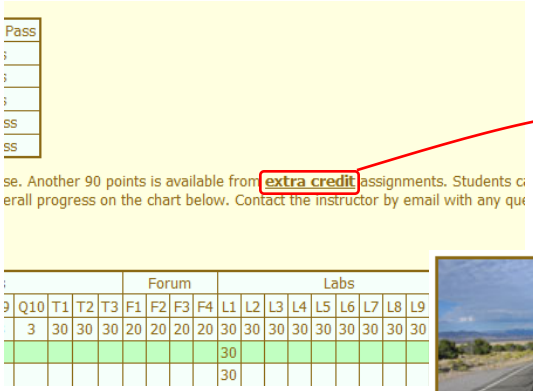
A student can earn up to 560 total points doing the activities listed above. The course grade is based on the number of points earned.

Percentage	Total Points	Letter Grade	Pass/No Pass
90% or higher	504 or higher	A	Pass
80% to 89.9%	448 to 503	B	Pass
70% to 79.9%	392 to 447	C	Pass
60% to 69.9%	336 to 391	D	No pass
0% to 59.9%	0 to 335	F	No pass

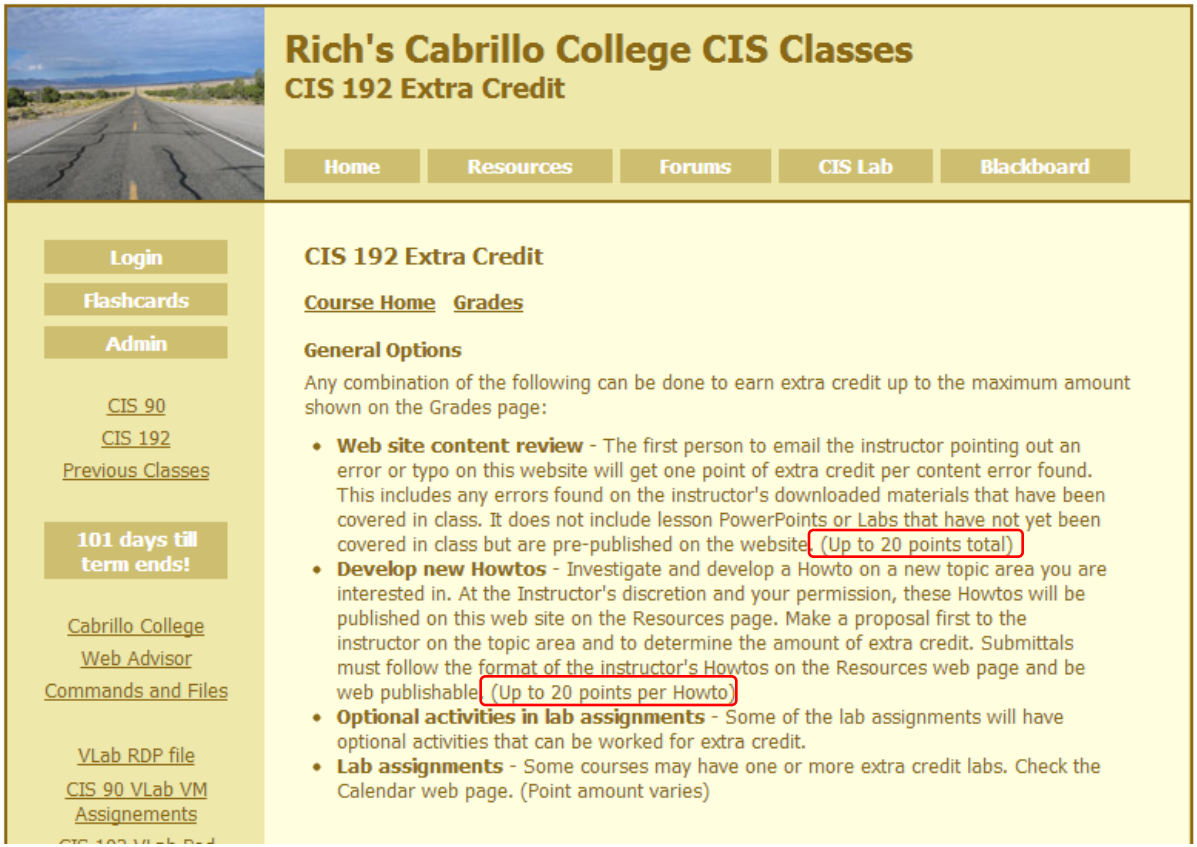
For some flexibility, personal preferences or family emergencies there is an additional 90 points available of **extra credit** activities.

Select the grade you want and earn that many points!

Extra Credit



Link to Extra Credit page is on the Grades page



Note the cap on all extra credit is 90 points

Stay on top of deliverables with the Calendar web page

Be ready for the first minute quizzes

3	2/26	**Quiz 2** **IP Routing and** • Describe the • Describe the • Describe how accomplished • Use the Simp... a routing table • View, add, and delete entries in a routing table • By properly configuring routing tables on hosts and routers, configure a LAN of multiple segments which allows all hosts to communicate with each other. **Materials** • Presentation slides (download) **TBA Assignment** • Lab 3 (Routing) **CCC Confer** • Enter virtual classroom • Class archives	14 15	Lab 2
4	3/5	**Quiz 3** **TCP and the Transport Layer** • SBCs, Dynamips/Dynagen, VirtualBox • Dynamic routing • Quagga routing suite • RIPv2 implementation skills • Transport layer • TCP and UDP protocols • Service ports and sockets **Materials** • Presentation slides (download) **TBA Assignment** • Practice Test 1 • Lab 4 (Dynamic Routing and Port Forwarding) **CCC Confer** • Enter virtual classroom • Class archives	15	Lab 3 5 posts

Lab 2 is due 11:59PM tonight

Lab 3 and five forum posts are due 11:59PM March 5th

First test is the following week!

Routing

Routing Summary

sign post

```
[root@lilly ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.15.48     0.0.0.0         255.255.255.240 U     0      0        0 eth1
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.1.1      0.0.0.0         UG    0      0        0 eth0
[root@lilly ~]#
```

routing table

- **Routers** operate at **layer 3** and make decisions on where to send a packet.
- The routing decision is based on the routing table.
- If there is no route, the packet is dropped

Now let's looks at some of the details

Routers and the Network Layer (Layer 3)

Cisco commercial Router (IOS)

Linux Server with multiple NICs

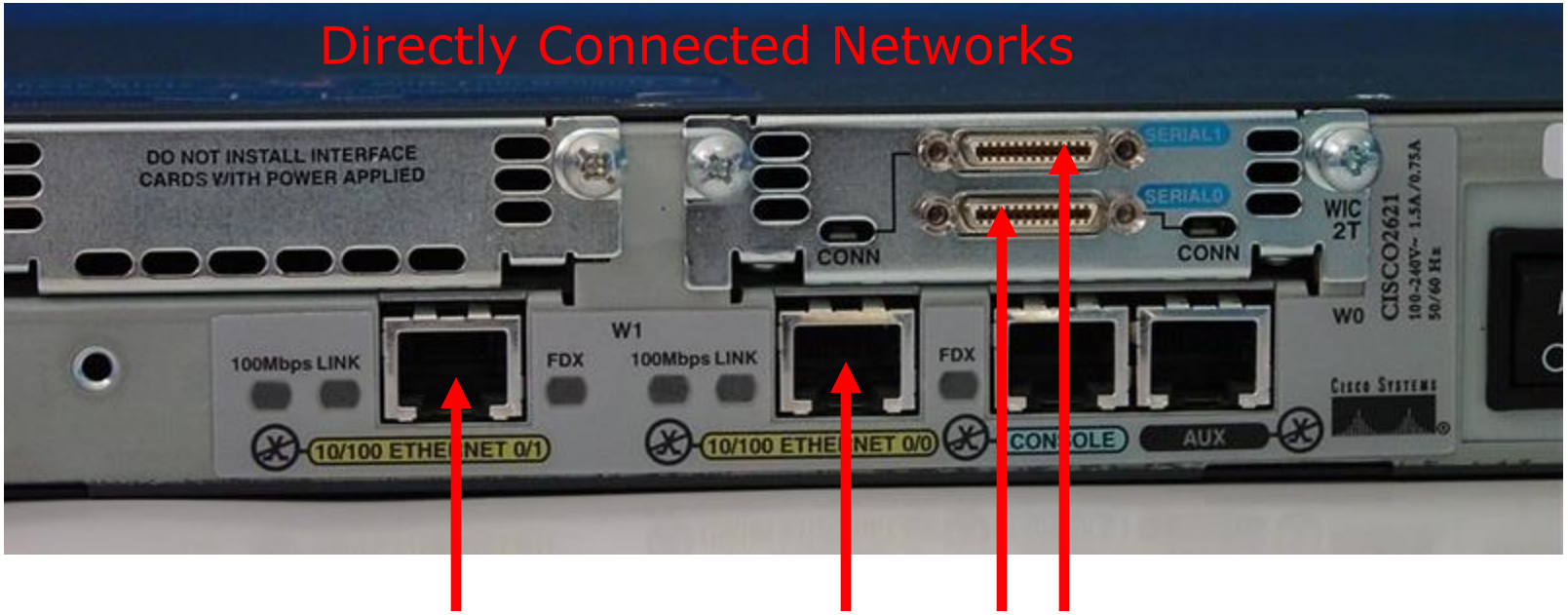
Linksys home Router

Routerboard “kit” with MikroTik RouterOS (based on Linux 2.6 kernel)

Routers and the Network Layer (Layer 3)

Routers

- Networking devices that make best path decisions (which interface to forward the IP packet) based in Layer 3 IP Destination Address.
- Routers connect multiple networks.

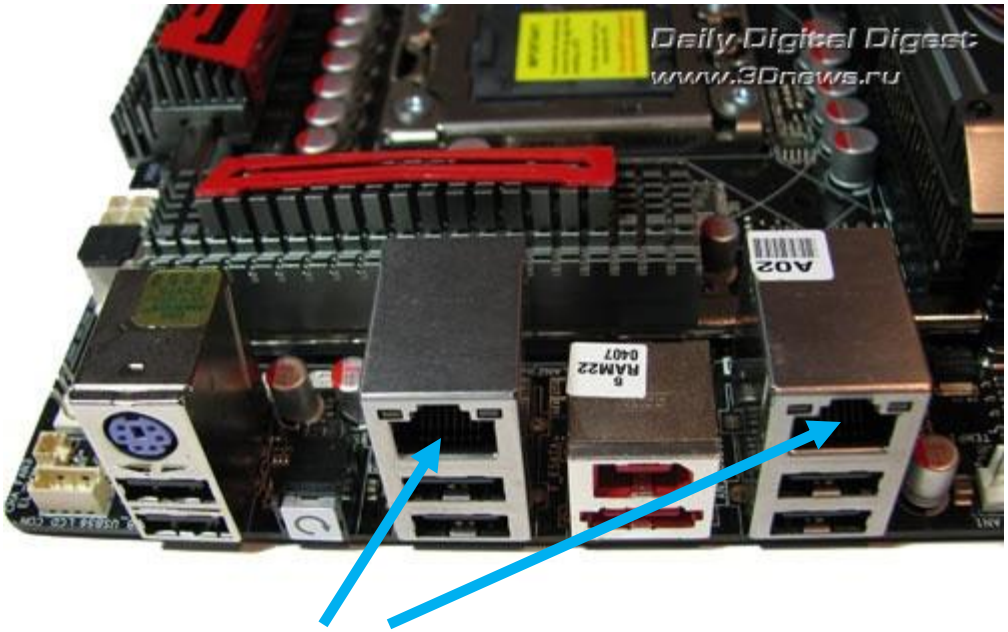


Each interface connects to a different network. Each interface has an IP address/mask for that network.

Routers and the Network Layer (Layer 3)

Physical Linux routers (a computer with multiple NICs)

- Networking devices that make best path decisions (which interface to forward the IP packet) based in Layer 3 IP Destination Address.
- Linux routers connect multiple networks.



Directly Connected Networks

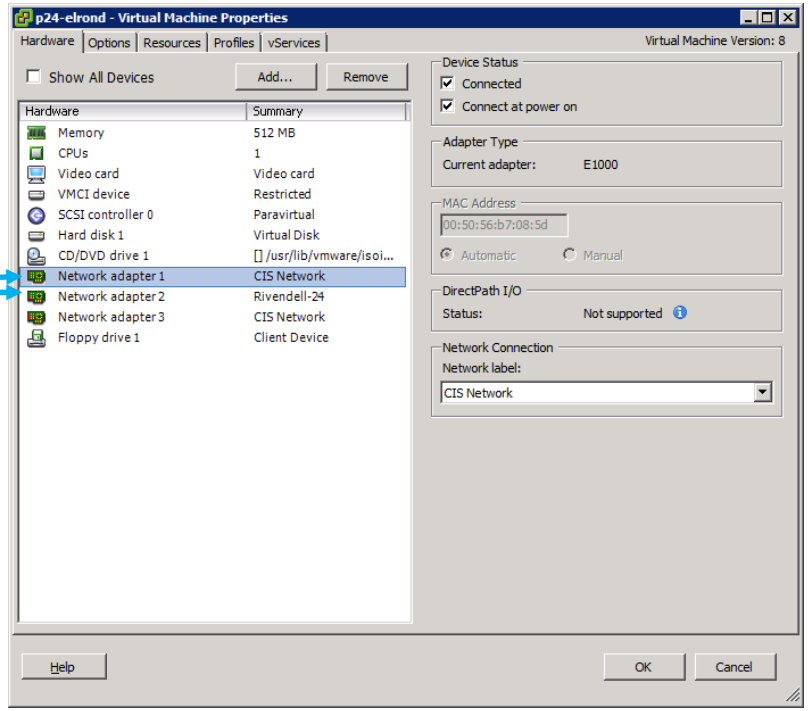
Each interface connects to a different network. Each interface has an IP address/mask for that network.

Routers and the Network Layer (Layer 3)

Virtual Linux routers

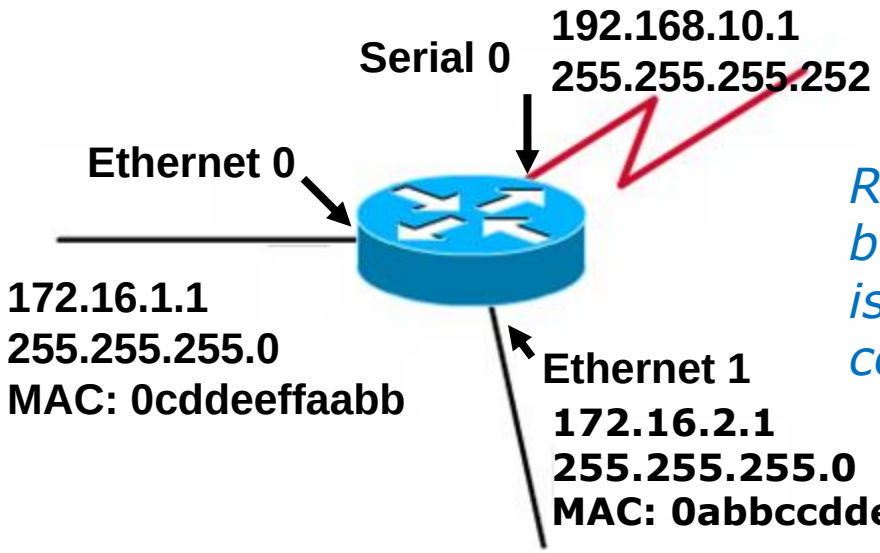
- Networking devices that make best path decisions (which interface to forward the IP packet) based in Layer 3 IP Destination Address.
- Virtual Linux routers connect multiple networks.

Each interface connects to a different network. Each interface has an IP address/mask for that network.



