Lesson Module Status

- Slides – draft
- Properties - done
- Flashcards - na
- 1st minute quiz – done
- Web Calendar summary – done
- Web book pages – done
- Commands – done
- Howtos – done
- Skills pacing - na
- Lab – done
- Depot (VMs) – done
- Special:

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/

Teach & Confer is a live interactive classroom to meet with your students.

STUDENT LOG IN

View Teach & Confer Archives

www.confer.org
dial-in: 888-886-3951
passcode: 439080


Joe A.

Joe P.


Chris B.


Chuck


Rich

Chris H.

Lieven

Julio


Jack

Edwin

Kay

Aaron


Joe B.

Junious


Drew


Josh

Ryan

Patrick


John

Robert


Casady


Edgar

Brynden

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

First Minute Quiz

Please take out a blank piece of paper, switch off your monitor, close your books, put away your notes and answer these questions:

- What Wireshark display filter would only show ARP or ICMP protocol packets?
- With an IP address of 172.30.4.100 and a netmask of 255.255.0.0, what is the broadcast address?
- What does the C flag mean when viewing ARP cache entries with arp -n?

Online CCC Confer users can email the answers to risimms@cabrillo.edu

Routing and Subnetting

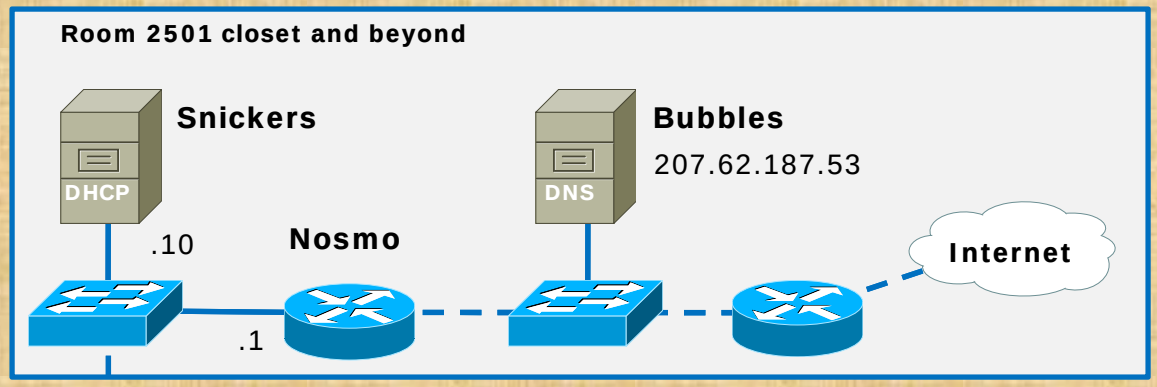
Related Course Objectives	Agenda
• Configure appropriate IP addresses, network and subnet masks, and broadcast addresses based on the size and number of network segments required. • Connect multiple network segments together using Linux servers as routers and configuring the appropriate routing tables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

Questions on previous material

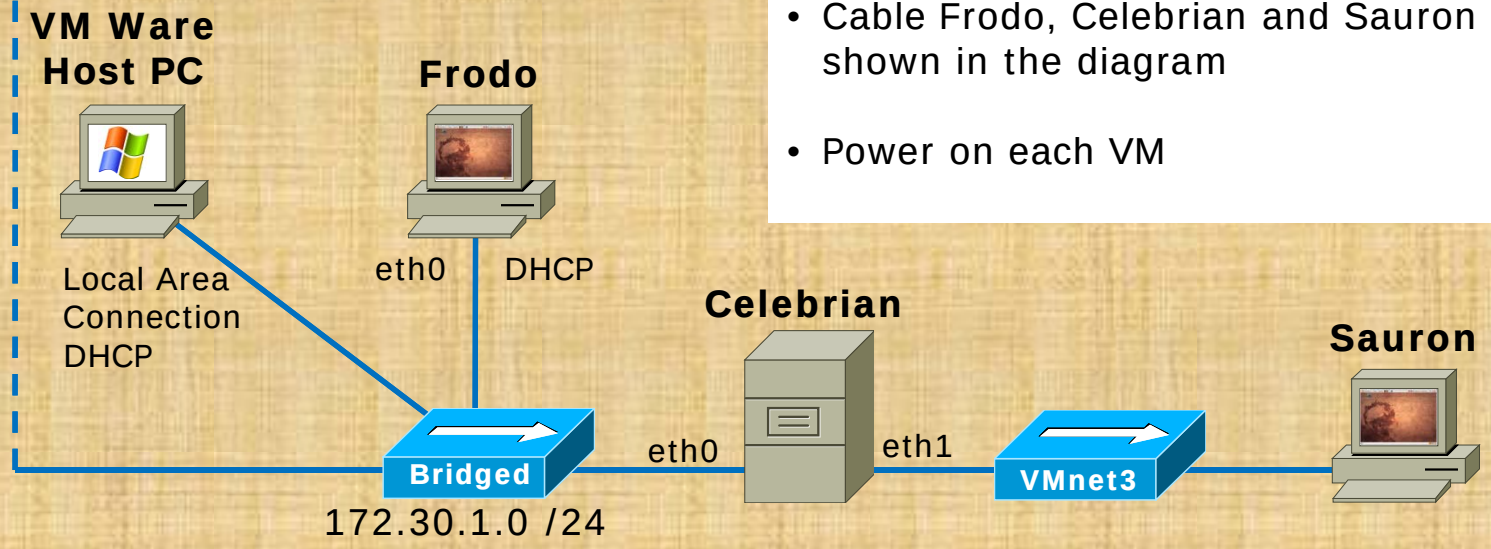
Questions?

- Previous lesson material
- Lab assignment
- How this class works

Class Exercise – Fire up the VMs for tonight



- Revert VMs to their snapshots
- Cable Frodo, Celebrian and Sauron as shown in the diagram
- Power on each VM



Housekeeping

- Roll Call
- Using CCC Confer improvements

1. Share multiple apps instead of entire desktop (to eliminate using ½ of screen for control panel
2. Repeat all questions for online listeners
    - Please help by reminding me to do this when I forget
3. In class students, please help me monitor chat window for ?'s from online students

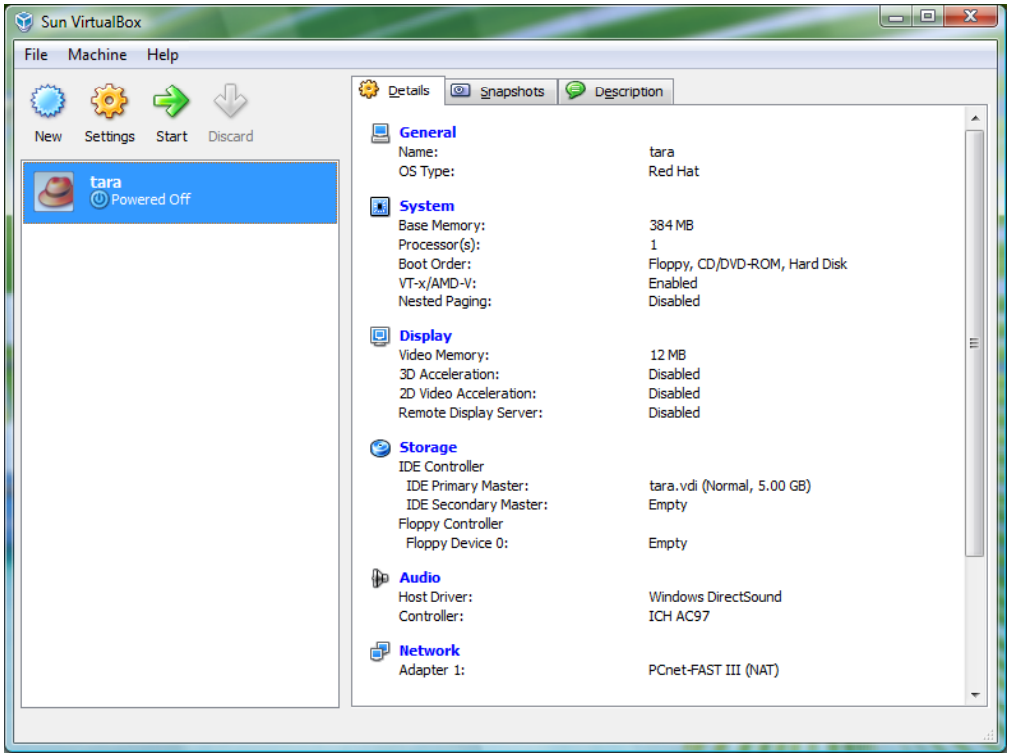
Rich's Lab Schedule

Current schedule is Monday 1-3:30 and Wednesday 1-3:30

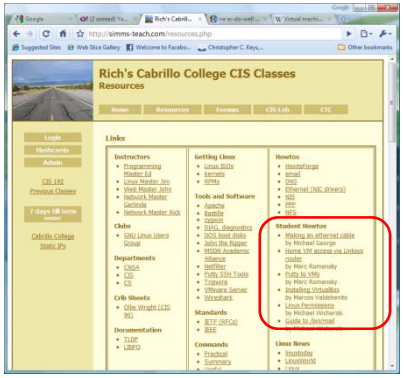
Some other possible options:

- Tuesday 2-7
- Monday 1-4 and Tuesday 5-7
- **Monday 1-4 and Wednesday 5-7**

Sun VirtualBox 3.1.4



Marcos did a Howto on VirtualBox which is on the Resources page of the web site in the Student Howtos section



Thanks Jack for pointing out the support for multiple internal networks!