Directly Connected Networks

Routers belong to networks

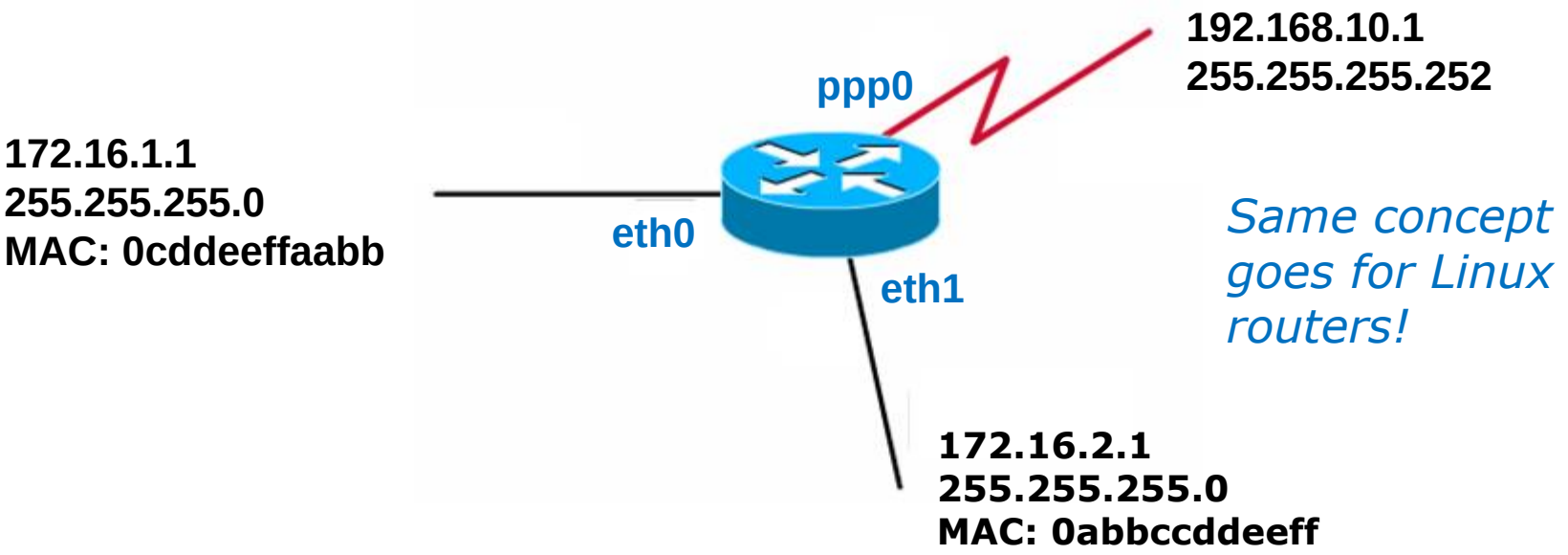


Router interfaces belonging to networks is a very important concept!

Directly Connected Networks

- Router interfaces must be members of different networks.
- Router interfaces participate in the network like other hosts on that network.
- Ethernet interfaces:
  - Have MAC Addresses
  - ARP Tables
  - Participate in the ARP Request and ARP Reply process like other hosts on that network.

Linux routers belong to networks



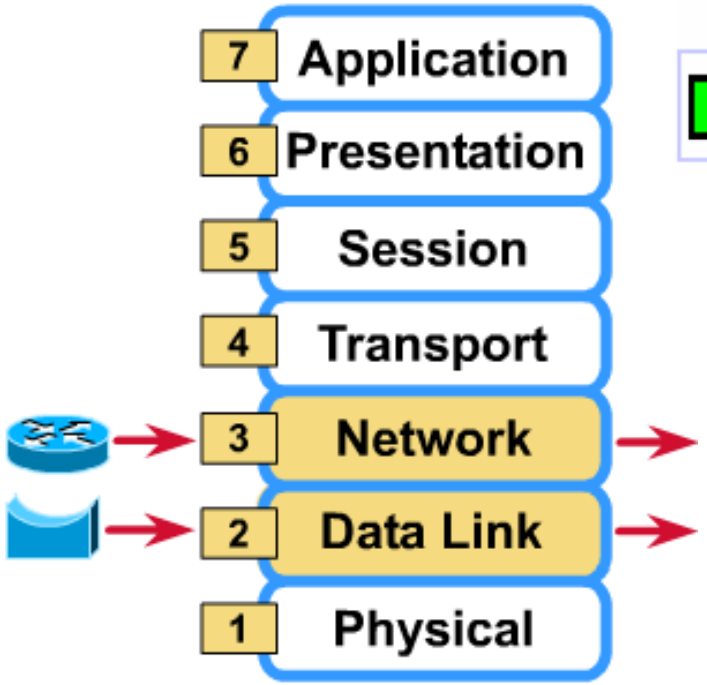
Directly Connected Networks

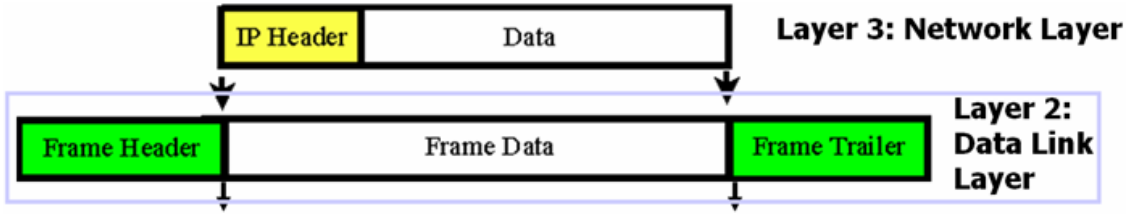
- Linux router interfaces must be members of different networks.
- Router interfaces participate in the network like other hosts on that network.
- Ethernet interfaces:
  - Have MAC Addresses
  - ARP Tables
  - Participate in the ARP Request and ARP Reply process like other hosts on that network.

Rick Graziani modified by Rich Simms
graziani@cabrillo.edu

Routers

- Make routing decisions based on Layer 3 information:
  - Destination IP address

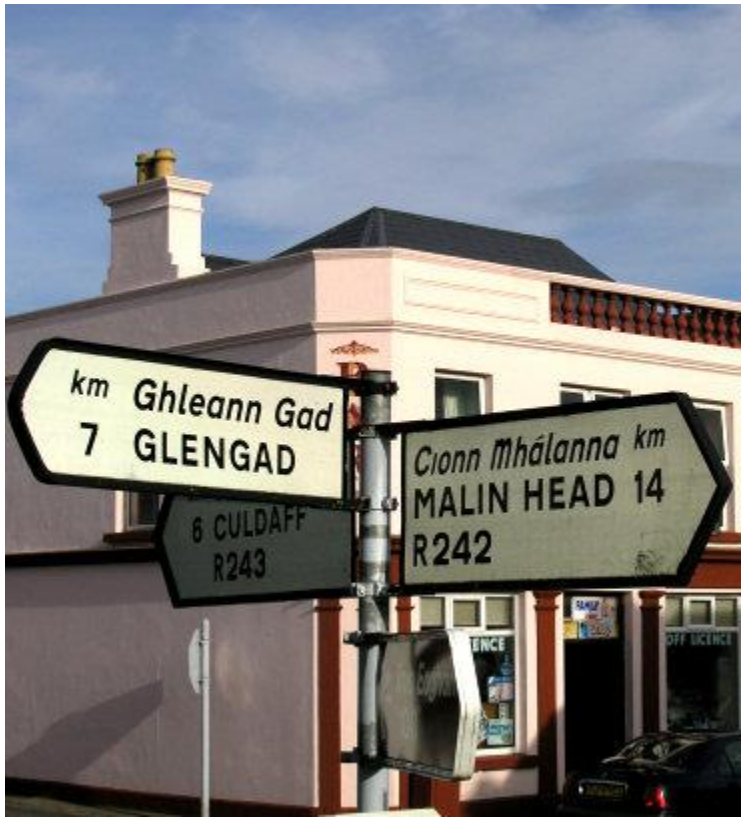




Routers make the routing decision by looking at the destination IP address and the routing table

Rick Graziani
graziani@cabrillo.edu

Routers and the Network Layer



- To get to GLENGAD **go left**
- To get to CULDAFF **go straight on**
- To get to MALIN HEAD **go back**

Using a routing table to make routing decisions is like using a signpost and deciding which direction to go

Note: if there is no sign for your destination you may be LOST!

Routers and the Network Layer

```
[root@lilly ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.15.48     0.0.0.0         255.255.255.240 U     0      0        0 eth1
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.1.1      0.0.0.0         UG    0      0        0 eth0
[root@lilly ~]#
```

this is a routing table

- To get to the 10.10.15.48/28 network **take the eth1 interface**
- To get to the 172.30.1.0/24 network **take the eth0 interface**
- To get to the 169.254.0.0/16 network **take the eth1 interface**
- For all other networks **proceed to the 172.30.1.1 router via the eth0 interface get more directions there**

Note: if there is no route to the destination the router will DROP the packet!

Routing Types

Two types of routes … static or dynamic

- A router must learn about non-directly connected networks either statically or dynamically.
- **Directly connected networks** are networks that the router is connected to, has an IP address/mask.
- **Non-directly connected networks** are remote networks connected to other routers.

Static

Uses a programmed route that a network administrator enters into the router

Dynamic

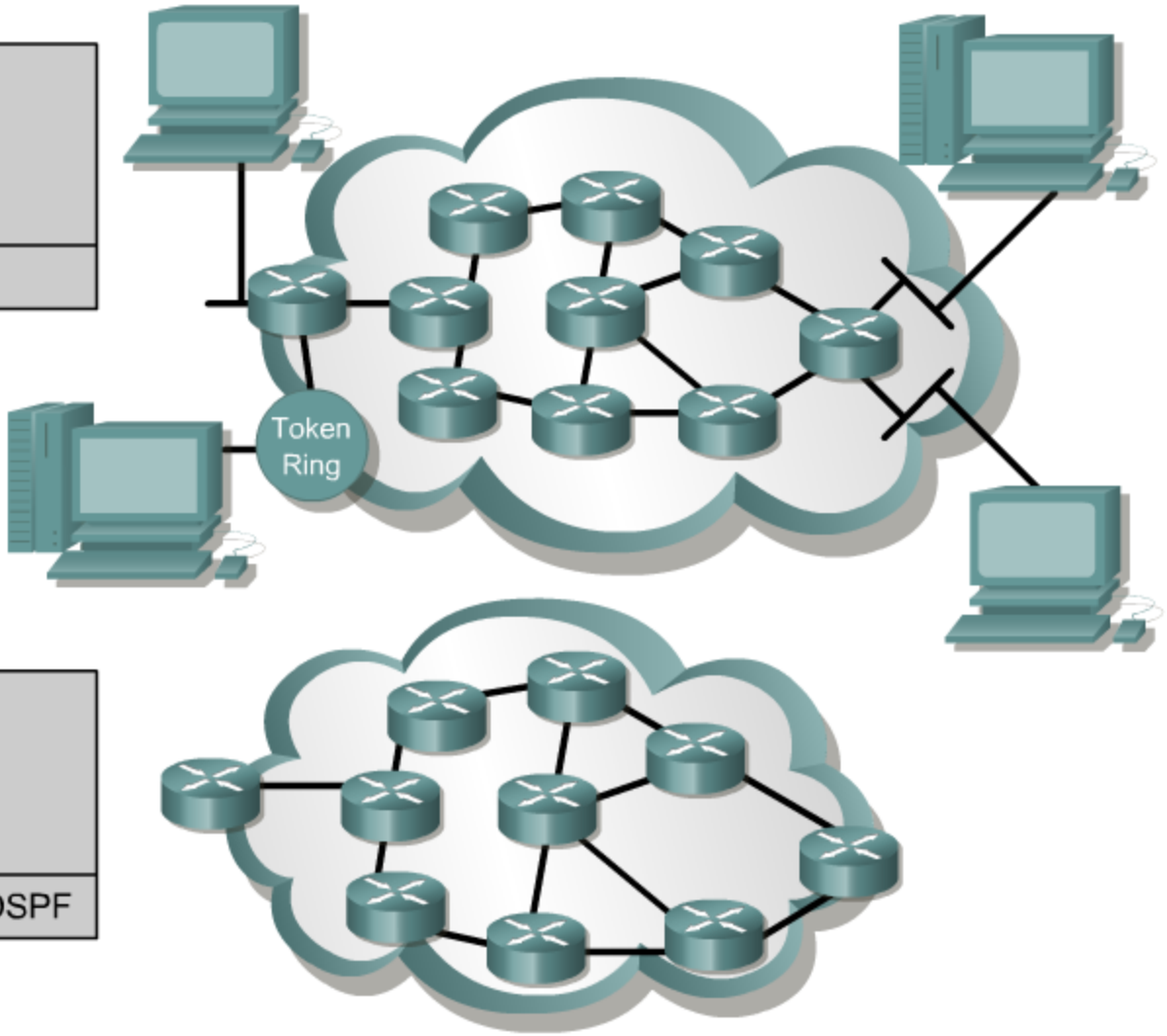
Uses a route that a routing protocol adjusts automatically for topology or traffic changes

Two types of destinations … directly connected or not-directly connected

Routed Protocols vs. Routing Protocols

Routed protocol used between routers to direct user traffic
Examples: IP and IPX

An important distinction!



Routing protocol used between routers to maintain tables
Examples: RIP, IGRP, OSPF

Rick Graziani
graziani@cabrillo.edu

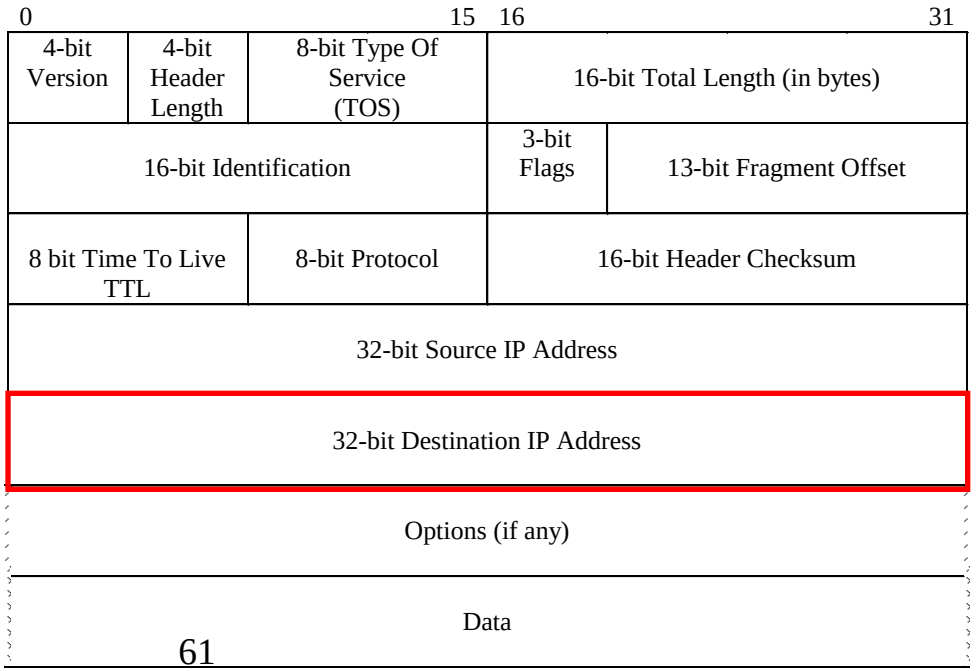
Routed Protocol

- IP is a routed protocol
- A routed protocol is a layer 3 protocol that contains network addressing information.
- This network addressing information is used by routers to determine the which interface, which next router, to forward this packet.

IP packets are routable because of the way IP address are organized and used around the world.

Ethernet frames are not routable. They would require massive routing tables and be an update nightmare!

IP Header



Rick Graziani
graziani@cabrillo.edu

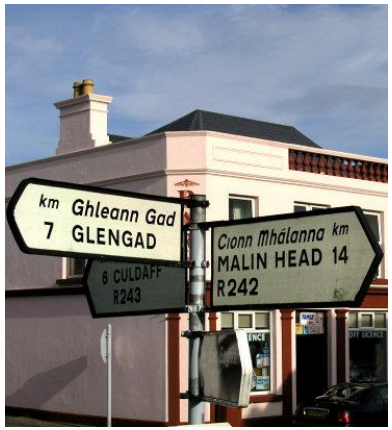
Routing Protocols

- Protocols used by routers to build routing tables.
- Routing tables are used by routers to forward packets.
  - **RIP**
  - **IGRP** and **EIGRP**
  - **OSPF**
  - **IS-IS**
  - **BGP**

This makes maintaining routing tables much more practical!

We will play with some of these next week!

Routing Summary



sign post

```
[root@lilly ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.15.48     0.0.0.0         255.255.255.240 U     0      0        0 eth1
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.1.1      0.0.0.0         UG    0      0        0 eth0
[root@lilly ~]#
```

routing table

- Routing is making a decision on where to send a packet.
- The decision is based on the routing table.
- If there is no route, the packet is dropped

Packet Forwarding

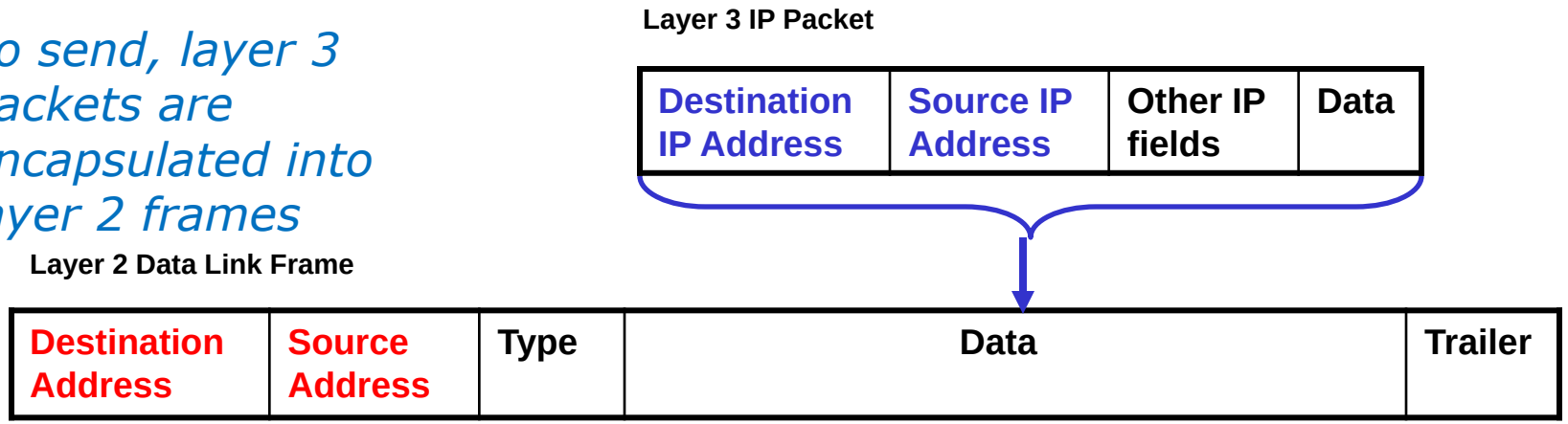
Packet Forwarding Summary

- Frame arrives.
- Layer 3 packet yanked (unencapsulated) from frame and the old frame is discarded.
- Routing decision is made using the destination IP address of the layer 3 packet and the routing table.
- A new layer 2 frame is created containing (encapsulating) the layer 3 packet.
- The new frame is sent out the interface determined by the routing decision.

Now let's looks at some of the details

Encapsulation

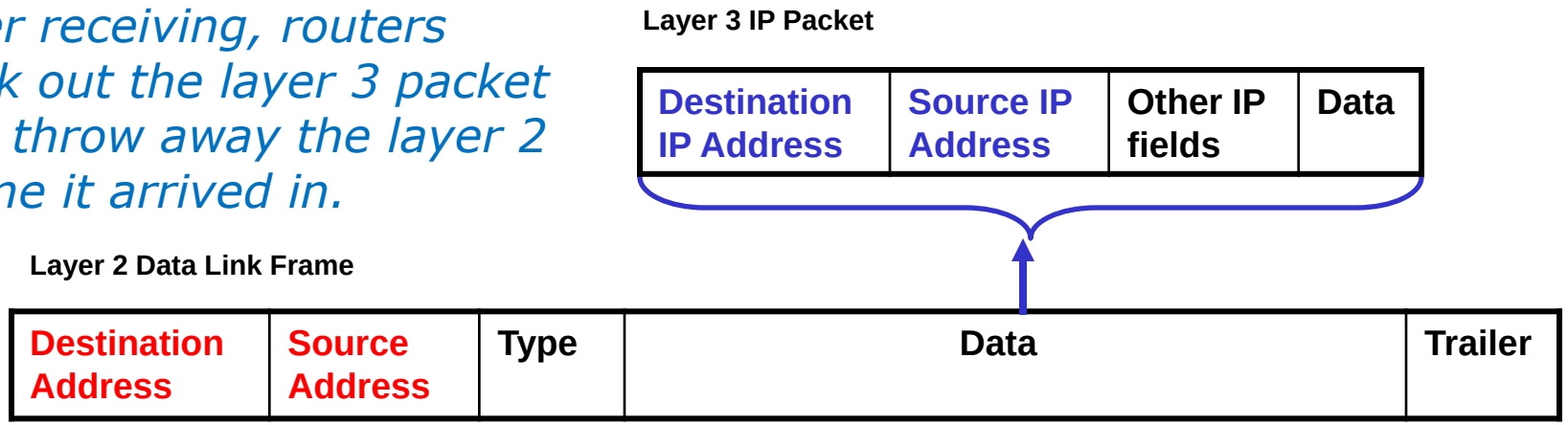
To send, layer 3 packets are encapsulated into layer 2 frames



- Layer 3 packets are encapsulated into Layer 2 frames by the host.
- **Hubs**: Only flood out the Layer 1 bits (repeater)
- **Switches**: Examine only Layer 2 information:
  1. Learn (Source MAC Address)
  2. Forward (Destination MAC Address)
- **Layer 2 frames** can be non-Ethernet frames, such as serial frames:
  - PPP, HDLC, Frame Relay, ATM, ISDN, etc.
  - Point-to-point serial frames (PPP, HDLC) are not multi-access networks and the Destination Address is many times just a layer 2 broadcast address.

Encapsulation

After receiving, routers yank out the layer 3 packet and throw away the layer 2 frame it arrived in.



- **Routers**:
  1. Un-encapsulate Layer 3 packet from Layer 2 frame.
  2. Lookup Layer 3 packet, Destination IP Address, in Routing Table.
  3. Encapsulate Layer 3 packet into new Layer 2 frame and forward out proper (exit) interface.

- **Note**: Destination IP Address and Source IP Address are not in their proper order.

Encapsulation

The layer 3 packet is then placed in a new layer 2 frame appropriate for the next network

These addresses do **not** change!

Layer 3 IP Packet

Destination IP Address	Source IP Address	Other IP fields	Data

These change from host to router, router to router, and router to host.

Layer 2 Data Link Frame

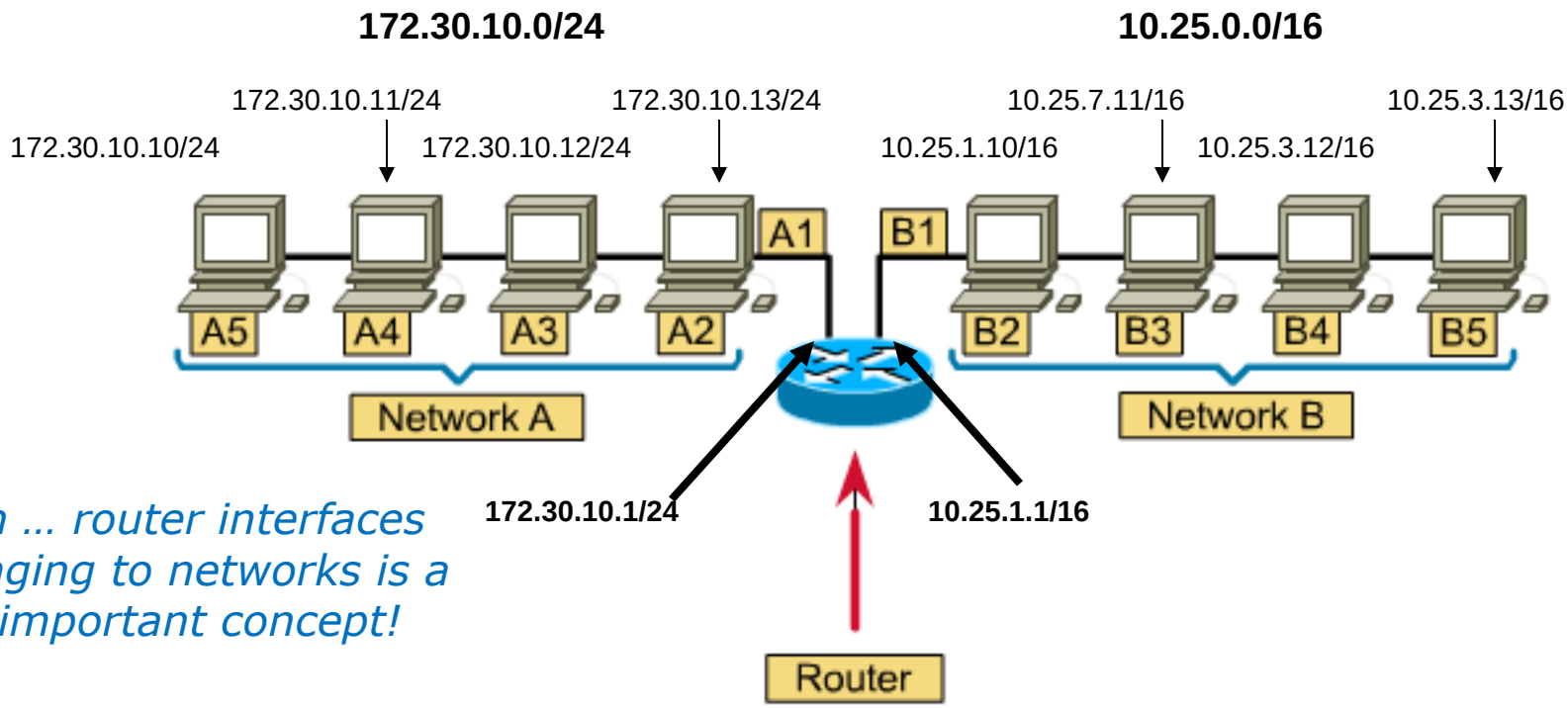
Destination Address	Source Address	Type	Data	Trailer

Next hop Data Link Address of Host or Router's interface

Current Data Link Address of Host or Router's exit interface

- **Note**: The only time Destination and Source IP Addresses change is with NAT/PAT. The only device that is aware of the change is the device doing the NAT, but for all intents and purposes the rule remains the same, IP Addresses do NOT change.

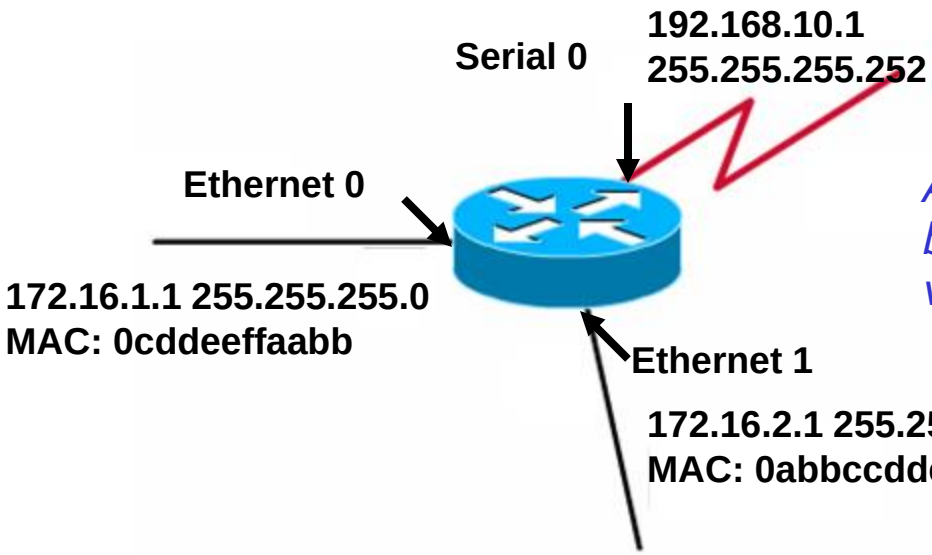
A router interface is a host on that network



Again … router interfaces belonging to networks is a very important concept!

- Since the interface where the router connects to a network is considered to be part of that network.
- Router interfaces have an IP Address and Subnet Mask which makes them a host on the network they are attached.
- Router interfaces must belong to separate networks!

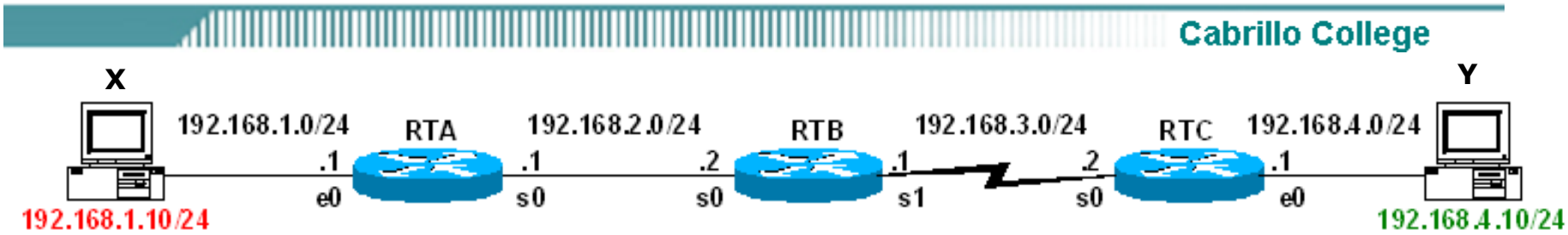
Routers belong to networks



Again … router interfaces belonging to networks is a very important concept!

- Router interfaces must be members of different networks.
- Router interfaces participate in the network like other hosts on that network.
- Ethernet interfaces:
  - Have MAC Addresses
  - ARP Tables
  - Participate in the ARP Request and ARP Reply process like other hosts on that network.

Packet Forwarding



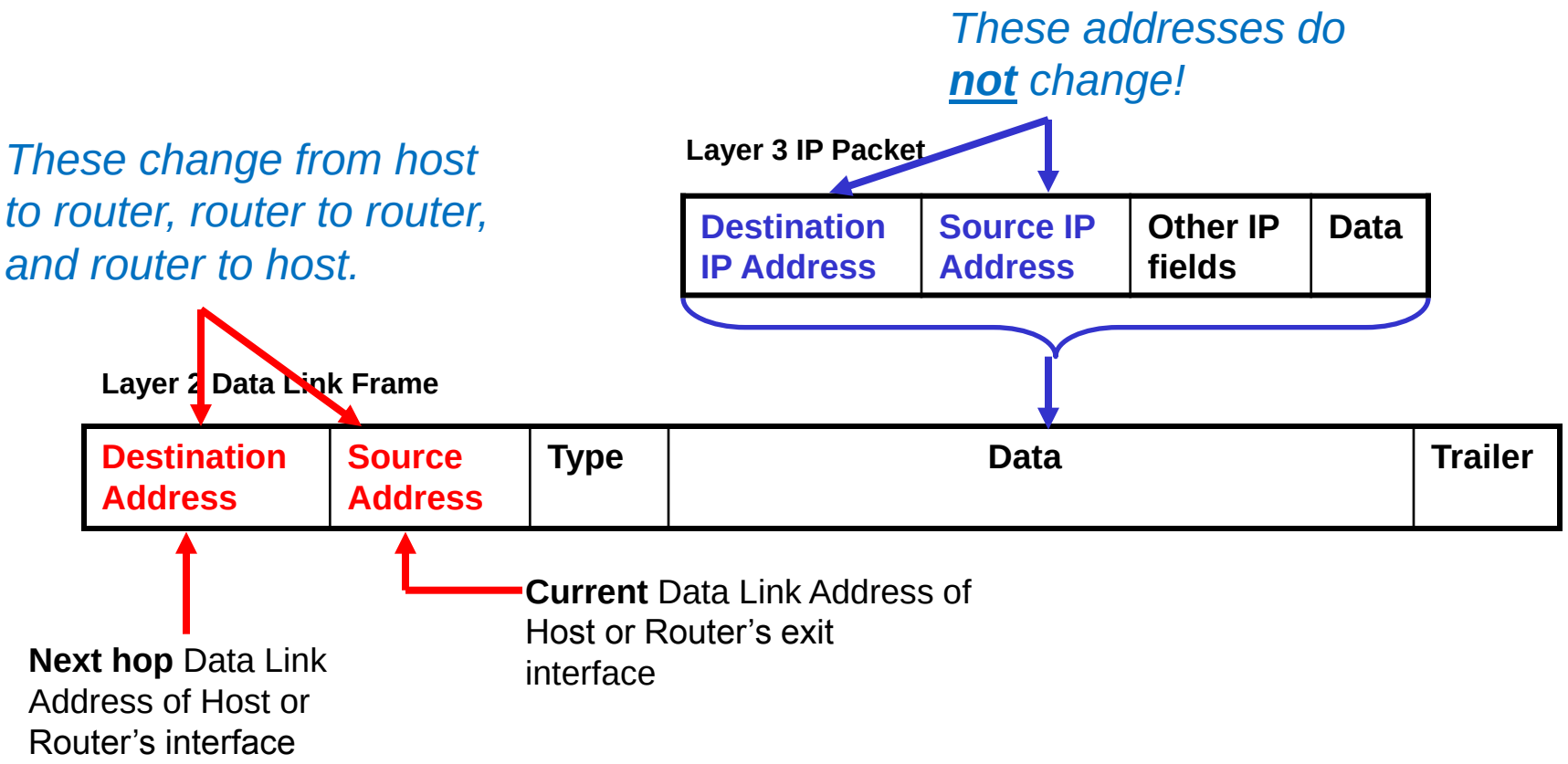
Let's do an example showing how a IP packet travels from network to network to get to its eventual destination.