- ✓ Free download
- ✓ Virtual serial ports
- ✓ Has multiple internal networks
- ✓ Supported on Windows 7 (unlike VMware Server 1.x)
- ✓ Multiple snapshots

Turning in work and getting it graded work back on Opus

Benji (simmsben) submits his lab2 for grading

```
[simmsben@opus ~]$ scp lab2 cis192@opus.cabrillo.edu:lab2.simmsben
cis192@opus.cabrillo.edu's password:
lab2                                          100% 4760     4.7KB/s   00:00
```

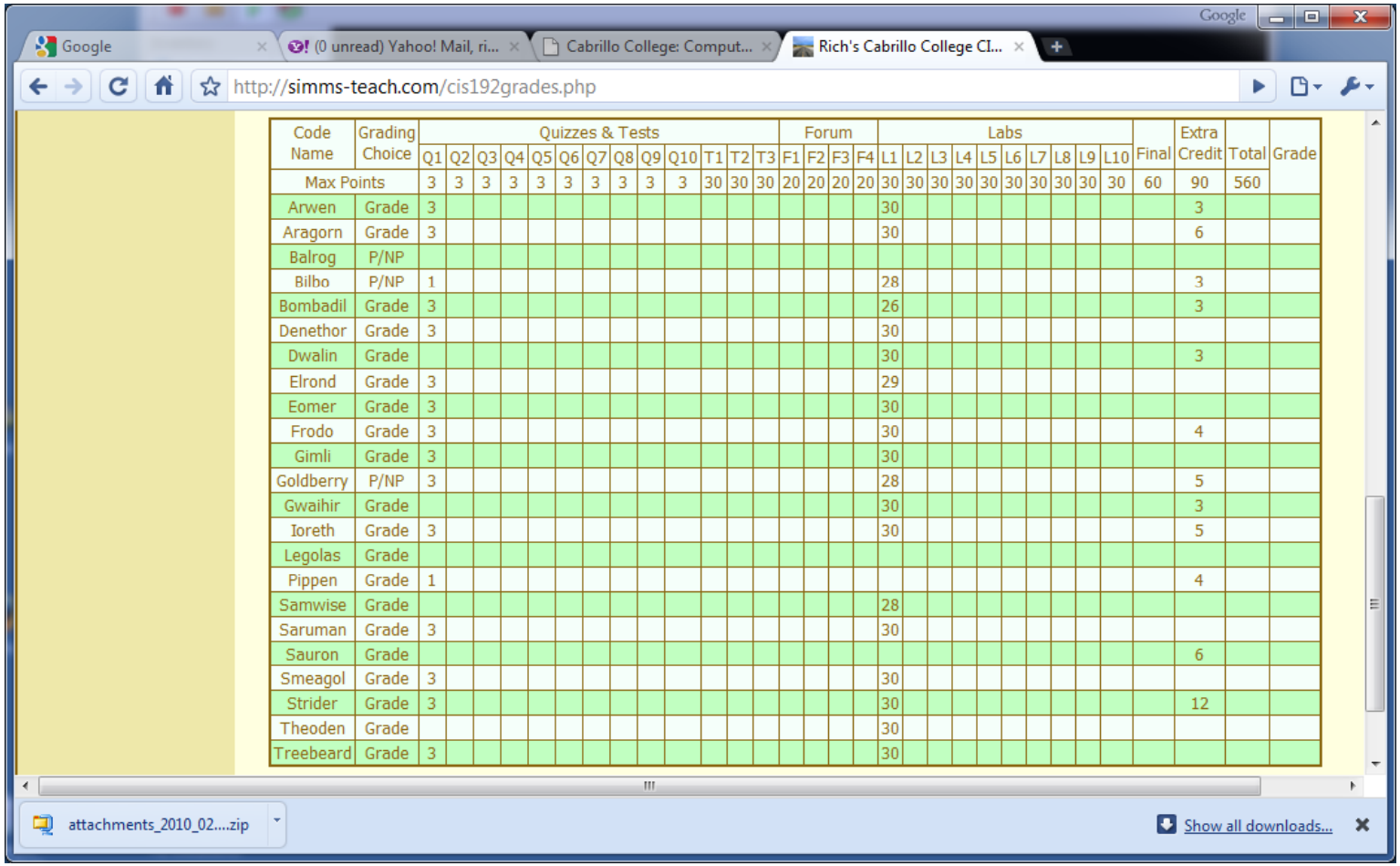
Benji verifies his lab was submitted

```
[simmsben@opus ~]$ ls /home/turnin/cis192/*.simmsben
ls: /home/turnin/cis192/lab1.simmsben: Permission denied
ls: /home/turnin/cis192/lab2.simmsben: Permission denied
```

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

```
[simmsben@opus ~]$ ls -l
total 36
-r-------- 1 simmsben staff  6040 Feb 19 17:22 lab01.simmsben.graded
-rw-r--r-- 1 simmsben cis192 4760 Feb 23 08:52 lab1
-rw-r--r-- 1 simmsben cis192 4760 Feb 23 08:45 lab2
[simmsben@opus ~]$
```

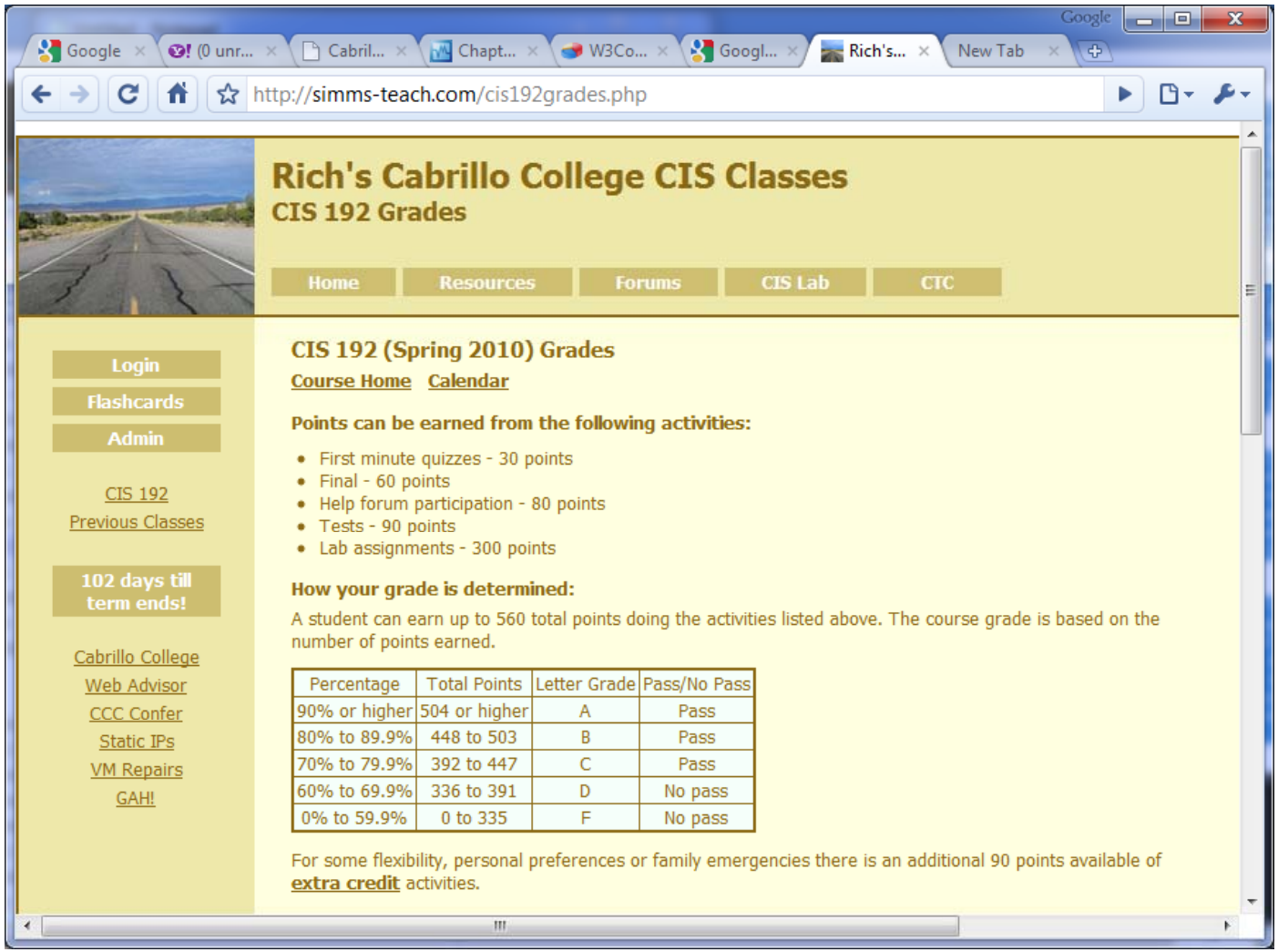
Grades Web Page



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	
Arwen	Grade	3																	30											3		
Aragorn	Grade	3																	30											6		
Balrog	P/NP																															
Bilbo	P/NP	1																	28											3		
Bombadil	Grade	3																	26											3		
Denethor	Grade	3																	30													
Dwalin	Grade																		30											3		
Elrond	Grade	3																	29													
Eomer	Grade	3																	30													
Frodo	Grade	3																	30											4		
Gimli	Grade	3																	30													
Goldberry	P/NP	3																	28											5		
Gwaihir	Grade																		30											3		
Ioreth	Grade	3																	30											5		
Legolas	Grade																															
Pippen	Grade	1																												4		
Samwise	Grade																		28													
Saruman	Grade	3																	30													
Sauron	Grade																													6		
Smeagol	Grade	3																	30													
Strider	Grade	3																	30											12		
Theoden	Grade																		30													
Treebeard	Grade	3																	30													