Packet Forwarding

- Host X has a packet(s) to send to Host Y
- A router generally relays a packet from one data link to another, using two basic functions:
  1. a path determination function - **Routing**
  2. a switching function - **Packet Forwarding**

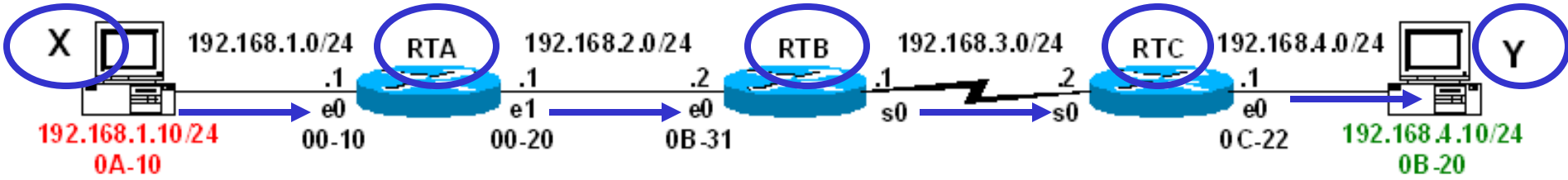
- Let's go through all of the stages these routers use to route and switch this packet.
- **Note**: Data link addresses have been abbreviated.

Remember: Encapsulation



- **Now, let's do an example…**

Remember: Encapsulation



Layer 2 Data Link Frame **Layer 3 IP Packet**

Dest. MAC 0B-20	Source MAC 0C-22	Type 800	Dest. IP 192.168.4.10	Source IP 192.168.1.10	IP fields	Data	Trailer

- This is just a summary.

Linux Routing and Packet Forwarding

- Linux has routing and packet forwarding already built in
- Routing tables are always maintained
- Packet forwarding needs to be enabled

Enable Packet Forwarding
(Red Hat Family)

Temporary

Copy a 1 into **/proc/sys/net/ipv4/ip_forward**

```
[root@elrond ~]# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Or copy a 0 to disable packet forwarding

Enable Packet Forwarding
(Red Hat Family)

Permanent

Edit **/etc/sysctl.conf**

```
[root@elrond ~]# cat /etc/sysctl.conf
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 1

< snipped >

# Controls the maximum number of shared memory segments, in pages
kernel.shmall = 268435456
[root@elrond ~]#
```

enable packet forwarding

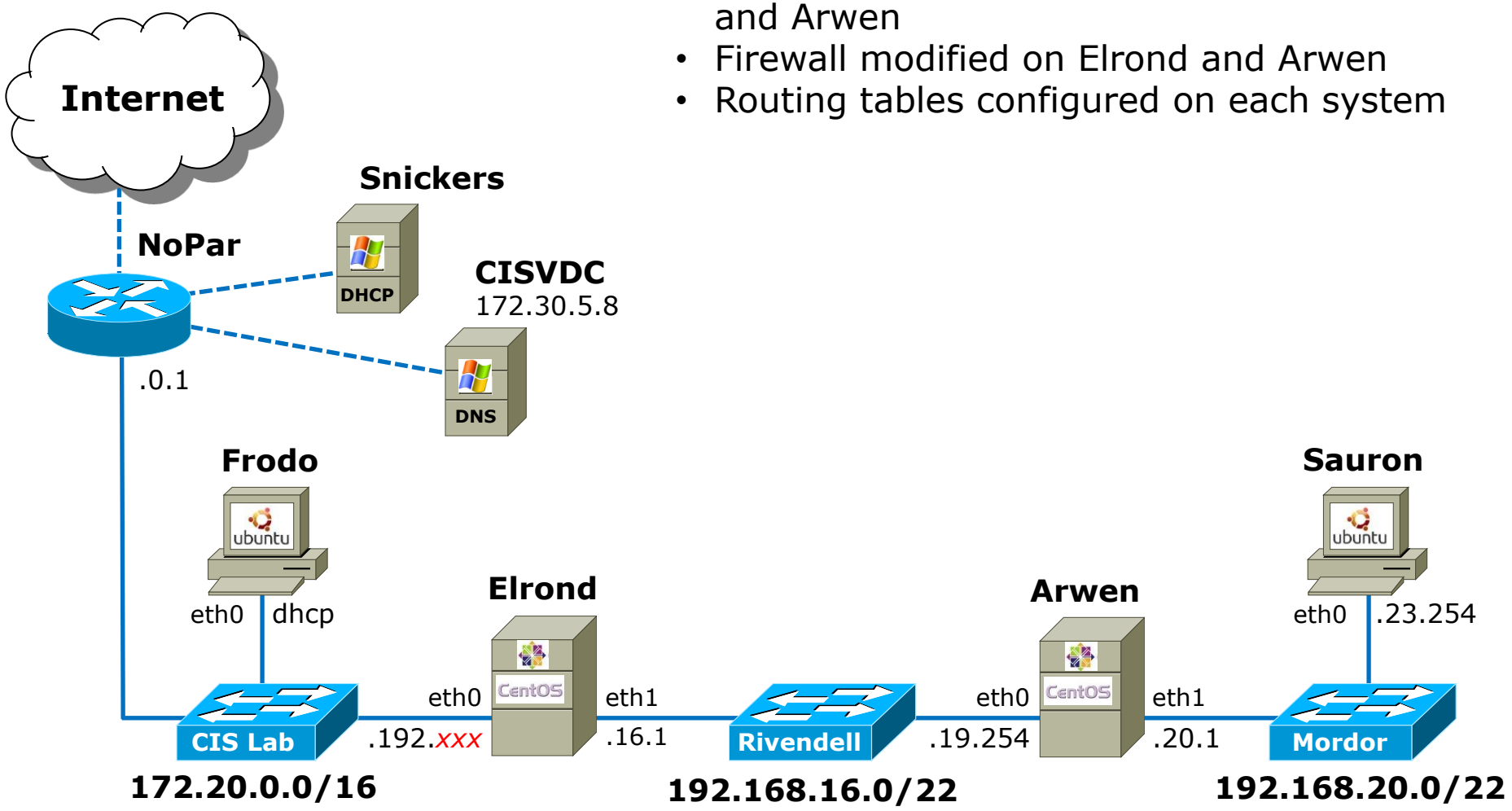
Or set a 0 to disable packet forwarding

Packet Forwarding Summary

- Frame arrives.
- Layer 3 packet yanked (unencapsulated) from frame and the old frame is discarded.
- Routing decision is made using the destination IP address of the layer 3 packet and the routing table.
- A new layer 2 frame is created containing (encapsulating) the layer 3 packet.
- The new frame is sent out the interface determined by the routing decision.

Frodo cannot ping Sauron unless:

- Packet forwarding is enabled on Elrond and Arwen
- Firewall modified on Elrond and Arwen
- Routing tables configured on each system



Enable packet forwarding enabled Elrond and Arwen

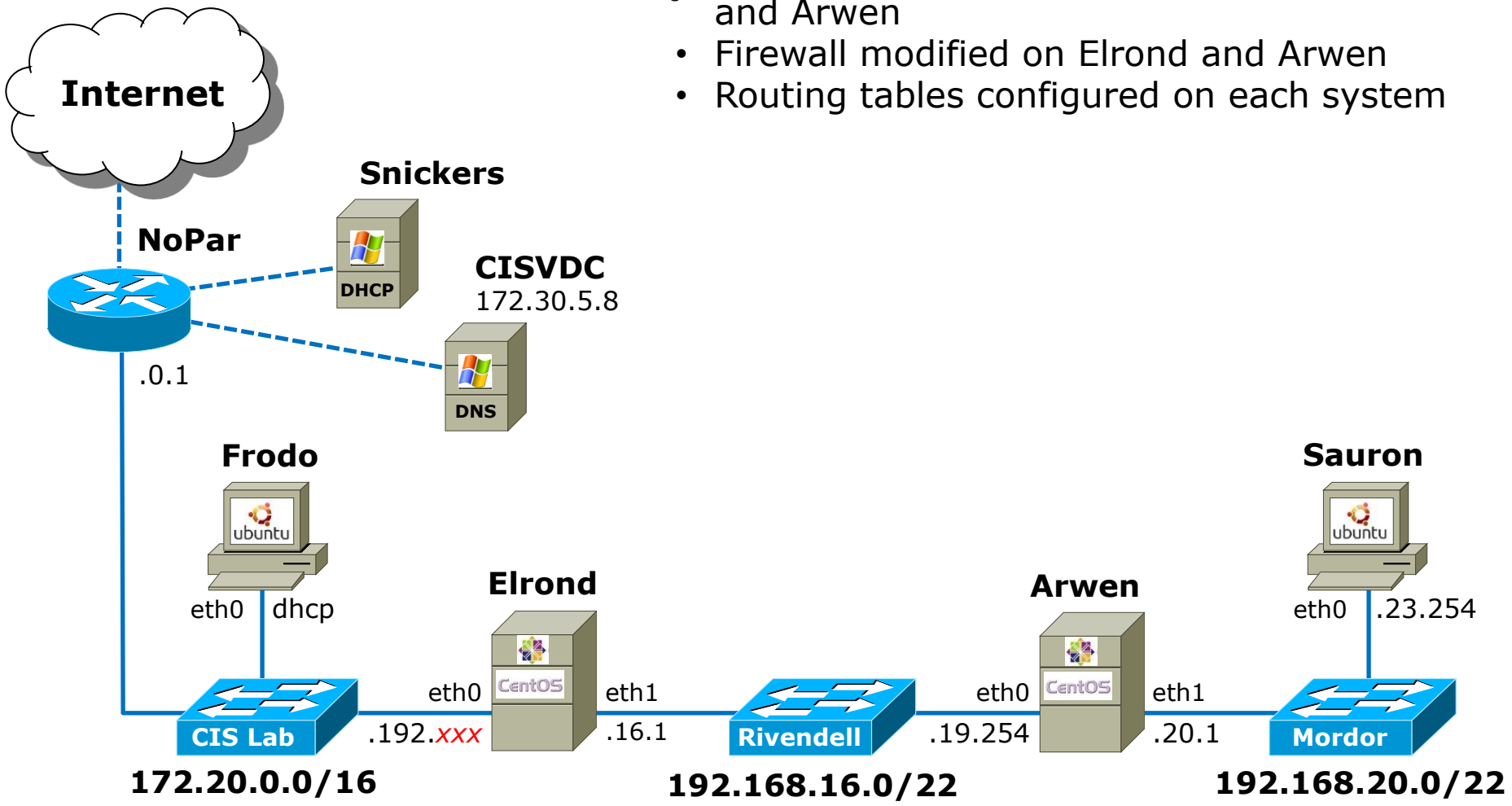
```
[root@p24-elrond ~]# echo 1 > /proc/sys/net/ipv4/ip_forward
[root@p24-elrond ~]# cat /proc/sys/net/ipv4/ip_forward
1
```

```
[root@p24-arwen ~]# echo 1 > /proc/sys/net/ipv4/ip_forward
[root@p24-arwen ~]# cat /proc/sys/net/ipv4/ip_forward
1
```

Done!

Frodo cannot ping Sauron unless:

- ✓ Packet forwarding is enabled on Elrond and Arwen
- Firewall modified on Elrond and Arwen
- Routing tables configured on each system



Modify firewalls to allow forwarding

```
[root@p24-elrond ~]# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@p24-elrond ~]#

[root@p24-elrond ~]# iptables -D FORWARD 1
[root@p24-elrond ~]#
```

Delete the default firewall rule on the FORWARD chain that blocks packet forwarding on Elrond

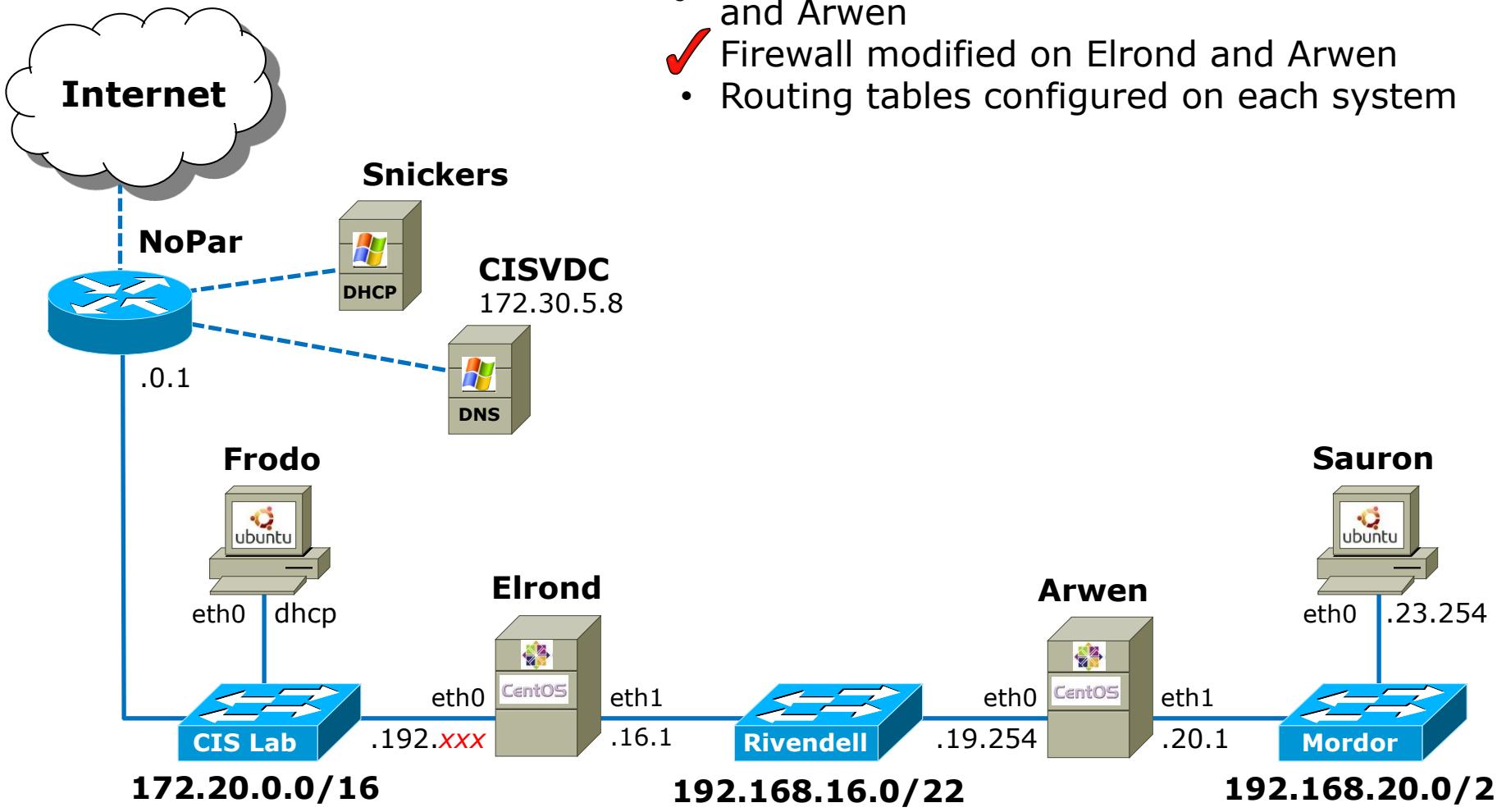
Modify firewalls to allow forwarding

```
[root@p24-arwen ~]# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@p24-arwen ~]#