Check your grading option, quiz #1, Lab #1 and extra credit

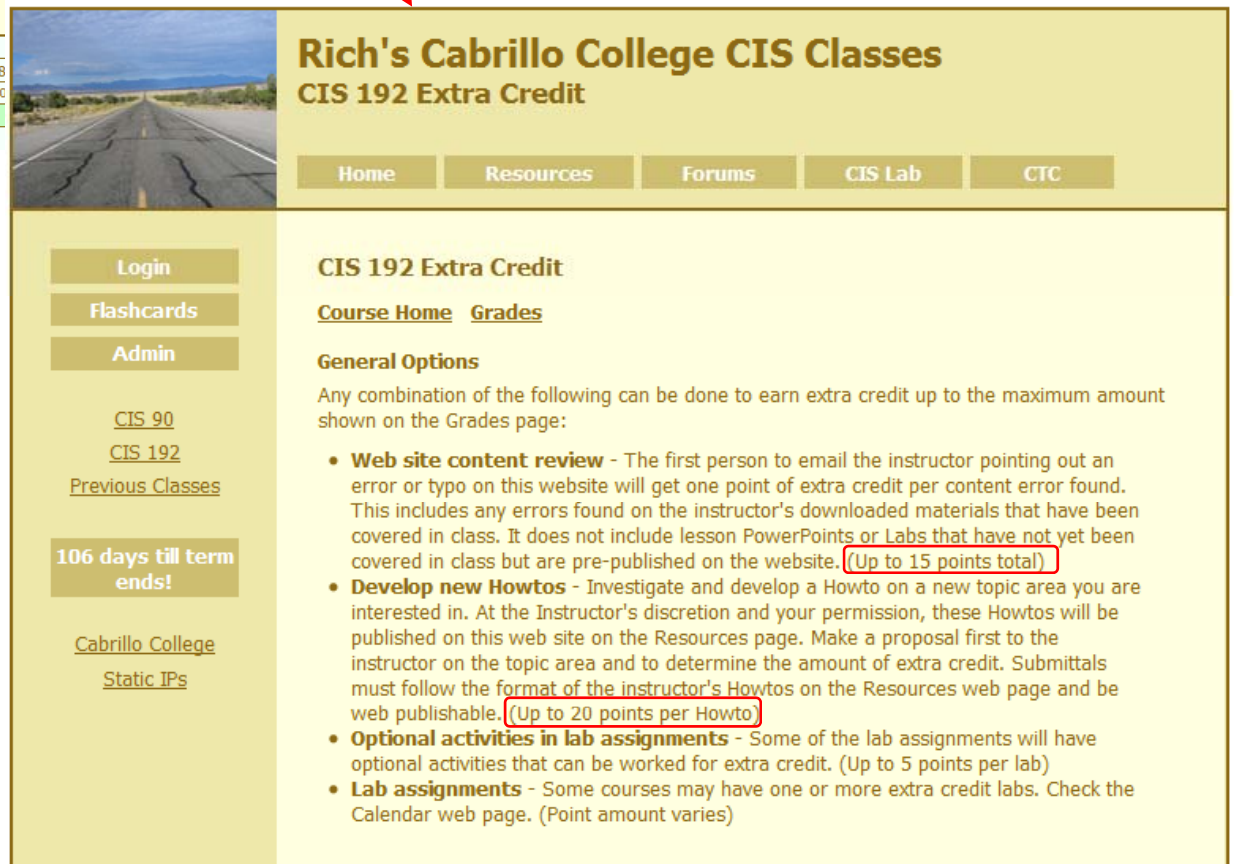
Grades Web Page



Select the grade you want and earn that many points!

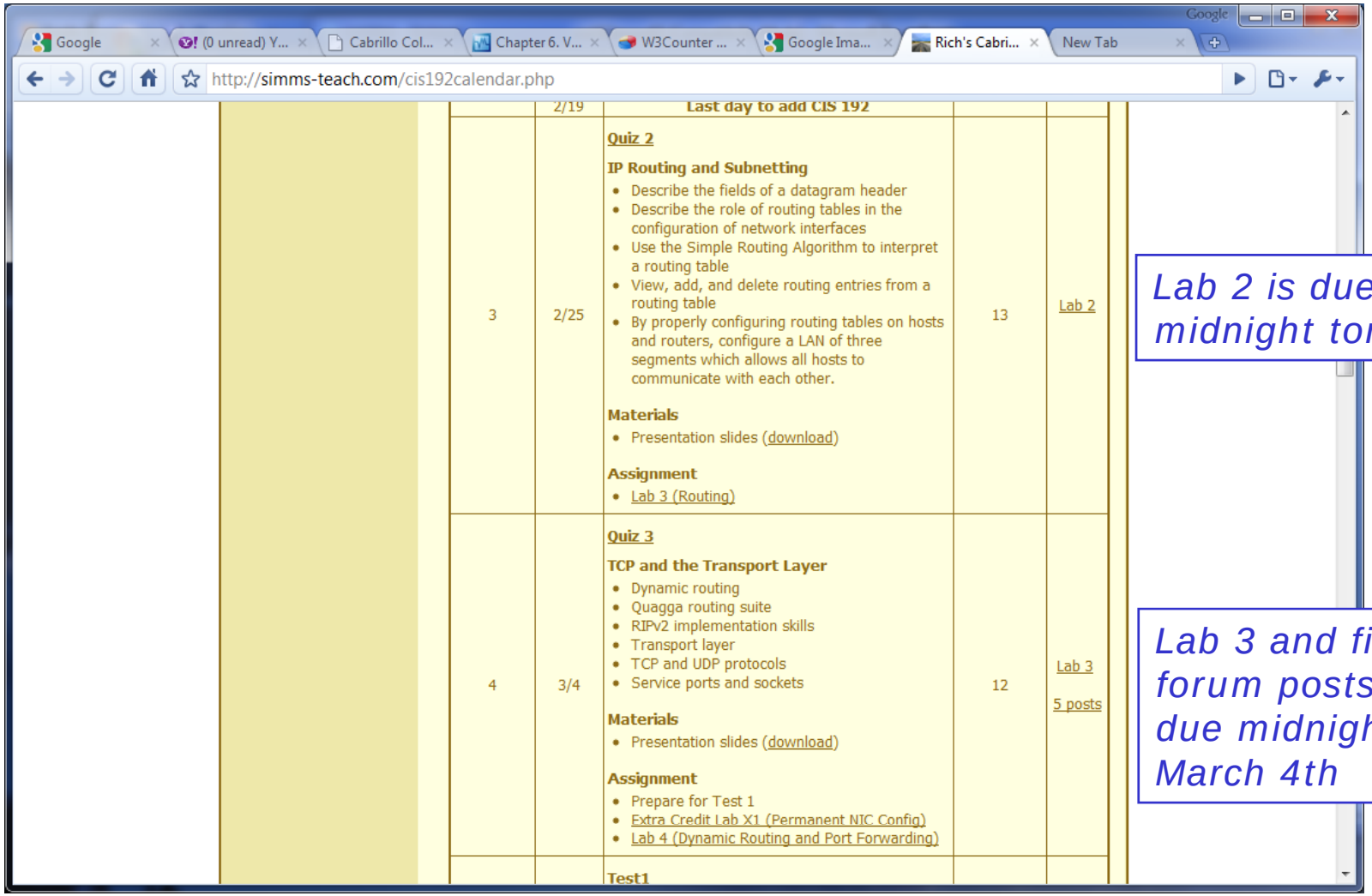
Extra Credit

Link to Extra Credit page is on the Grades page



Note the total extra credit cap is 90 points

Plan ahead with calendar web page



	2/19	**Last day to add CIS 192**		
3	2/25	**Quiz 2** **IP Routing and Subnetting** • Describe the fields of a datagram header • Describe the role of routing tables in the configuration of network interfaces • Use the Simple Routing Algorithm to interpret a routing table • View, add, and delete routing entries from a routing table • By properly configuring routing tables on hosts and routers, configure a LAN of three segments which allows all hosts to communicate with each other. **Materials** • Presentation slides (download) **Assignment** • Lab 3 (Routing)	13	Lab 2
4	3/4	**Quiz 3** **TCP and the Transport Layer** • Dynamic routing • Quagga routing suite • RIPv2 implementation skills • Transport layer • TCP and UDP protocols • Service ports and sockets **Materials** • Presentation slides (download) **Assignment** • Prepare for Test 1 • Extra Credit Lab X1 (Permanent NIC Config) • Lab 4 (Dynamic Routing and Port Forwarding)	12	Lab 3 5 posts
		Test1		

Lab 2 is due midnight tonight

Lab 3 and five forum posts are due midnight March 4th

First test coming up during Lesson 5

Review

Treat VMs as real computers

Always shutdown any running VMs when you are finished

- Powering off a VM is the same as holding down the power button on a real computer. Any pending drive writes will be lost and open files may become corrupted.
- Shutting down your host VMware station before shutting down running VMs can also result in corrupted files for the same reason as above.
- The fastest way to shutdown Linux is to use: **init 0**
- Closing the VMware Server Console does not shut down the VM which will continue to run.
-  Beware! The Shutdown and Restart buttons will only do a graceful shutdown if VMware Tools have been installed on the VM. Otherwise this is the same as pulling the power cable or pressing the motherboard reset button and you can lose data. The CentOS VMs have VMware Tools installed.

Linux Networking Review Examples

startx (start up X windows and desktop)
Alternate command Terminals
- Ctrl-alt-1
- Ctrl-alt-2
- Ctrl-alt-3
- Ctrl-alt-4
- Ctrl-alt-5
- Ctrl-alt-6

Graphical desktop
- Ctrl-alt-7

shutdown now
init 0 (fastest way to shut down)
init 1 (minimal system)
init 3 (normal system)
init 5 (X windows graphics mode)

su (gets you to root, without path so must use \sbin\ifconfig)
su - (gets you to root with root's path)

Linux Networking Review Examples

lspci	(list PCI hardware including NIC)
dmesg	(shows ton of HW info)
more /var/log/dmesg	(same as dmesg, but not the latest entries)
lsmod	(view already installed drivers)
more /proc/modules	(same as lsmod, different format)

To research Linux network driver info:

http://www.tldp.org/HOWTO/text/Ethernet-HOWTO
http://www.tldp.org/HOWTO/Hardware-HOWTO/nic.html

Network drivers (hopefully already in /lib/modules/2.4.20-8/kernel/drivers/net)

- e100 - for Intel Ethernet PRO 100 NIC
- 8139too - for D-Link NIC with RealTek 8129/8139 chipsets
- 3c59x - for 3Com 3c905x NICs
- pcnet32 - for AMD 79c970 NICs in VMware VMs
- tulip - for Lite-on Communications LNE 100TX cards with DEC chipsets

Leave off the .o or .ko suffix when specifying drivers:

insmod 3c59x	(installs 3c59x driver - no dependencies)
modprobe pcnet32	(installs driver with/withoot dependencies)
rmmod pcnet32	(removes pcnet32 driver)

Linux Networking Review Examples

Controlling interfaces

ifconfig	(shows active interfaces)
ifconfig eth0 up	(brings up eth0)
ifconfig eth0 down	(brings up eth0)

Configuring interfaces

```
ifconfig eth1 172.30.4.106 netmask 255.255.255.0   (adds IP/mask)
ifconfig eth1:1 172.30.4.206 netmask 255.255.255.0   (adds alias IP/mask)
```

Configure default gateways with:

```
route add default gw 192.168.2.6
route del default gw 192.168.2.6
```

To display routing table:

```
route –n
```

TUI interface configuration tool:

netconfig	(RH9)
system-config-network	(newer versions of RH)

Linux Networking Review Examples

To show name servers:

```
cat /etc/resolv.conf
```

To configure nameservers:

```
echo “nameserver 207.62.187.53” > /etc/resolv.conf
```

or vi /etc/resolv.conf (and hand edit)

```
nameserver 207.62.187.53        (IP address of primary name server)
nameserver XXX.XXX.XXX.XXX      (IP address of secondary name server)
search cabrillo.edu             (domain suffix to add for short names)
```

Remote logins:

```
ssh 172.30.4.107                (uses current login name)
ssh root@172.30.4.107           (specifies a login name)
```

Linux Networking Review Examples

arp –n	(show arp cache)
service arpwatch start | stop	(Red Hat, monitor arp cache entries)
/etc/init.d/arpwatch start | stop	(Ubuntu, monitor arp cache entries)
/var/arpwatch/arp.dat	(Red Hat, file of collected IP/MAC pairs)
/var/lib/arpwatch/arp.dat	(Ubuntu, file of collected IP/MAC pairs)
yum install arpwatch	(Red Hat new package install, needs Internet)
apt-get install ipcalc	(Ubuntu new package install, needs Internet)

ping 172.30.4.1
ping –Rc3 opus.cabrillo.eduipcalc
traceroute <ip address>
mtr google.com
netstat –i (show interface stats)

tcpdump src 172.30.1.105 or dst 172.30.1.105
wireshark

Tangent on the older Red Hat 9

(Red Hat Family)

What happens with RH9 if you don't specify the broadcast address for the 172 net in the classroom?

```
[root@rh9 root]# ifconfig eth0 172.30.1.201 netmask 255.255.255.0
[root@rh9 root]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:28:3A:0C
          inet addr:172.30.1.201  Bcast:172.30.255.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:926 (926.0 b)  TX bytes:830 (830.0 b)
          Interrupt:9 Base address:0x10a4
```

Answer: Trouble! We are using a /24 network in the classroom yet RH9 calculates a /16 based broadcast address!

RH9 calculates the broadcast address using the older classful method. The 172 network is a class B network where the first 16 bits form the network portion of the address. In CIDR this would be /16 network

Tangent on the older Red Hat 9

(Red Hat Family)

Specify both the netmask and broadcast address when using RH9

```
[root@rh9 root]# ifconfig eth0 172.30.1.201 netmask 255.255.255.0 broadcast 172.30.1.255
[root@rh9 root]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:28:3A:0C
          inet addr:172.30.1.201  Bcast:172.30.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:926 (926.0 b)  TX bytes:830 (830.0 b)
          Interrupt:9 Base address:0x10a4

[root@rh9 root]#
```

Note the broadcast address is calculated correctly with the newer VMs we are using in this course. The only VM that uses RH9 is Nosmo. For the rest of the class VMs you do not have to specify the broadcast address on the ifconfig command.

Subnetting

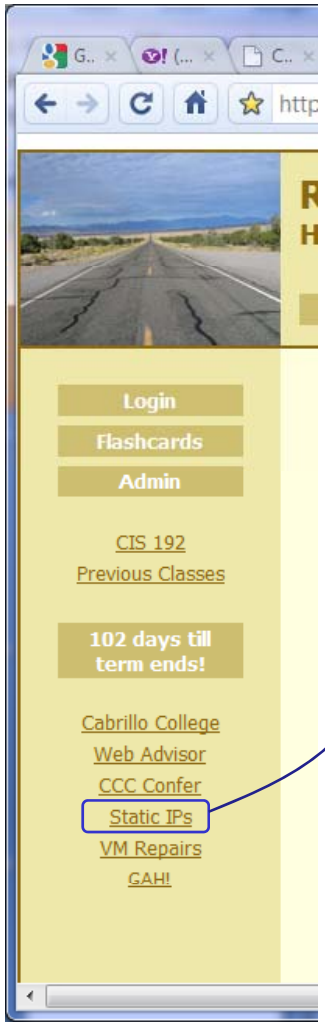
By hand

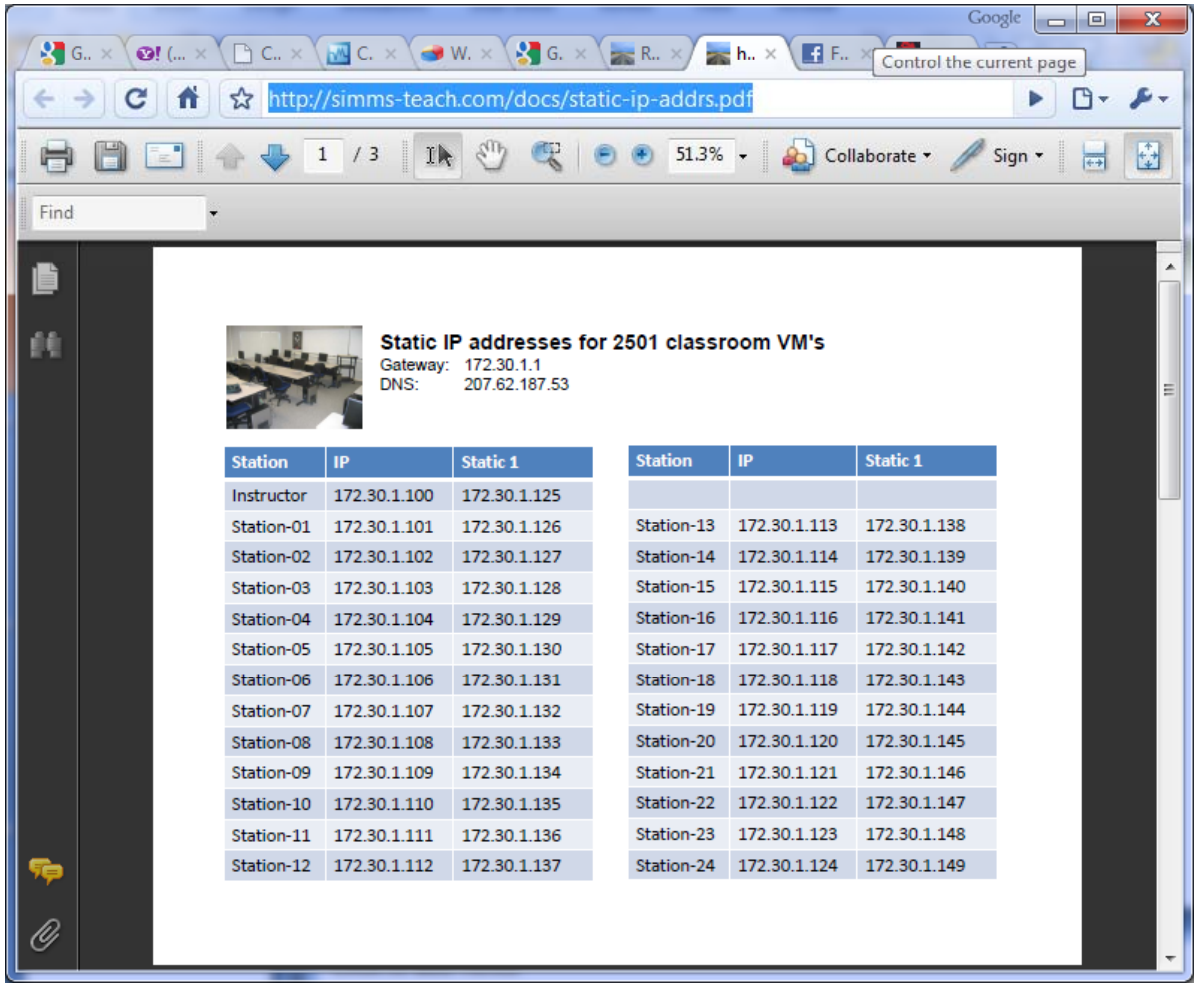
```
0000 0001 = 1
0000 0010 = 2
0000 0100 = 4
0000 1000 = 8
0001 0000 = 16
0010 0000 = 32
0100 0000 = 64
1000 0000 = 128

1100 0000 = 192
1110 0000 = 224
1111 0000 = 240
1111 1000 = 248
1111 1100 = 252
1111 1110 = 254
1111 1111 = 255
```