[root@p24-arwen ~]# iptables -D FORWARD 1
[root@p24-arwen ~]#
```

Delete the default firewall rule on the FORWARD chain that blocks packet forwarding on Arwen

Frodo cannot ping Sauron unless:

- ✓ Packet forwarding is enabled on Elrond and Arwen
- ✓ Firewall modified on Elrond and Arwen
- Routing tables configured on each system

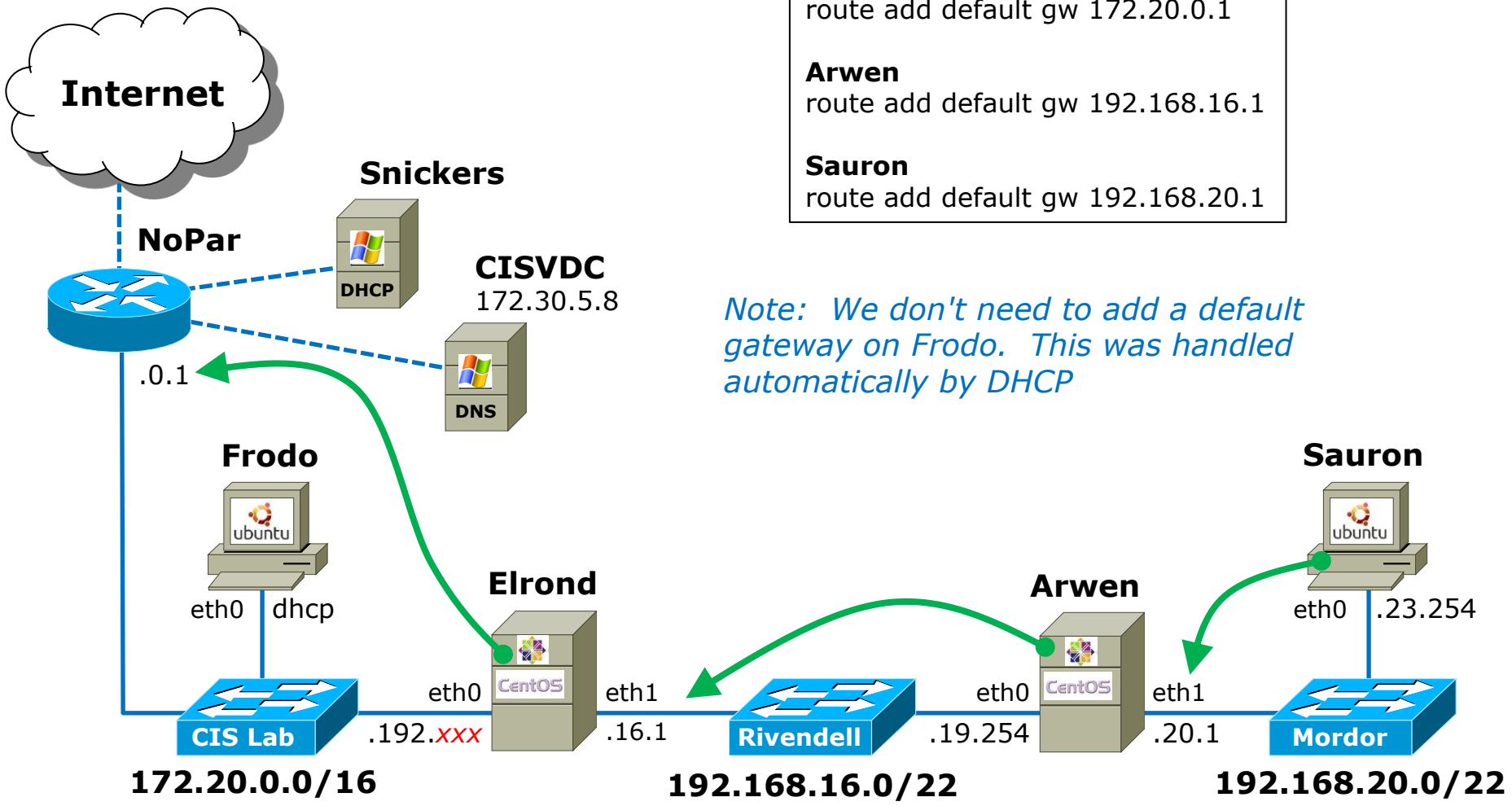


Add default gateways that direct traffic toward the Internet

Elrond
route add default gw 172.20.0.1

Arwen
route add default gw 192.168.16.1

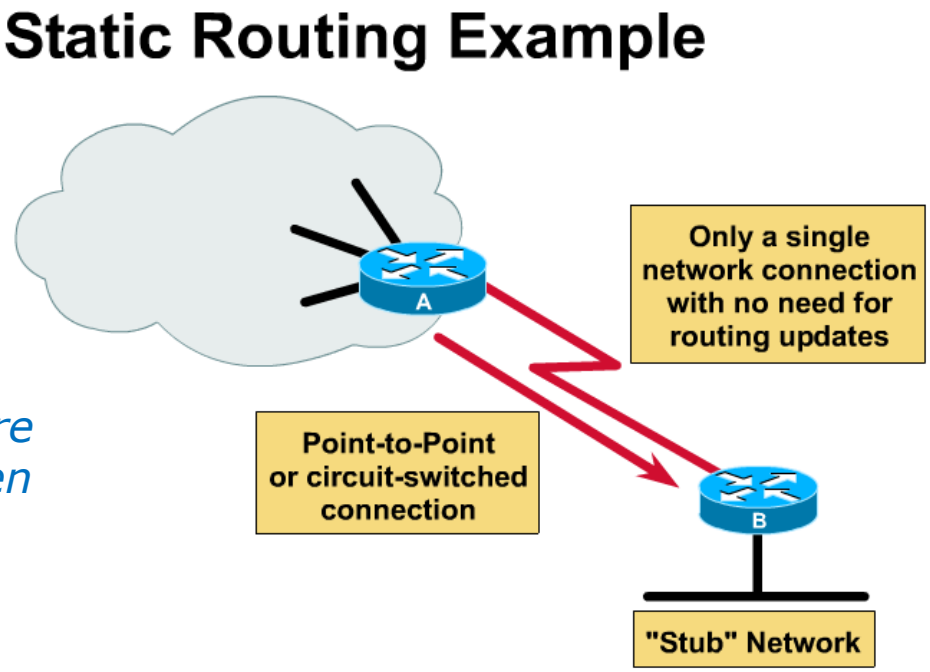
Sauron
route add default gw 192.168.20.1



Note: We don't need to add a default gateway on Frodo. This was handled automatically by DHCP

Static Routes

Common uses for Static Routes

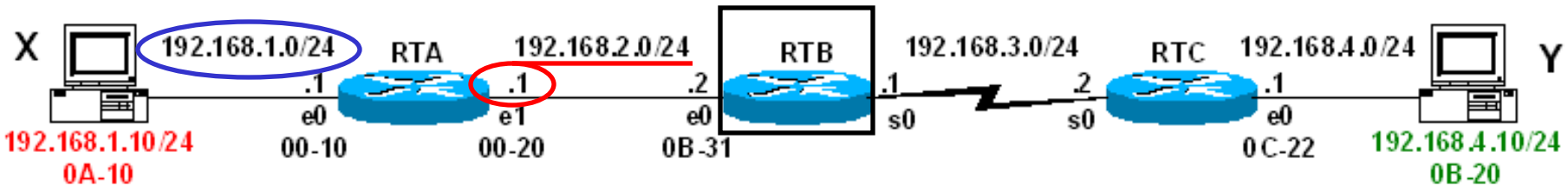


Static routes are very useful even in a dynamic routing world

Static routes in the real-world

- Soon we will learn about **dynamic routing protocols** (RIP, etc.), where routers can learn automatically about networks, without the manual configuration of static routes.
- **Does this mean that static routes are never used in the real-world?**
- **No!** Static routes are used in conjunction with dynamic routing protocols.
- It is common to use a static route where using a dynamic routing protocols would have disadvantages or where it just not needed.

Static Route Examples



- A router must learn about non-directly connected networks.
- To do this with static routes on a **Cisco router:**

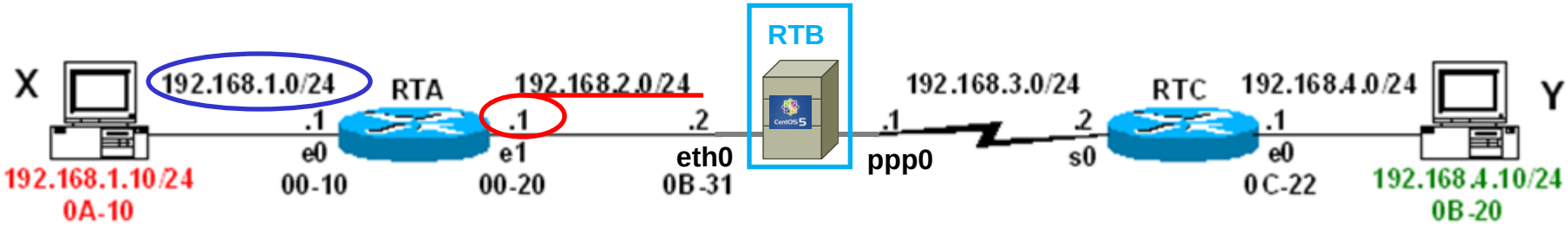
RTB(config)# ip route *network-address mask next-hop-ip-address*

To reach hosts like Host X in the 192.168.1.0/24 network:

RTB(config)# ip route 192.168.1.0 255.255.255.0 192.168.2.1

What would be the static route to reach hosts like Host Y in the 192.168.4.0/24 network?

Static Route Examples



- A router must learn about non-directly connected networks.
- To do this with static routes on a **Linux router** use:

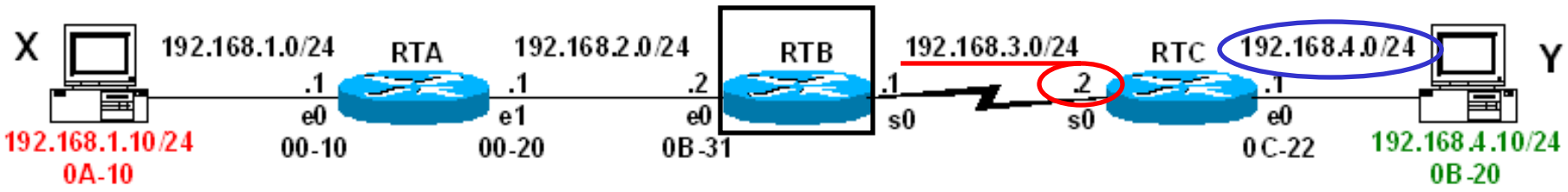
[root@RTB ~#] route add -net *network/prefix* **gw** *next-hop*

To reach hosts like Host X in the 192.168.1.0/24 network:

[root@RTB ~#] route add -net 192.168.1.0/24 gw 192.168.2.1

What would be the static route to reach hosts like Host Y in the 192.168.4.0/24 network?

Static Route Examples

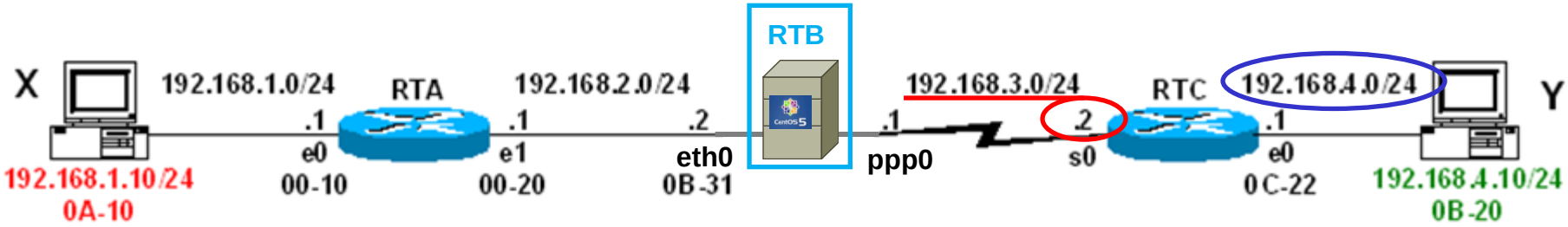


RTB(config)# ip route *network-address mask next-hop-ip-address*

To reach hosts like Host Y in the 192.168.4.0/24 network:

RTB(config)# ip route 192.168.4.0 255.255.255.0 192.168.3.2

Static Route Examples

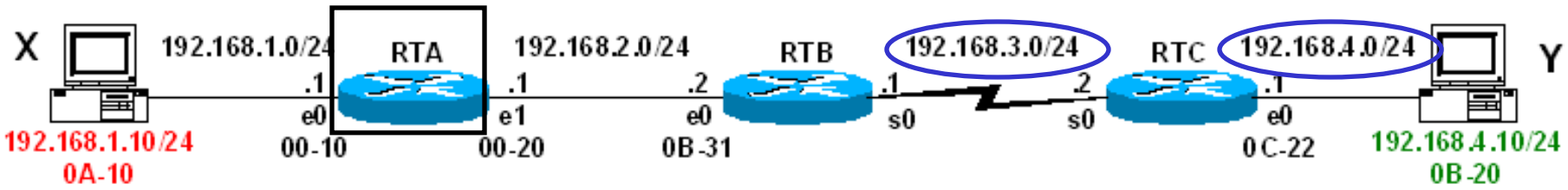


[root@RTB ~#] route add -net *network/prefix* **gw** *next-hop*

To reach hosts like Host Y in the 192.168.4.0/24 network:

[root@RTB ~#] route add -net 192.168.4.0/24 gw 192.168.3.2

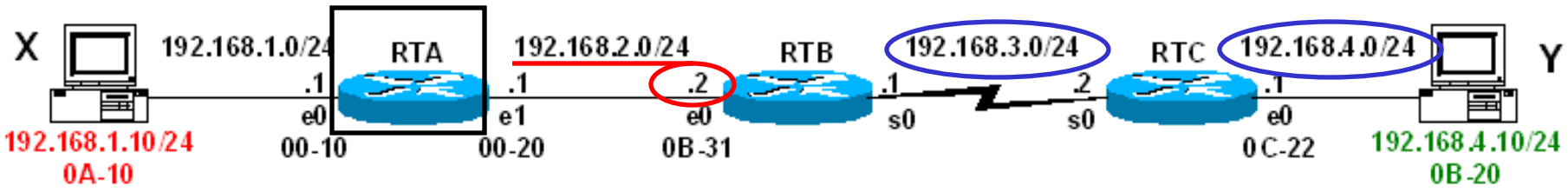
Static Route Examples



What would be the static routes for RTA to reach 192.168.3.0/24 and 192.168.4.0/24 networks?

RTA(config)# ip route *network-address mask next-hop-ip-address*

Static Route Examples



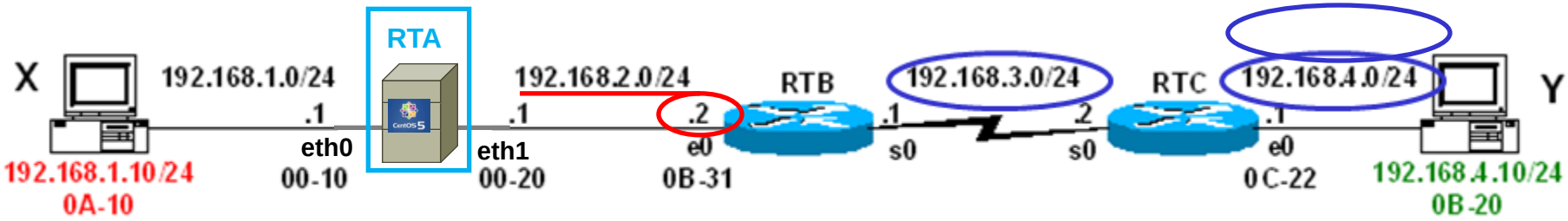
RTB(config)# ip route *network-address mask next-hop-ip-address*

The static routes for RTA to reach 192.168.3.0/24 and 192.168.4.0/24 networks:

RTA(config)# ip route 192.168.3.0 255.255.255.0 192.168.2.2

RTA(config)# ip route 192.168.4.0 255.255.255.0 192.168.2.2

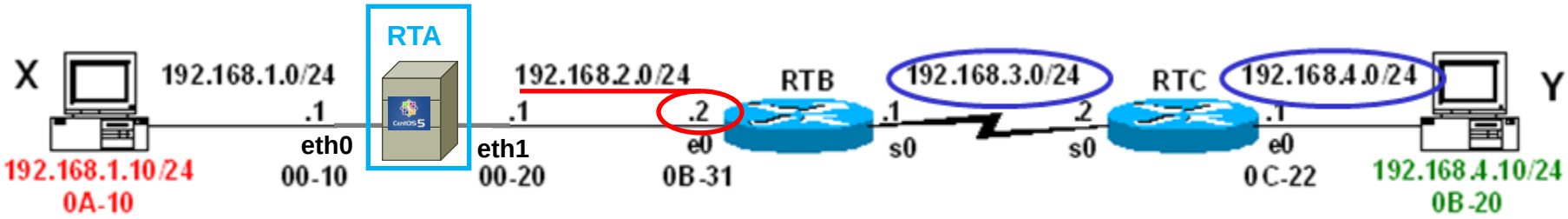
Static Route Examples



What would be the static routes for RTA to reach 192.168.3.0/24 and 192.168.4.0/24 networks?

[root@RTB ~#] route add -net *network/prefix* **gw** *next-hop*

Static Route Examples



[root@RTA ~#] route add -net *network/prefix* **gw *next-hop***

The static routes for RTA to reach 192.168.3.0/24 and 192.168.4.0/24 networks:

[root@RTA ~#] route add -net 192.168.3.0/24 gw 192.168.2.2
[root@RTA ~#] route add -net 192.168.4.0/24 gw 192.168.2.2

Setting Static Routes
(Red Hat Family)

Temporary

- **route add -net** *network* **netmask** *mask* **gw** *next-hop*
- **route del -net** *network* **netmask** *mask* **gw** *next-hop*

alternate syntax

[root@elrond ~]# **route add -net 192.168.3.0 netmask 255.255.255.0 gw 192.168.2.123**
[root@elrond ~]# **route del -net 192.168.3.0 netmask 255.255.255.0 gw 192.168.2.123**

- **route add -net** *network/prefix* **gw** *next-hop*
- **route del -net** *network/prefix* **gw** *next-hop*

[root@elrond ~]# **route add -net 192.168.3.0/24 gw 192.168.2.123**
[root@elrond ~]# **route del -net 192.168.3.0/24 gw 192.168.2.123**

Permanent

- Edit **/etc/sysconfig/network-scripts/route-eth***