With ipcalc

```
root@frodo:~# ipcalc -nb 10.10.15.48/28
Address:   10.10.15.48
Netmask:   255.255.255.240 = 28
Wildcard:  0.0.0.15
=>
Network:   10.10.15.48/28
HostMin:   10.10.15.49
HostMax:   10.10.15.62
Broadcast: 10.10.15.63
Hosts/Net: 14                    Class A, Private
Internet
```

Tonight we will use the 10.10.15.48/28 network for Celebrian and Sauron



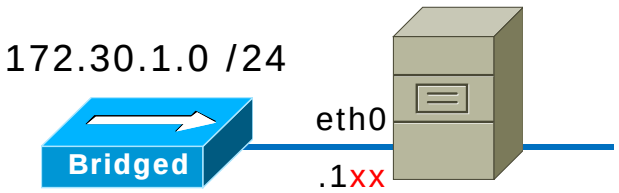


Static IP addresses for 2501 classroom VM's
Gateway: 172.30.1.1
DNS: 207.62.187.53

Station	IP	Static 1
Instructor	172.30.1.100	172.30.1.125
Station-01	172.30.1.101	172.30.1.126
Station-02	172.30.1.102	172.30.1.127
Station-03	172.30.1.103	172.30.1.128
Station-04	172.30.1.104	172.30.1.129
Station-05	172.30.1.105	172.30.1.130
Station-06	172.30.1.106	172.30.1.131
Station-07	172.30.1.107	172.30.1.132
Station-08	172.30.1.108	172.30.1.133
Station-09	172.30.1.109	172.30.1.134
Station-10	172.30.1.110	172.30.1.135
Station-11	172.30.1.111	172.30.1.136
Station-12	172.30.1.112	172.30.1.137

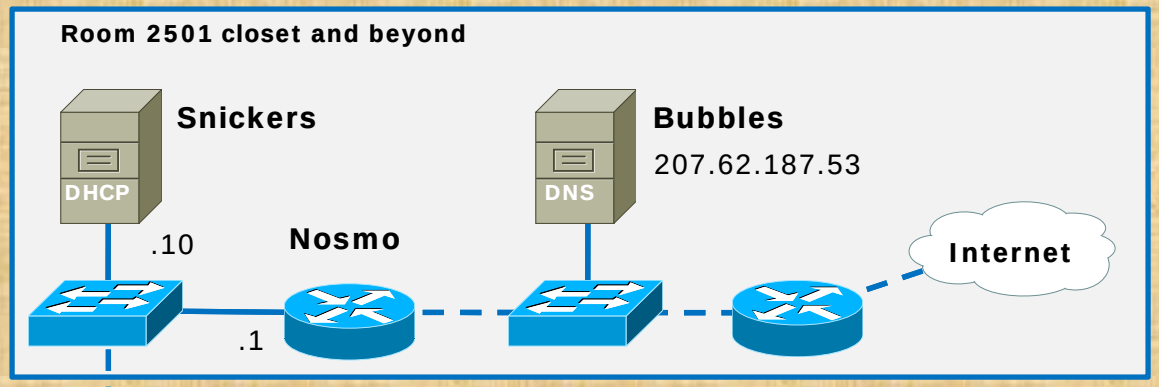
Station	IP	Static 1
Station-13	172.30.1.113	172.30.1.138
Station-14	172.30.1.114	172.30.1.139
Station-15	172.30.1.115	172.30.1.140
Station-16	172.30.1.116	172.30.1.141
Station-17	172.30.1.117	172.30.1.142
Station-18	172.30.1.118	172.30.1.143
Station-19	172.30.1.119	172.30.1.144
Station-20	172.30.1.120	172.30.1.145
Station-21	172.30.1.121	172.30.1.146
Station-22	172.30.1.122	172.30.1.147
Station-23	172.30.1.123	172.30.1.148
Station-24	172.30.1.124	172.30.1.149

Celebrian



What static IP address can the student sitting at station 12 in this room use for Celebrian?

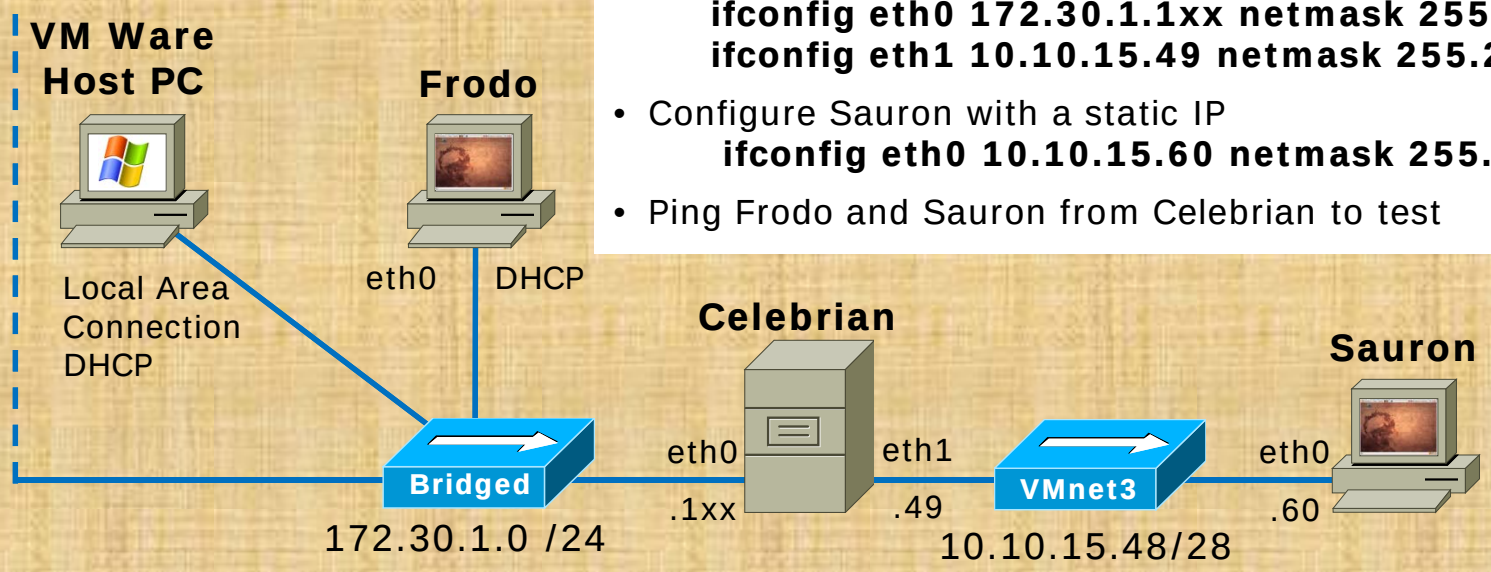
Class Exercise – join VMs to network




```
Network:    10.10.15.48/28
HostMin:    10.10.15.49
HostMax:    10.10.15.62
Broadcast:  10.10.15.63
```

Configure Celebrian and Sauron interfaces

- Configure Celebrian with static IPs
  ifconfig eth0 172.30.1.1xx netmask 255.255.255.0
  ifconfig eth1 10.10.15.49 netmask 255.255.255.240
- Configure Sauron with a static IP
  ifconfig eth0 10.10.15.60 netmask 255.255.255.240
- Ping Frodo and Sauron from Celebrian to test



Why can't Frodo ping Sauron?

Permanent Network Configuration

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 --status-all
- service *xxxxxx* <stop|start|restart|status>

Show all services

Control a specific service

System startup configuration

- chkconfig --list
- chkconfig [--level levels] *xxxxxx* <on|off>

Configuring Permanent Network Settings
(Red Hat Family)

Restarting network services

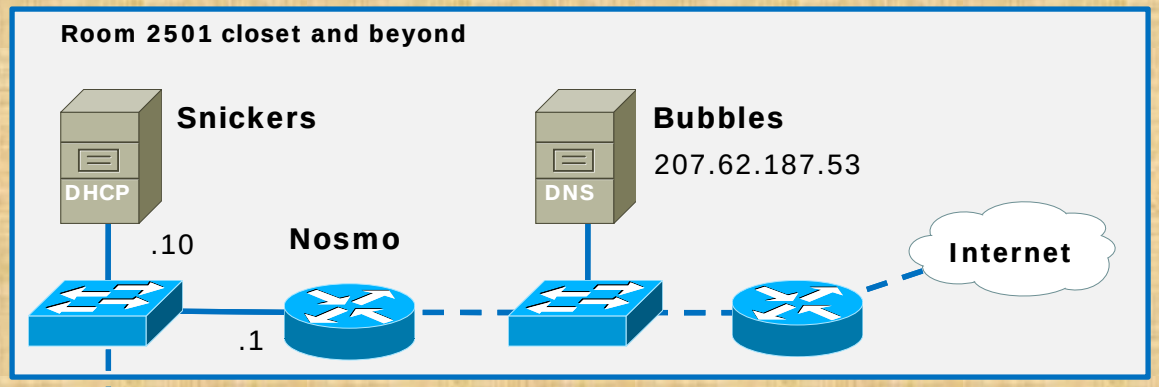
```
[root@elrond ~]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:
Determining IP information for eth0... done.
                                                           [  OK  ]
```

or

```
[root@elrond ~]# /etc/init.d/network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:
Determining IP information for eth0... done.
                                                           [  OK  ]
```

For Ubuntu 8.10: **/etc/init.d/networking restart**
For OpenSUSE 11.1: **rcnetwork restart**

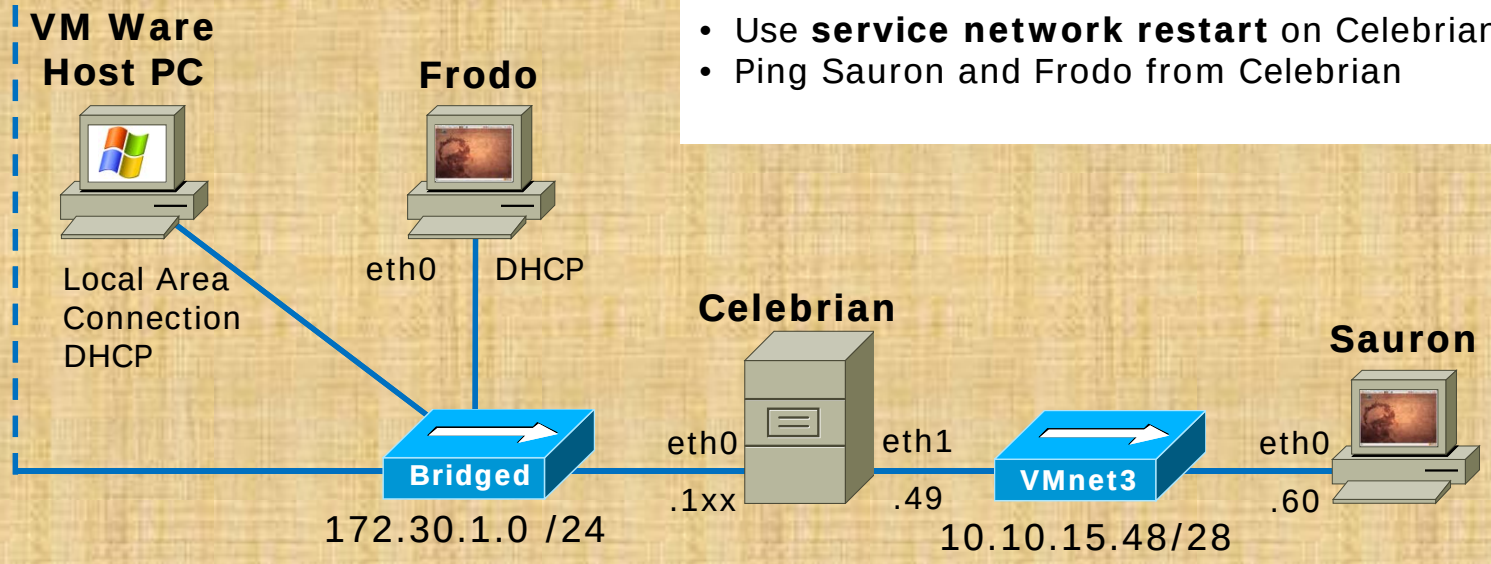
Class Exercise – Restart the network service



We can restart the systems, or to same time just restart the network services

Restart network service on Celebrian

- Use **service network restart** on Celebrian
- Ping Sauron and Frodo from Celebrian



Why can't Celebrian ping Sauron or Frodo now?

Set Static IP Address and Subnet Mask

(Red Hat Family)

Temporary

- **ifconfig eth0 172.30.4.125 netmask 255.255.255.0 broadcast 172.30.4.255**

Permanent

Use tab completion when typing!

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

There is a different file for each interface

```
[root@elrond ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0
ONBOOT=yes
HWADDR=00:0c:29:ba:63:76
BOOTPROTO=static
IPADDR=172.30.4.125
NETMASK=255.255.255.0
BROADCAST=172.30.4.255
[root@elrond ~]#
```

Add these static IP settings

Specifying the broadcast is optional. It was required on older versions of Linux where the netmask differed from the classful mask.

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

For new settings to take effect

Set Static IP Address and Subnet Mask

(Red Hat Family)

Verify

- Use **ifconfig** and **ping** commands

```
[root@elrond ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:BA:63:76
          inet addr:172.30.4.125  Bcast:172.30.4.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feba:6376/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:556 errors:0 dropped:0 overruns:0 frame:0
          TX packets:495 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:61635 (60.1 KiB)  TX bytes:82641 (80.7 KiB)
          Interrupt:177 Base address:0x1400

[root@elrond ~]# ping -c 1 172.30.4.1
PING 172.30.4.1 (172.30.4.1) 56(84) bytes of data.
64 bytes from 172.30.4.1: icmp_seq=1 ttl=255 time=3.61 ms

--- 172.30.4.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.617/3.617/3.617/0.000 ms
[root@elrond ~]#
```

Set Dynamic IP Address and Subnet Mask

(Red Hat Family)

Temporary

- **dhclient** and **dhclient –r** to release

Permanent

Use tab completion when typing!

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

There is a different file for each interface

```
[root@elrond ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0
ONBOOT=yes
HWADDR=00:0c:29:ba:63:76
BOOTPROTO=dhcp
[root@elrond ~]#
```

Add this for DHCP

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

For new settings to take effect

Set Dynamic IP Address and Subnet Mask

(Red Hat Family)

Verify

- Use **ifconfig** and **ping** commands

```
[root@elrond ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:BA:63:76
          inet addr:172.30.4.168  Bcast:172.30.4.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feba:6376/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3548 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2135 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:824033 (804.7 KiB)  TX bytes:287392 (280.6 KiB)
          Interrupt:177 Base address:0x1400

[root@elrond ~]# ping -c 1 google.com
PING google.com (74.125.67.100) 56(84) bytes of data.
64 bytes from gw-in-f100.google.com (74.125.67.100): icmp_seq=1 ttl=240 time=44.0 ms

--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 44.088/44.088/44.088/0.000 ms
[root@elrond ~]#
```

Snickers will set the DNS server as well

Getting a dynamic address using dhclient

```
[root@elrond ~]# dhclient
Internet Systems Consortium DHCP Client V3.0.5-RedHat
Copyright 2004-2006 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/eth1/00:0c:29:ba:63:80
Sending on   LPF/eth1/00:0c:29:ba:63:80
Listening on LPF/eth0/00:0c:29:ba:63:76
Sending on   LPF/eth0/00:0c:29:ba:63:76
Sending on   Socket/fallback
DHCPDISCOVER on eth1 to 255.255.255.255 port 67 interval 6
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 4
DHCPOFFER from 172.30.4.10
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 172.30.4.10
bound to 172.30.4.168 -- renewal in 3414 seconds.
[root@elrond ~]# _
```

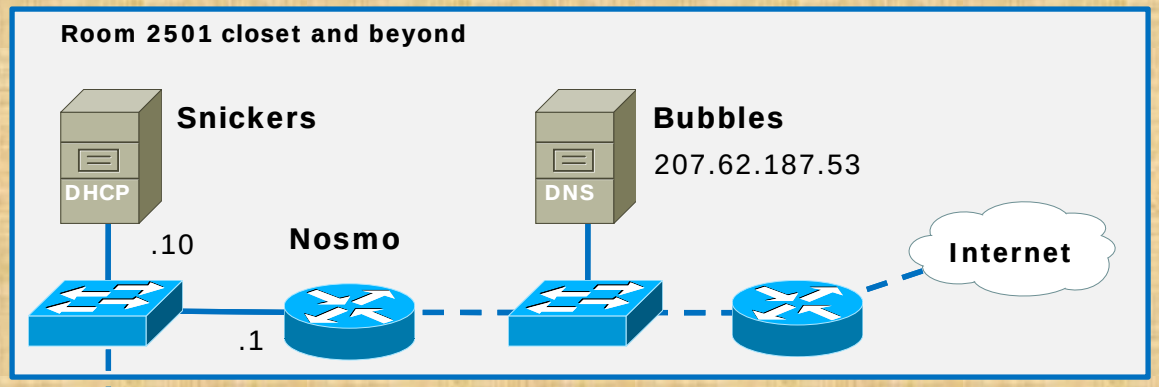
*Using **dhclient** to get an IP address*