```
[root@elrond ~]# cat /etc/sysconfig/network-scripts/route-eth1
192.168.3.0/24 via 192.168.2.123
[root@elrond ~]# service network restart
```

Add static routes that direct traffic toward the Rivendell and Mordor

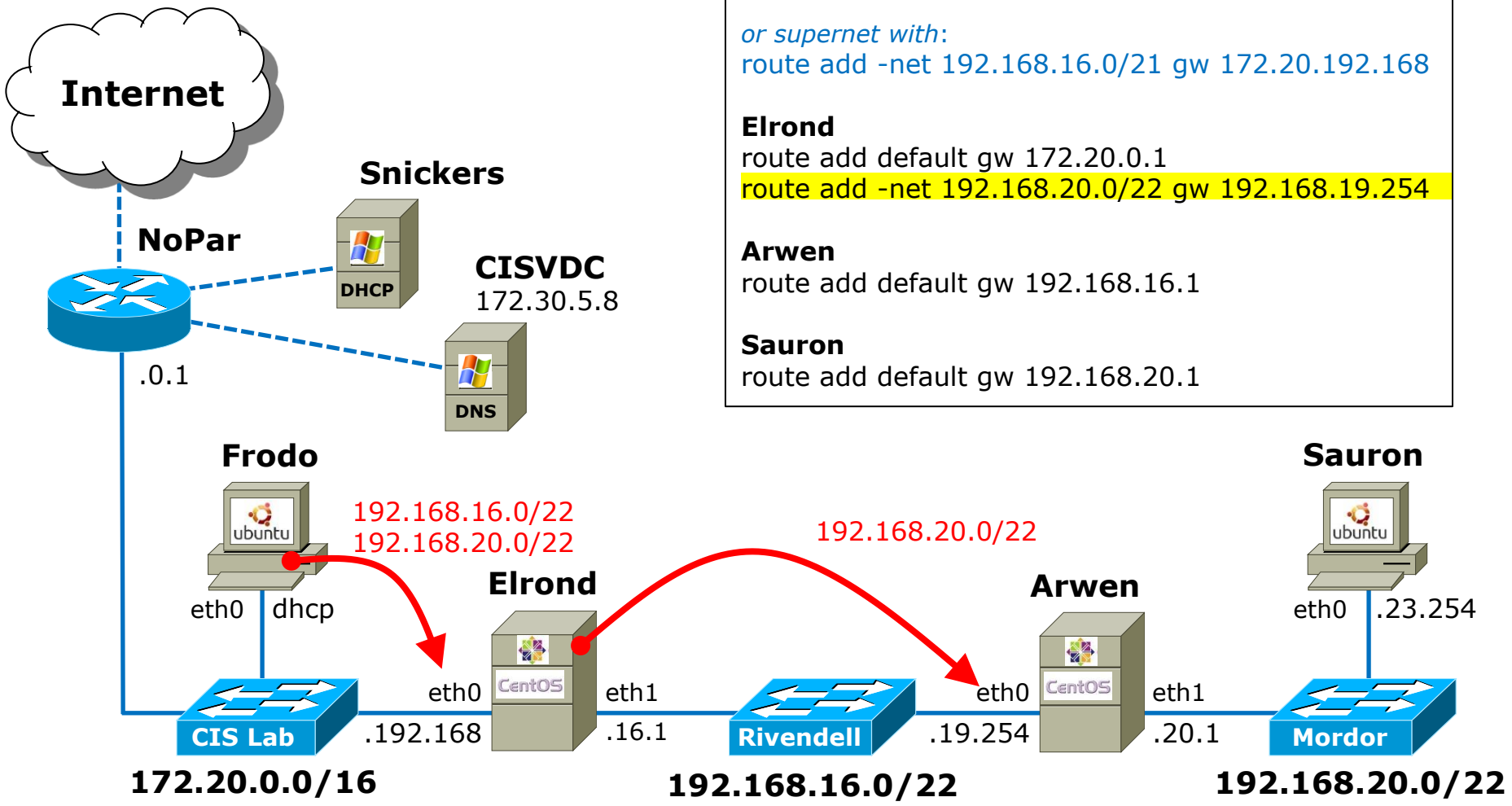
Frodo
route add -net 192.168.16.0/22 gw 172.20.192.168
route add -net 192.168.20.0/22 gw 172.20.192.168

or supernet with:
route add -net 192.168.16.0/21 gw 172.20.192.168

Elrond
route add default gw 172.20.0.1
route add -net 192.168.20.0/22 gw 192.168.19.254

Arwen
route add default gw 192.168.16.1

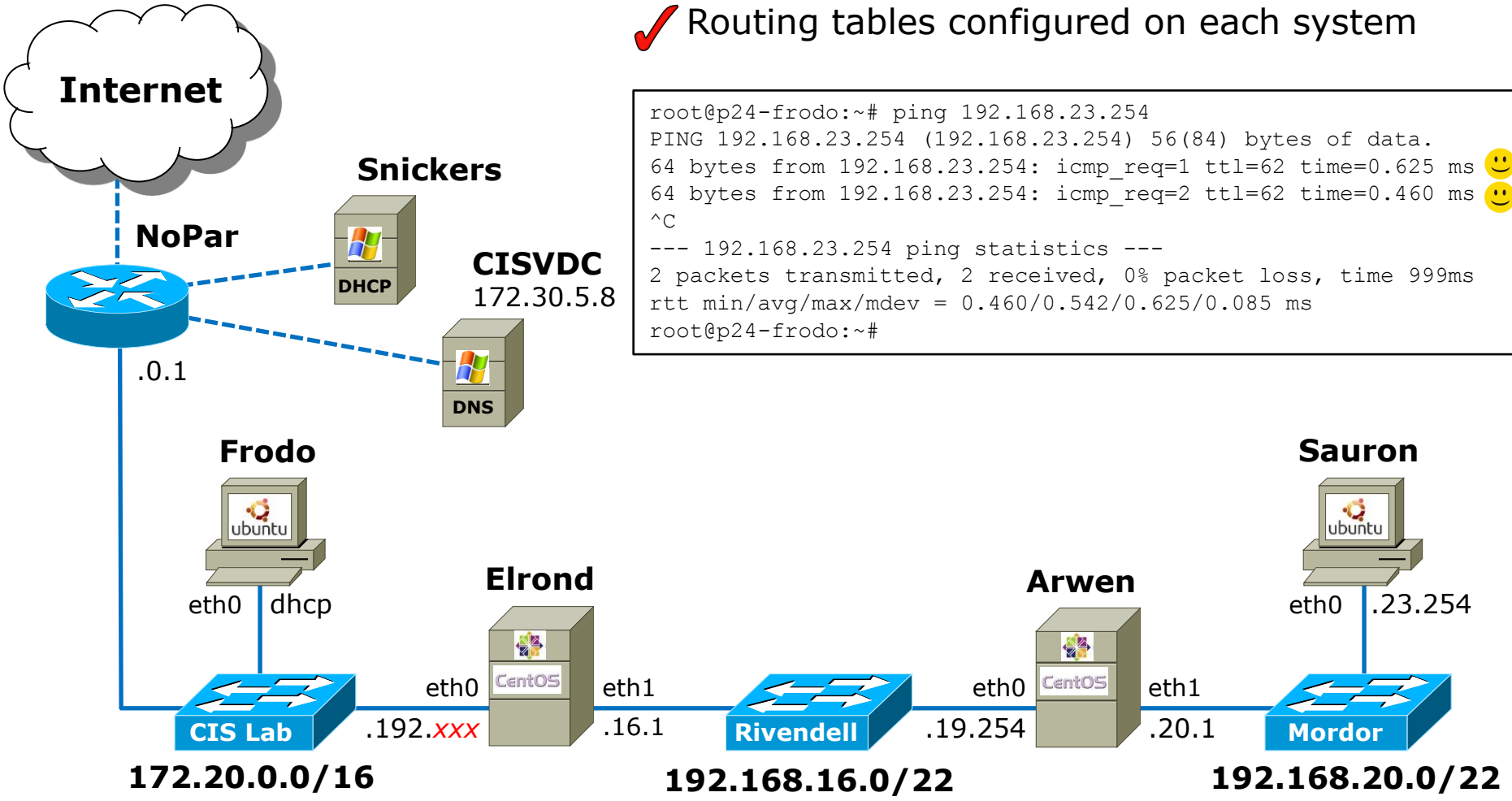
Sauron
route add default gw 192.168.20.1



Frodo cannot ping Sauron unless:

- ✓ Packet forwarding is enabled on Elrond and Arwen
- ✓ Firewall modified on Elrond and Arwen
- ✓ Routing tables configured on each system

```
root@p24-frodo:~# ping 192.168.23.254
PING 192.168.23.254 (192.168.23.254) 56(84) bytes of data.
64 bytes from 192.168.23.254: icmp_req=1 ttl=62 time=0.625 ms
64 bytes from 192.168.23.254: icmp_req=2 ttl=62 time=0.460 ms
^C
--- 192.168.23.254 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.460/0.542/0.625/0.085 ms
root@p24-frodo:~#
```



Routing Table

Routing Table Overview

- Directly connected networks are automatically added to the routing table.
- Static routes can be added using the route command.
- Default gateways can be added using the route command.
- Dynamic routing services that use routing protocols like RIP and OSPF can add additional routes to the routing table.

The Routing Table

Routing Table

shows IP addresses instead of names (faster)

```
[root@p24-elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.20.0    192.168.19.254  255.255.252.0   UG    0      0        0 eth1
192.168.16.0    0.0.0.0         255.255.252.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1003   0        0 eth1
172.20.0.0      0.0.0.0         255.255.0.0     U     0      0        0 eth0
0.0.0.0         172.20.0.1      0.0.0.0         UG    0      0        0 eth0
[root@p24-elrond ~]#
```

Destination shows the networks that a route exists for.
The 0.0.0.0 network is used for the default route.

The Routing Table

Routing Table

```
[root@p24-elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.20.0    192.168.19.254  255.255.252.0   UG    0      0        0 eth1
192.168.16.0    0.0.0.0         255.255.252.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1003   0        0 eth1
172.20.0.0      0.0.0.0         255.255.0.0     U     0      0        0 eth0
0.0.0.0         172.20.0.1      0.0.0.0         UG    0      0        0 eth0
[root@p24-elrond ~]#
```

*Gateway specifies the next-hop router or uses 0.0.0.0 for local **directly-connected** interfaces*

The Routing Table

Routing Table

```
[root@p24-elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.20.0    192.168.19.254  255.255.252.0   UG    0      0        0 eth1
192.168.16.0    0.0.0.0         255.255.252.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1003   0        0 eth1
172.20.0.0      0.0.0.0         255.255.0.0     U     0      0        0 eth0
0.0.0.0         172.20.0.1      0.0.0.0         UG    0      0        0 eth0
[root@p24-elrond ~]#
```

Genmask is the mask applied to incoming destination IP addresses to determine if a route exists. These are sorted by longest (best match) to shortest prefix.

The Routing Table

Routing Table

```
[root@p24-elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.20.0    192.168.19.254  255.255.252.0   UG    0      0        0 eth1
192.168.16.0    0.0.0.0         255.255.252.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1003   0        0 eth1
172.20.0.0      0.0.0.0         255.255.0.0     U     0      0        0 eth0
0.0.0.0         172.20.0.1      0.0.0.0         UG    0      0        0 eth0
[root@p24-elrond ~]#
```

Note the genmask of 0.0.0.0 is used for the default route. Applying this mask to any address will always result in a match.

The Routing Table

Routing Table

```
[root@p24-elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.20.0    192.168.19.254  255.255.252.0   UG    0      0        0 eth1
192.168.16.0    0.0.0.0         255.255.252.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1003   0        0 eth1
172.20.0.0      0.0.0.0         255.255.0.0     U     0      0        0 eth0
0.0.0.0         172.20.0.1      0.0.0.0         UG    0      0        0 eth0
[root@p24-elrond ~]#
```

Possible flags include:

- *U (route is up)*
- *H (target is a host)*
- *G (gateway)*

The Routing Table

Routing Table

```
[root@p24-elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.20.0    192.168.19.254  255.255.252.0   UG    0      0        0 eth1
192.168.16.0    0.0.0.0         255.255.252.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1003   0        0 eth1
172.20.0.0      0.0.0.0         255.255.0.0     U     0      0        0 eth0
0.0.0.0         172.20.0.1      0.0.0.0         UG    0      0        0 eth0
[root@p24-elrond ~]#
```

Metric:
The distance to the target (usually counted in hops). It is not used by recent kernels, but may be needed by routing daemons.

The Routing Table

Routing Table

```
[root@p24-elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.20.0    192.168.19.254  255.255.252.0   UG    0      0        0 eth1
192.168.16.0    0.0.0.0         255.255.252.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1003   0        0 eth1
172.20.0.0      0.0.0.0         255.255.0.0     U     0      0        0 eth0
0.0.0.0         172.20.0.1      0.0.0.0         UG    0      0        0 eth0
[root@p24-elrond ~]#
```

Ref:
Number of references to this route.
(when showing cache)

The Routing Table

Routing Table

```
[root@p24-elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.20.0    192.168.19.254  255.255.252.0   UG    0      0        0 eth1
192.168.16.0    0.0.0.0         255.255.252.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1003   0        0 eth1
172.20.0.0      0.0.0.0         255.255.0.0     U     0      0        0 eth0
0.0.0.0         172.20.0.1      0.0.0.0         UG    0      0        0 eth0
[root@p24-elrond ~]#
```

Use:
Count of lookups for the route
(when showing cache)

The Routing Table

Routing Table

```
[root@p24-elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.20.0    192.168.19.254  255.255.252.0   UG    0      0        0 eth1
192.168.16.0    0.0.0.0         255.255.252.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1003   0        0 eth1
172.20.0.0      0.0.0.0         255.255.0.0     U     0      0        0 eth0
0.0.0.0         172.20.0.1      0.0.0.0         UG    0      0        0 eth0
[root@p24-elrond ~]#
```

Iface:
Interface to which packets for this route will be sent.

The Routing Table
Supernetting

Routing Table

```
root@p24-frodo:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.20.0.1      0.0.0.0         UG    0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth0
172.20.0.0      0.0.0.0         255.255.0.0     U     1      0        0 eth0
192.168.16.0    172.20.192.168  255.255.252.0   UG    0      0        0 eth0
192.168.20.0    172.20.192.168  255.255.252.0   UG    0      0        0 eth0
```

*Note: these two routes could be replaced with a single route for **192.168.16.0/21**. This is super-netting (the reverse of sub-netting)*

Netmask: 255.255.248.0 = 21
Netmask: 255.255.252.0 = 22

The Routing Algorithm

(How the decision is made)

Routing Algorithm

The purpose of the Routing Algorithm is to get the packet to its destination network.

- Compute the network of the destination IP address
- Does the destination network match that on a local interface?
  If so, send it out that interface
- Does the destination network match one or more listed in the routing table?
  If so, send it using the best match (largest genmask) route
- Is there a default route listed in the routing table?
  If so, use that gateway
  Otherwise, drop the packet - "network is unreachable"

The Routing Algorithm

Compute the network

The network number is obtained by applying the genmask to the incoming IP destination address.

Example: 192.168.23.254 with genmask 255.255.252.0 is 192.168.20.0

```
Address:    192.168.23.254        11000000.10101000.000101 11.11111110
Netmask:    255.255.252.0 = 22    11111111.11111111.111111 00.00000000
Network:    192.168.20.0/22       11000000.10101000.000101 00.00000000

[root@p24-elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.20.0    192.168.19.254  255.255.252.0   UG    0      0        0 eth1
192.168.16.0    0.0.0.0         255.255.252.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1003   0        0 eth1
172.20.0.0      0.0.0.0         255.255.0.0     U     0      0        0 eth0
0.0.0.0         172.20.0.1      0.0.0.0         UG    0      0        0 eth0
[root@p24-elrond ~]#
```

On Elrond, an incoming packet with a destination address of 192.168.23.254 will be sent out eth1 to the next-hop router at 192.168.19.254

route command -n option