Releasing a dynamic address using dhclient

```
[root@elrond ~]# dhclient -r
Internet Systems Consortium DHCP Client V3.0.5-RedHat
Copyright 2004-2006 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/eth1/00:0c:29:ba:63:80
Sending on   LPF/eth1/00:0c:29:ba:63:80
Listening on LPF/eth0/00:0c:29:ba:63:76
Sending on   LPF/eth0/00:0c:29:ba:63:76
Sending on   Socket/fallback
DHCPRELEASE on eth0 to 172.30.4.10 port 67
[root@elrond ~]# _
```

*Using **dhclient –r** to release an IP address*

Class Exercise – Configure interface settings permanently

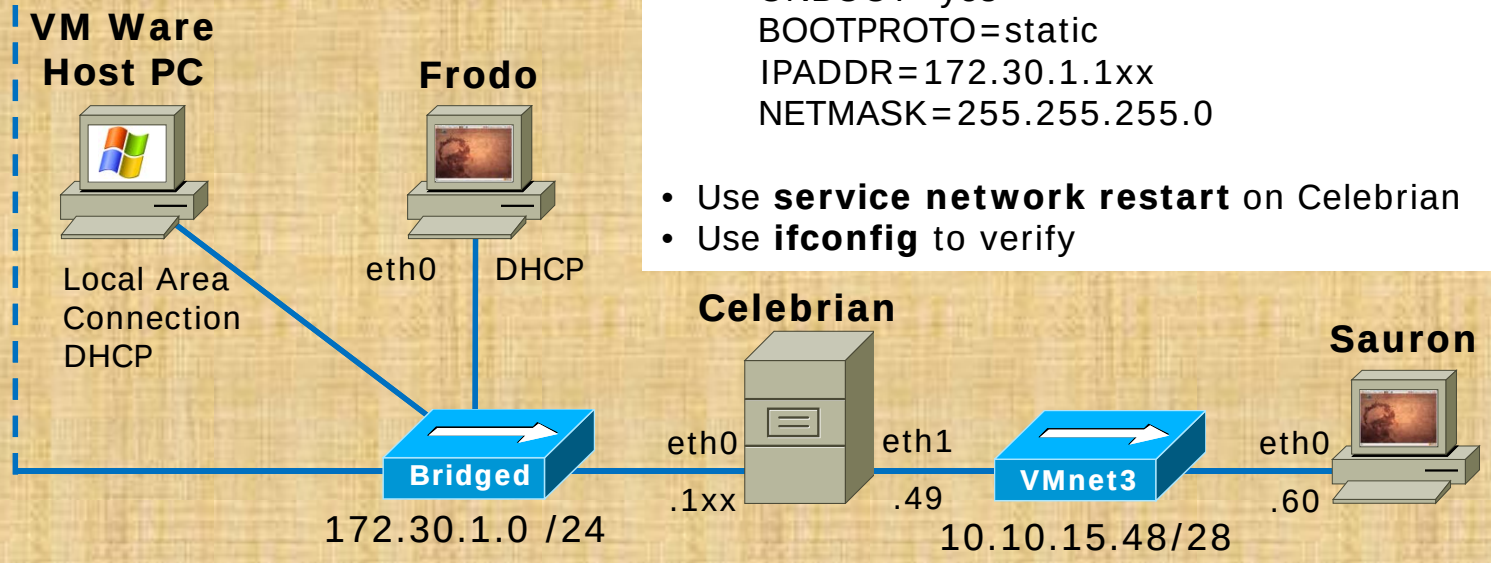


*Lets configure the **eth0** interface on Celebrian*

- Edit **/etc/sysconfig/network-scripts/ifcfg-eth0** to be:

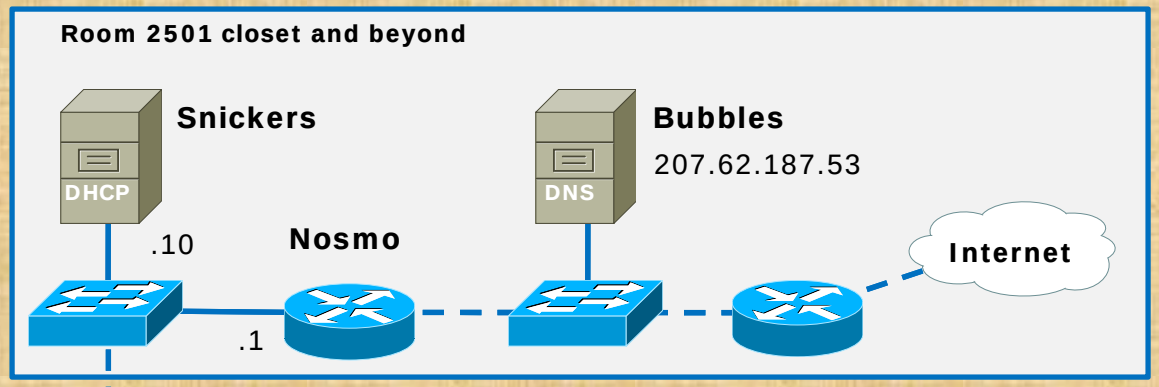
```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=172.30.1.1xx
NETMASK=255.255.255.0
```

- Use **service network restart** on Celebrian
- Use **ifconfig** to verify



Can Celebrian ping Frodo now?

Class Exercise – Configure interface settings permanently

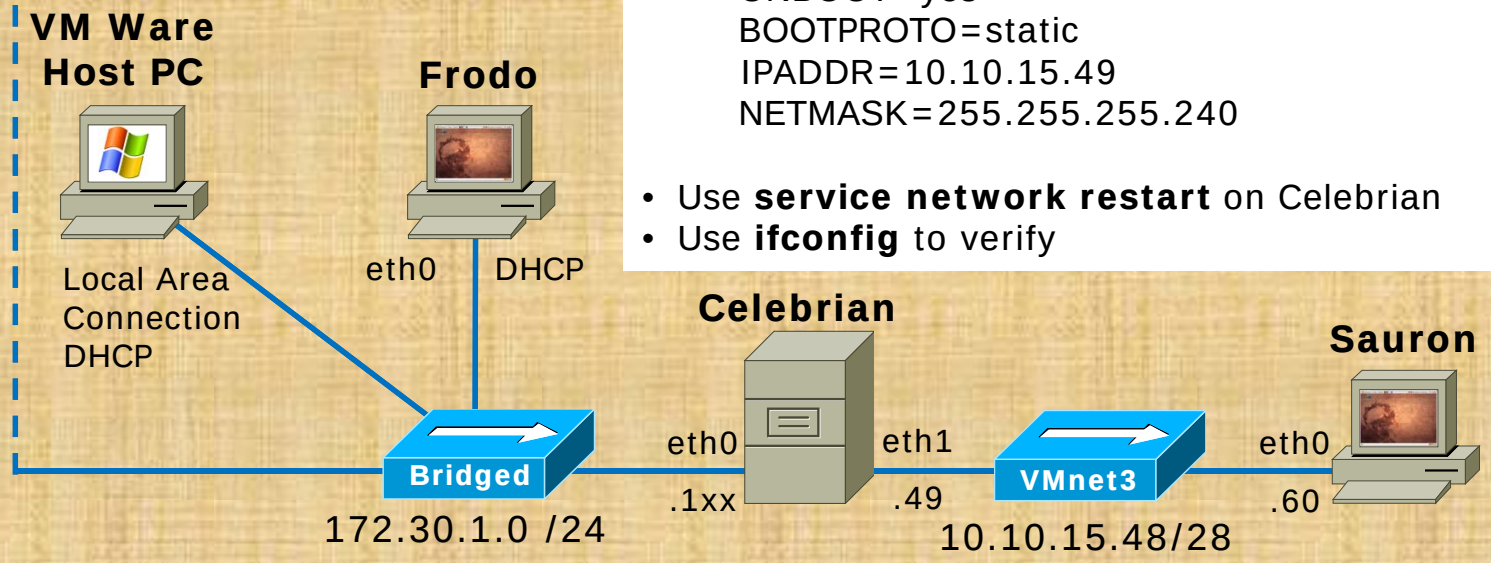


*Lets configure the **eth1** interface on Celebrian*

- Edit **/etc/sysconfig/network-scripts/ifcfg-eth1** to be:

```
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=static
IPADDR=10.10.15.49
NETMASK=255.255.255.240
```

- Use **service network restart** on Celebrian
- Use **ifconfig** to verify



Can Celebrian ping Sauron and Frodo now?