```
[root@p24-elrond ~]# route
```

show route table with names

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.20.0    arwen           255.255.252.0   UG    0      0        0 eth1
192.168.16.0    *               255.255.252.0   U     0      0        0 eth1
link-local      *               255.255.0.0     U     1002   0        0 eth0
link-local      *               255.255.0.0     U     1003   0        0 eth1
172.20.0.0      *               255.255.0.0     U     0      0        0 eth0
default         lab-router      0.0.0.0         UG    0      0        0 eth0
```

```
[root@p24-elrond ~]# route -n
```

show route table with IP addresses

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.20.0    192.168.19.254  255.255.252.0   UG    0      0        0 eth1
192.168.16.0    0.0.0.0         255.255.252.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1003   0        0 eth1
172.20.0.0      0.0.0.0         255.255.0.0     U     0      0        0 eth0
0.0.0.0         172.20.0.1      0.0.0.0         UG    0      0        0 eth0
[root@p24-elrond ~]#
```

route command for viewing cache

```
[root@p24-elrond ~]# route -C
Kernel IP routing cache
Source          Destination     Gateway         Flags Metric Ref    Use Iface
172.20.4.104    172.20.255.255  172.20.255.255  ibl   0      0       13 lo
172.20.4.105    172.20.255.255  172.20.255.255  ibl   0      0       32 lo
frodo           172.20.192.168  172.20.192.168  il    0      0       53 lo
monitor.cislab. 172.20.255.255  172.20.255.255  ibl   0      0        4 lo
172.20.4.105    255.255.255.255 255.255.255.255 ibl   0      0        0 lo
172.20.192.168  frodo           frodo                 0      1        0 eth0
globe.cislab.ne 172.20.255.255  172.20.255.255  ibl   0      0        1 lo
172.20.192.168  frodo           frodo                 0      1        0 eth0
globe.cislab.ne 172.20.255.255  172.20.255.255  ibl   0      0        1 lo
sauron          frodo           frodo                 0      0        7 eth0
172.20.192.168  cisvdc.cislab.n lab-router            0      0        6 eth0
frodo           sauron          arwen           i     0      0        7 eth1
172.20.4.58     172.20.255.255  172.20.255.255  ibl   0      0        0 lo
[root@p24-elrond ~]#
```

show route table cache with names

route command for viewing cache

```
[root@p24-elrond ~]# route -Cn
Kernel IP routing cache
Source          Destination     Gateway         Flags Metric Ref    Use Iface
172.20.192.168  172.30.5.8      172.20.0.1              0      0        9 eth0
172.20.4.104    172.20.255.255  172.20.255.255  ibl     0      0       14 lo
172.20.4.105    172.20.255.255  172.20.255.255  ibl     0      0       33 lo
172.30.5.8      172.20.192.168  172.20.192.168  l       0      0        9 lo
172.20.4.153    172.20.192.168  172.20.192.168  il      0      0       65 lo
172.20.0.7      172.20.255.255  172.20.255.255  ibl     0      0        4 lo
172.20.4.105    255.255.255.255 255.255.255.255 ibl     0      0        0 lo
172.20.192.168  172.20.4.153    172.20.4.153            0      1        0 eth0
172.20.90.10    172.20.255.255  172.20.255.255  ibl     0      0        1 lo
192.168.23.254  172.20.4.153    172.20.4.153            0      0        7 eth0
172.20.4.16     172.20.255.255  172.20.255.255  ibl     0      0        0 lo
172.20.192.168  172.30.5.8      172.20.0.1              0      0        9 eth0
172.20.4.153    192.168.23.254  192.168.19.254  i       0      0        7 eth1
172.20.4.58     172.20.255.255  172.20.255.255  ibl     0      0        0 lo
[root@p24-elrond ~]#
```

show route table cache with IP addresses

route command
flushing the cache

Flush the route cache

```
[root@p24-elrond ~]# ip route flush cache
[root@p24-elrond ~]# route -Cn
Kernel IP routing cache
Source          Destination     Gateway         Flags Metric Ref    Use Iface
172.20.4.153    172.20.192.168  172.20.192.168  il    0      0        9 lo
[root@p24-elrond ~]#
```

*Note: Use **route -CF** on Red Hat 9*

Exercise

1. Explore the routing table and cache on Celebrian
   - route
   - route -n
   - route -C
   - route -Cn

2. Flush the route cache
   - ip route flush cache
   - route -Cn

3. Identify the directly connected, static and default routes.

/etc/hosts

/etc/hosts

ping frodo vs **ping 172.20.4.153**
ssh frodo vs **ssh 172.20.4.153**
scp myfile frodo: vs **ssh myfile 172.20.4.153:**

- Before the Domain Name System (DNS) arrived on the scene, text files with IP/hostname associations were maintained for name resolution.
- Hostnames are much easier to remember than IP addresses!
- The file **/etc/hosts** files is still available to quickly add local hostnames for systems not resolved by your DNS servers.

/etc/hosts

```
[root@p24-elrond ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
172.20.4.153 frodo
172.20.0.1  lab-router nopar
192.168.19.254 arwen
192.168.23.254 sauron
[root@p24-elrond ~]#
```

```
[root@p24-elrond ~]# ping -c2 nopar
PING lab-router (172.20.0.1) 56(84) bytes of data.
64 bytes from lab-router (172.20.0.1): icmp_seq=1 ttl=255 time=4.16 ms
64 bytes from lab-router (172.20.0.1): icmp_seq=2 ttl=255 time=0.468 ms

--- lab-router ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.468/2.318/4.168/1.850 ms
[root@p24-elrond ~]#
```

Use /etc/hosts to refer to hosts by name rather than IP address

Trouble shooting

ICMP Errors

Host Unreachable

```
root@frodo:~# ping 192.168.2.128
PING 192.168.2.128 (192.168.2.128) 56(84) bytes of data.
From 172.30.4.125 icmp_seq=3 Destination Host Unreachable
From 172.30.4.125 icmp_seq=4 Destination Host Unreachable
From 172.30.4.125 icmp_seq=5 Destination Host Unreachable
^C
--- 192.168.2.128 ping statistics ---
9 packets transmitted, 0 received, +3 errors, 100% packet loss, time 8019ms
, pipe 3
root@frodo:~#
```

When the packet arrives at the destination network there is no active host to receive the packet. The host is offline, does not exist, or is not cabled to the network. The ARP request for this host's MAC address is failing.

ICMP Errors

TTL exceeded

```
root@frodo:~# ping 192.168.5.200
PING 192.168.5.200 (192.168.5.200) 56(84) bytes of data.
From 192.168.2.123 icmp_seq=1 Time to live exceeded
From 192.168.2.123 icmp_seq=2 Time to live exceeded
From 192.168.2.123 icmp_seq=3 Time to live exceeded
From 192.168.2.123 icmp_seq=4 Time to live exceeded
From 192.168.2.123 icmp_seq=5 Time to live exceeded
From 192.168.2.123 icmp_seq=6 Time to live exceeded
^C
--- 192.168.5.200 ping statistics ---
6 packets transmitted, 0 received, +6 errors, 100% packet loss, time 5030ms

root@frodo:~#
```

One router is forwarding the packet to the next-hop router. The next-hop router has no specific route for this packet but does have a default route back to the previous router! Loops back and forth the until TTL count is 0 and then the packet is dropped.

ICMP Errors

Network Unreachable

```
[root@legolas ~]# ping 172.30.4.1
connect: Network is unreachable
[root@legolas ~]#
```

There is no matching route in the route table.
To fix, add a default gateway or a static route

ICMP Errors

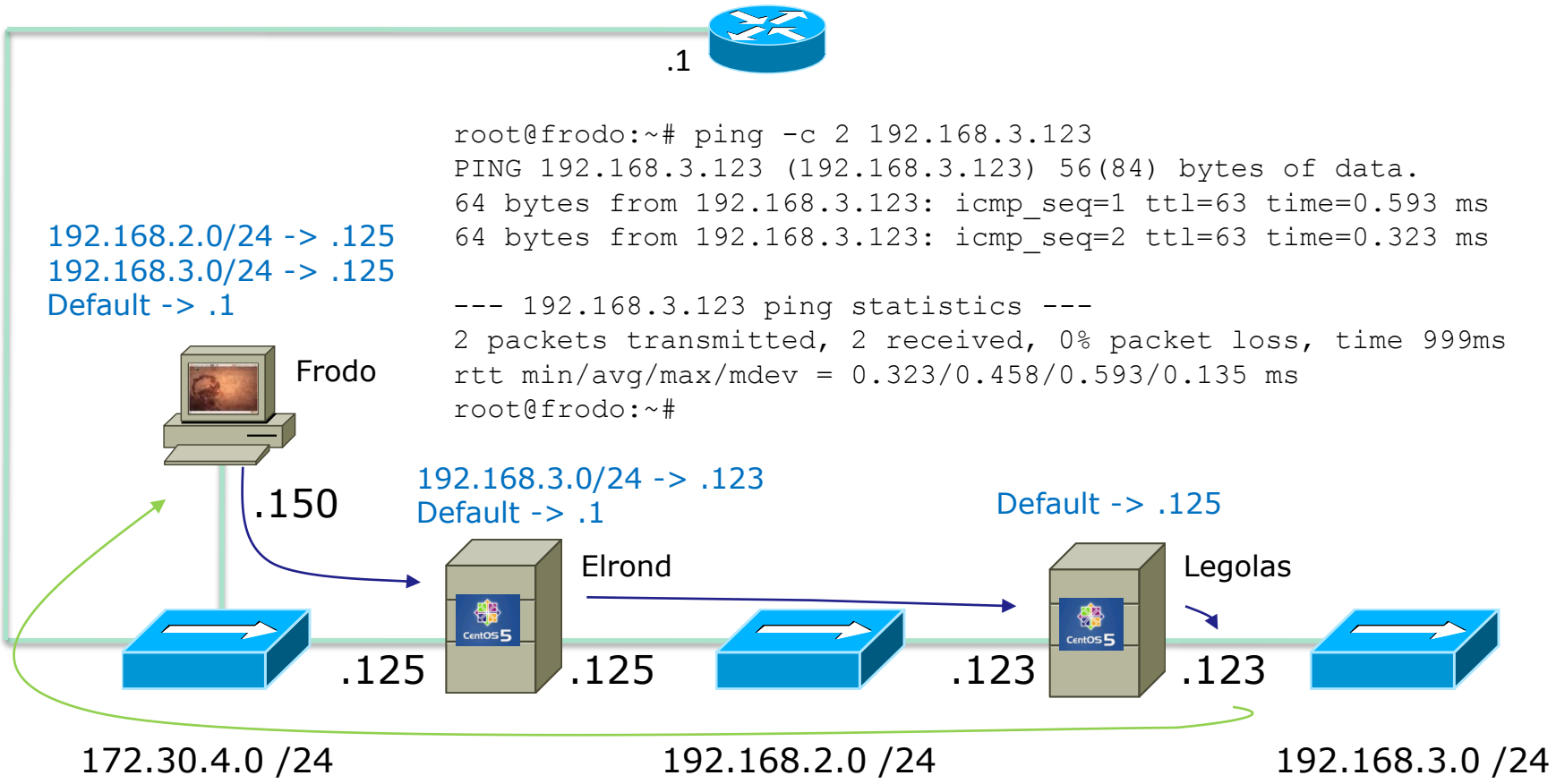
Nothing

```
[root@legolas ~]# ping 10.240.1.2
PING 10.240.1.2 (10.240.1.2) 56(84) bytes of data.

--- 10.240.1.2 ping statistics ---
8 packets transmitted, 0 received, 100% packet loss, time 7011ms

[root@legolas ~]#
```

No response! The ping is being sent out on a route where there is no route back!

ICMP Redirect

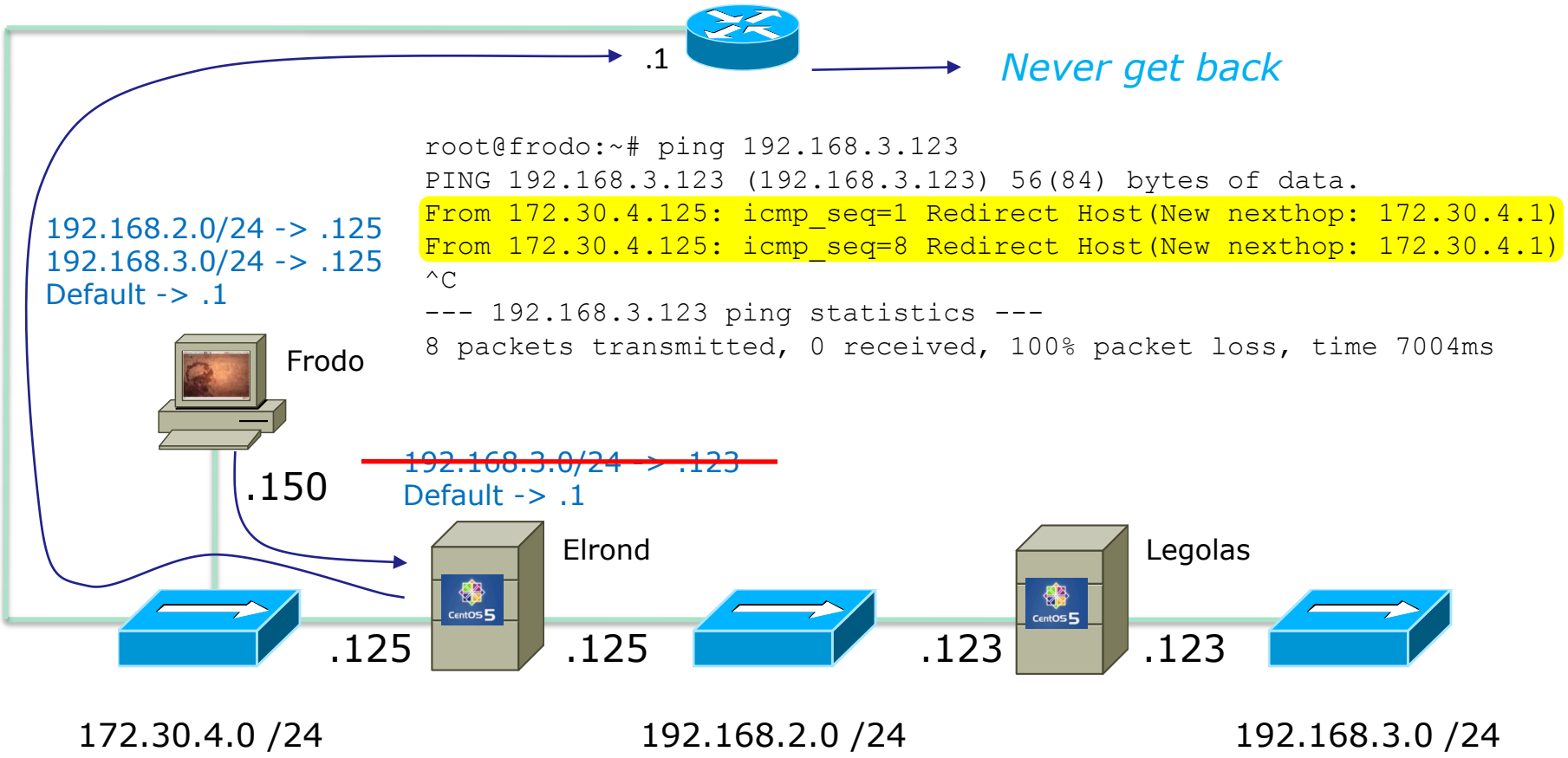
Routers will update each others caches when they discover an inefficient route



What happens if we forgot to add the static route on Elrond?

ICMP Redirect

Routers will update each others caches when they discover an inefficient route



Elrond tells Frodo there is a shorter route

Lab

Configuring a network (What you will be doing in Lab 3)

Overall steps:

- Cable systems
- Assign valid IP addresses to all hosts and routers
- Enable IP forwarding on all routers
- Open firewalls
- Configure the routing tables of all hosts and routers

*Tip: Use **ip route flush cache** when correcting any entries in the routing table*

Wrap

New commands, tools and services:

- service
- chkconfig
- dhclient
- ip
- route
- service
- chkconfig
- iptables

New Files and Directories (Red Hat):

- /etc/sysconfig/network
- /etc/sysconfig/network-scripts/ifcfg-eth*
- /etc/sysconfig/network-scripts/route-eth*
- /proc/sys/net/ipv4/ip_forward
- /etc/sysctl.conf

New Files and Directories (Ubuntu):

- /etc/hostname
- /etc/network/interfaces
- /etc/init.d/networking restart

VMware:

Next Class

Assignment: Check Calendar Page on web site to see what is due next week.

Quiz questions for next class:

- What command enables packet forwarding?
- What command deletes the first rule on the **iptables** FORWARD chain?
- What command would add a static route to the 192.168.20.0/22 network via a gateway at 192.168.19.254?

Backup

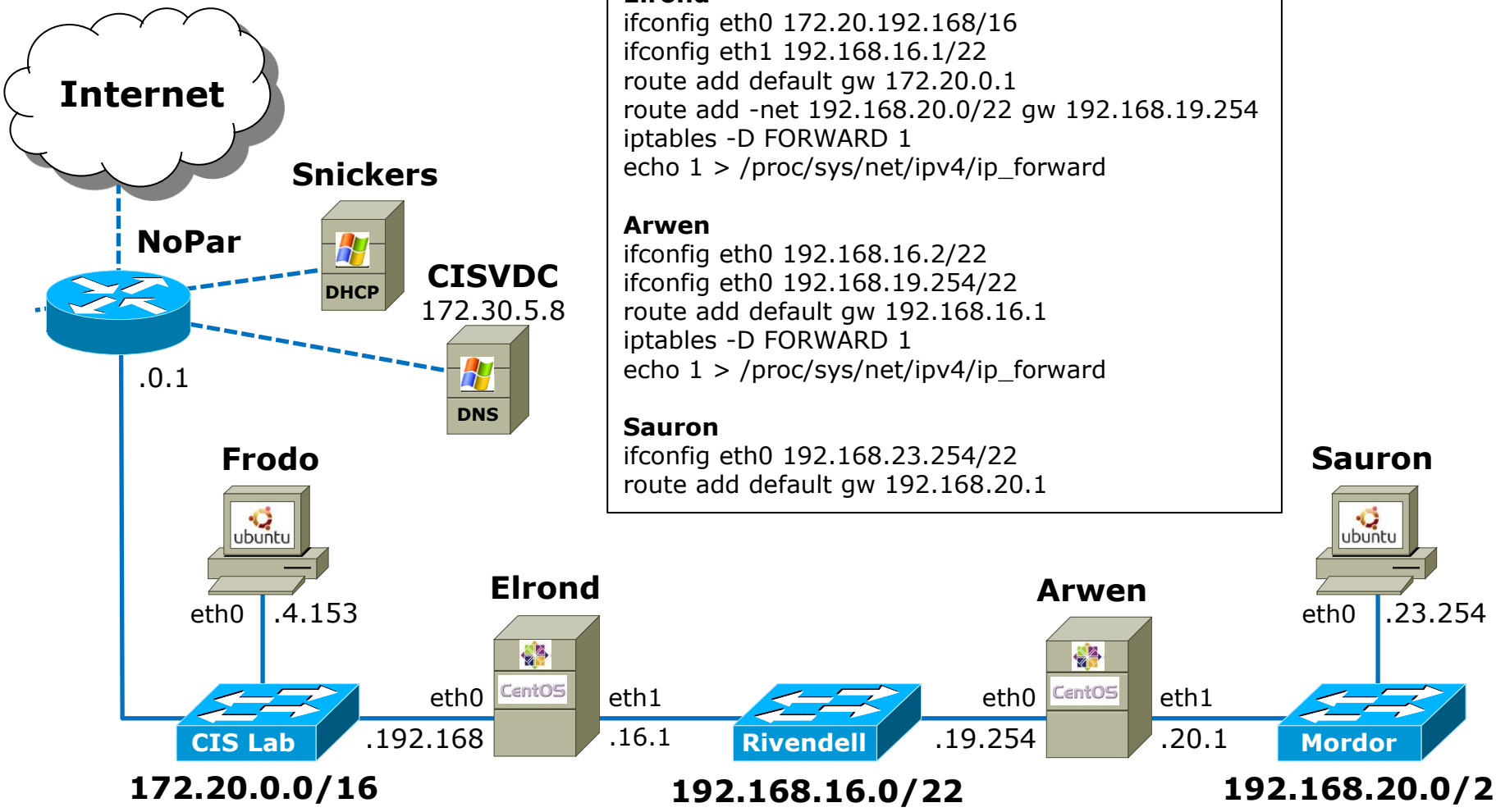
CRIB SHEET

Frodo
route add -net 192.168.16.0/21 gw 172.20.192.168

Elrond
ifconfig eth0 172.20.192.168/16
ifconfig eth1 192.168.16.1/22
route add default gw 172.20.0.1
route add -net 192.168.20.0/22 gw 192.168.19.254
iptables -D FORWARD 1
echo 1 > /proc/sys/net/ipv4/ip_forward

Arwen
ifconfig eth0 192.168.16.2/22
ifconfig eth0 192.168.19.254/22
route add default gw 192.168.16.1
iptables -D FORWARD 1
echo 1 > /proc/sys/net/ipv4/ip_forward

Sauron
ifconfig eth0 192.168.23.254/22
route add default gw 192.168.20.1



Permanent Network Configuration Review +

Configuring Network Settings

Different ways to configure network settings

1. **GUI tools**
   - Permanent,
   - Different for each distribution

2. The **ifconfig** and **route** commands
   - Temporary (till next restart)

3. Editing **configuration files** and restarting the network service
   - Permanent
   - Some variations between distributions
   - Requires network service to be restarted

Configuring Permanent Network Settings
(Red Hat Family)

Setting	File
IP address and subnet mask	/etc/sysconfig/network-scripts/ifcfg-eth*
Default gateway	/etc/sysconfig/network
DNS server(s)	/etc/resolv.conf
Hostname	/etc/sysconfig/network
Name/IP pairing	/etc/hosts

/etc/sysconfig/network-scripts/ifcfg-eth0

By the way - tab completes are wonderful

Managing System Services (daemons)
(Red Hat Family)

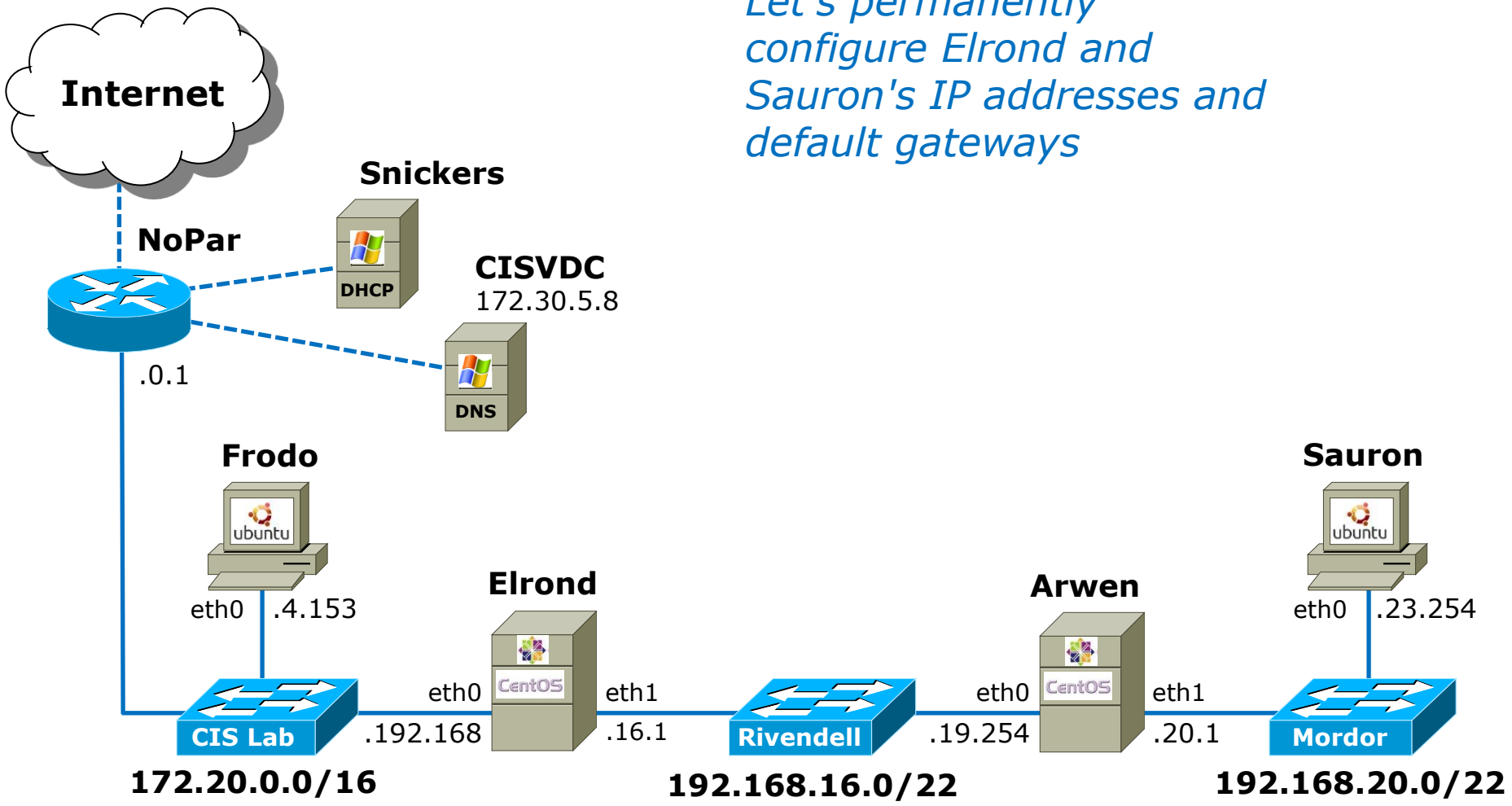
Manually starting and stopping

- **service network** <stop|start|restart|status>

System startup configuration

- **chkconfig network** <on|off>

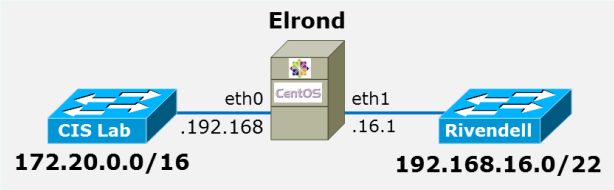
Let's permanently configure Elrond and Sauron's IP addresses and default gateways



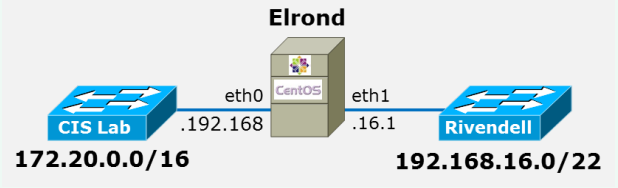
Set Permanent Static IP Address and Subnet Mask

(Red Hat Family)

```
[root@p24-elrond ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
BOOTPROTO="static"
NM_CONTROLLED="no"
ONBOOT="yes"
TYPE="Ethernet"
IPADDR="172.20.192.168"
NETMASK="255.255.0.0"
```

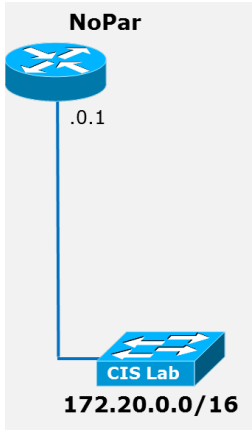



```
[root@p24-elrond ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE="eth1"
BOOTPROTO="static"
NM_CONTROLLED="no"
ONBOOT="yes"
TYPE="Ethernet"
IPADDR="192.168.16.1"
NETMASK="255.255.252.0"
```




```
[root@p24-elrond ~]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down interface eth1:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:                                [  OK  ]
Bringing up interface eth1:                                [  OK  ]
```

Configuring permanent default gateway
(Red Hat Family)




```
[root@p24-elrond ~]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=p24-elrond.rivendell
GATEWAY=172.20.0.1
```

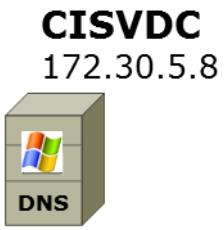
```
[root@p24-elrond ~]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down interface eth1:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:                                [  OK  ]
Bringing up interface eth1:                                [  OK  ]
```

```
[root@p24-elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.16.0    0.0.0.0         255.255.252.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1003   0        0 eth1
172.20.0.0      0.0.0.0         255.255.0.0     U     0      0        0 eth0
0.0.0.0         172.20.0.1      0.0.0.0         UG    0      0        0 eth0
```

Configuring DNS name servers

```
[root@p24-elrond ~]# cat /etc/resolv.conf
search cislab.net
nameserver 172.30.5.8
```

This will be appended to host names when trying to resolve them




```
[root@p24-elrond ~]# ping nopar -c2
PING nopar.cislab.net (172.20.0.1) 56(84) bytes of data.
64 bytes from nopar.cislab.net (172.20.0.1): icmp_seq=1 ttl=255 time=0.428 ms
64 bytes from nopar.cislab.net (172.20.0.1): icmp_seq=2 ttl=255 time=0.506 ms

--- nopar.cislab.net ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.428/0.467/0.506/0.039 ms
[root@p24-elrond ~]#
```

Configuring the hostname
(Red Hat Family)

Permanent: Step 1 - edit **/etc/sysconfig/network**

```
[root@elrond ~]# cat /etc/sysconfig/network
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=homer.localdomain
GATEWAY=172.30.4.1
[root@elrond ~]# init 6
```

change hostname

Restart

```
CentOS release 5.2 (Final)
Kernel 2.6.18-92.1.22.el5 on an i686

homer login: root
Password:
Last login: Fri Feb 20 01:23:44 from 172.30.4.103
[root@homer ~]# _
```

new hostname

Configuring the hostname
(Red Hat Family)

Permanent: Step 2 - edit **/etc/hosts**

```
[root@homer ~]# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       elrond.localdomain elrond localhost.localdomain localhost
::1             localhost6.localdomain6 localhost6
[root@homer ~]#
```

Be sure and update /etc/hosts with new hostname if necessary

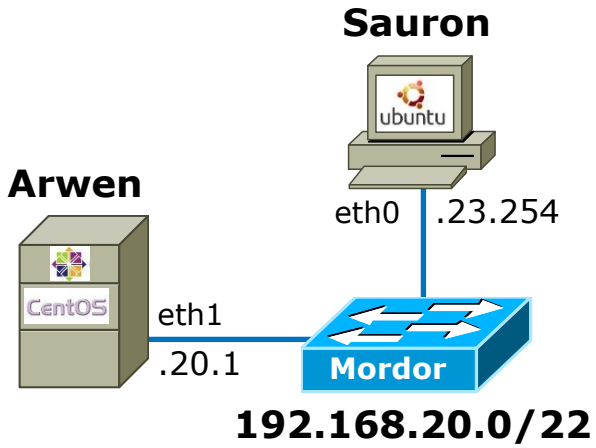
```
[root@homer ~]# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       homer.localdomain homer localhost.localdomain localhost
::1             localhost6.localdomain6 localhost6
[root@homer ~]#
```

Debian/Ubuntu Network Config

Debian/Ubuntu
Permanent Network Configuration

Configuring a static IP, mask and default gateway

```
root@p24-sauron:~# cat /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.23.254
netmask 255.255.252.0

gateway 192.168.20.1

root@p24-sauron:~#
```



Debian/Ubuntu
Permanent Network Configuration

Configuring DNS Settings

```
root@sun:~# cat /etc/resolv.conf
search cabrillo.edu
nameserver 10.240.1.2
root@sun:~#
```

Same as the Re[illegible]amily

[illegible] string is optional. [illegible] be appended to names being resolved if the name by itself does not resolve to an IP address

Update later for latest Ubuntu changes

Debian/Ubuntu
Permanent NIC Configuration

Restarting network service
(so new settings take effect)

/etc/init.d/networking restart

Debian/Ubuntu is a little different than the Red Hat family

```
root@p24-sauron:~# /etc/init.d/networking restart
 * Running /etc/init.d/networking restart is deprecated because it may not enable again some interfaces
 * Reconfiguring network interfaces...                                    ssh stop/waiting
ssh start/running, process 2410
                                                                        [ OK ]
root@p24-sauron:~#
```

Debian/Ubuntu
System Hostname

Changing system hostname

```
root@jin:~# cat /etc/hostname
jin
root@jin:~#

root@jin:~# vi /etc/hostname
root@jin:~# cat /etc/hostname
sun
root@jin:~# init 6

< system restart snipped >

root@sun:~#
root@sun:~#
```

Change hostname to sun

Restart system

Prompt string changes after reboot

Important, be sure to update /etc/hosts with new hostname!

Debian/Ubuntu
Permanent NIC Configuration

Static Routes

```
root@sun:~# cat /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 172.30.4.222
netmask 255.255.255.0
broadcast 172.30.4.255
network 172.30.4.0

gateway 172.30.4.1

up route add -net 192.168.2.0/24 gw 172.30.4.107
up route add -net 192.168.30.0/24 gw 172.30.4.107

root@sun:~#
```

Use up and down to configure what will happen when interface is brought up or down