Configuring the default gateway

(Red Hat Family)

Temporary

- **route add default gw 172.30.4.1**

Permanent

- Edit **/etc/sysconfig/network**

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

Add the gateway

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

For new settings to take effect

Configuring the default gateway

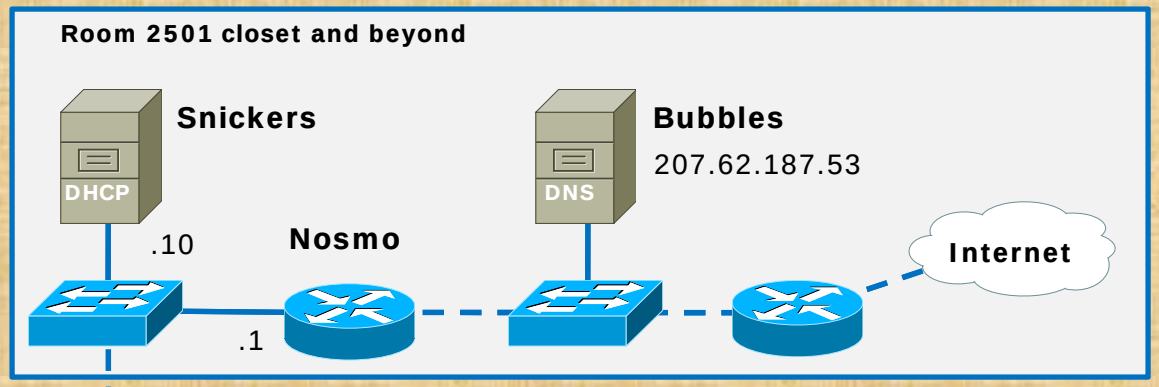
(Red Hat Family)

Verify

- Use **route** to verify

```
[root@elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.4.0      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 eth0
0.0.0.0         172.30.4.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond ~]#
```

Class Exercise – Permanent default gateway

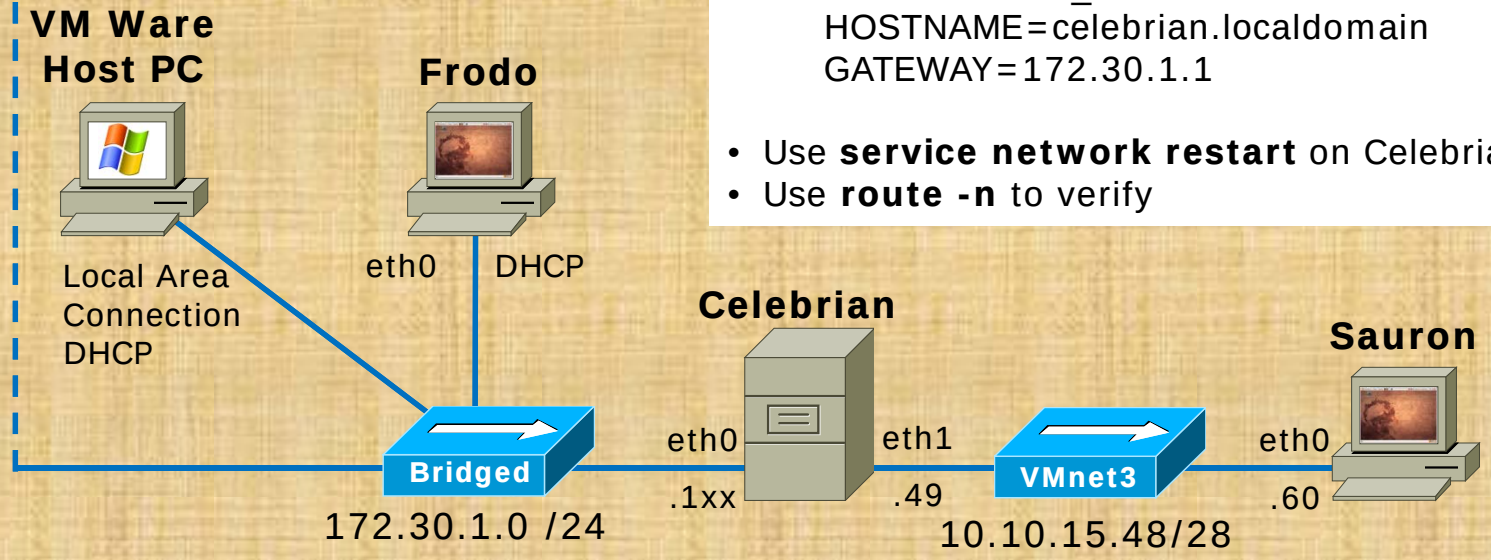


Lets configure the default gateway on Celebrian permanently

- Edit **/etc/sysconfig/network** to be:

  NETWORKING=yes
  NETWORKING_IPV6=no
  HOSTNAME=celebrian.localdomain
  GATEWAY=172.30.1.1

- Use **service network restart** on Celebrian
- Use **route -n** to verify



Can Celebrian ping Sauron and Frodo now?

Configuring the DNS

Permanent

- Edit **/etc/resolv.conf**

```
[root@elrond ~]# cat /etc/resolv.conf
search cabrillo.edu
nameserver 207.62.187.53
[root@elrond ~]#
```

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

May add up to three of these for primary , secondary and tertiary DNS servers

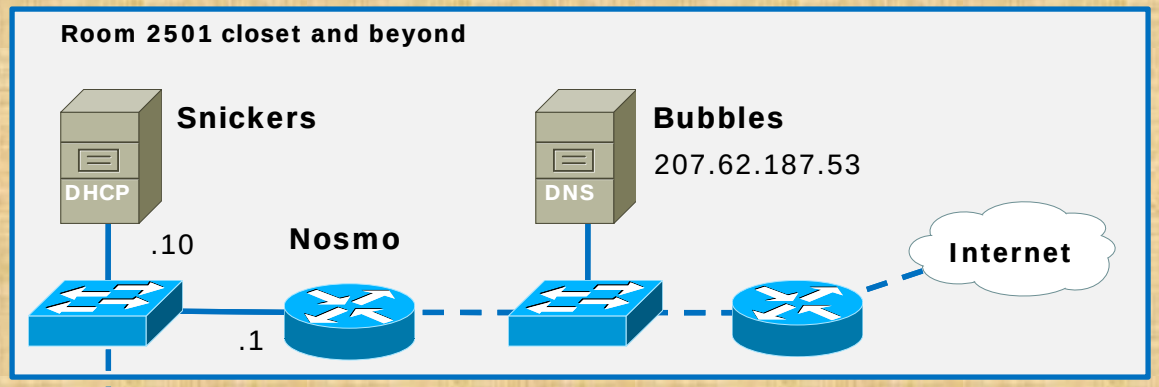
Verify

- Ping by hostname

```
[root@elrond ~]# ping -c 1 opus
PING opus.cabrillo.edu (207.62.186.9) 56(84) bytes of data.
64 bytes from opus.cabrillo.edu (207.62.186.9): icmp_seq=1 ttl=63 time=3.67 ms

--- opus.cabrillo.edu ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.671/3.671/3.671/0.000 ms
[root@elrond ~]#
```

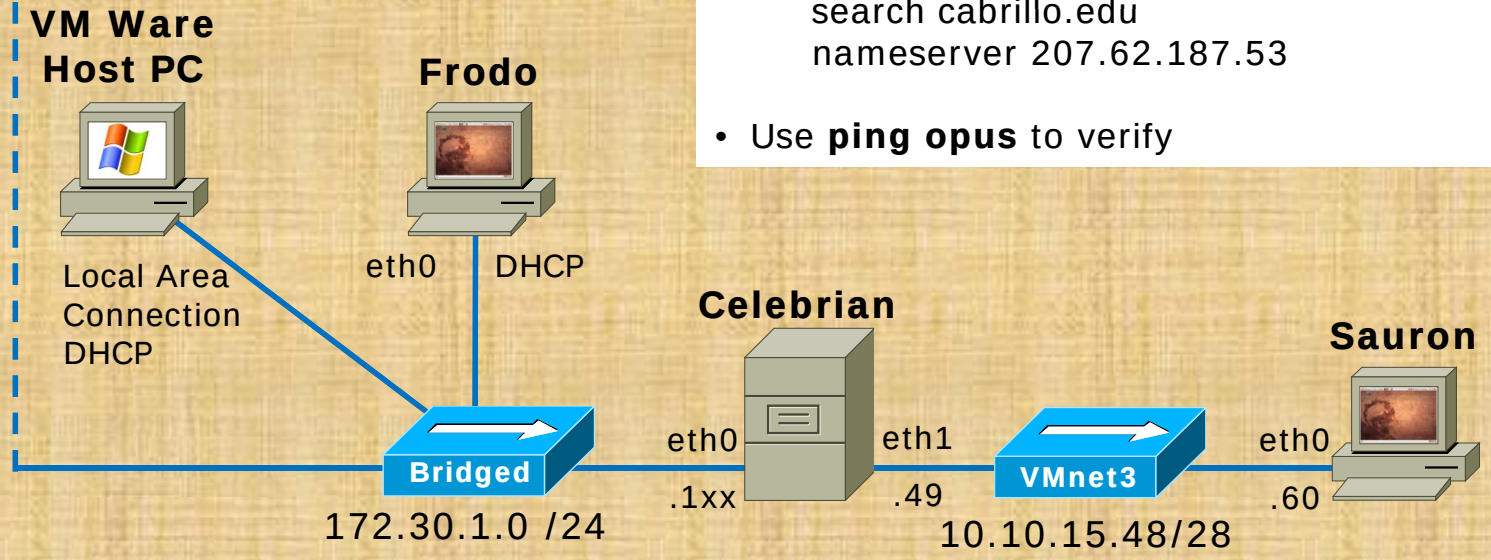
Note: On the ping, we can leave the .cabrillo.edu off the hostname since we have it in the search string in /etc/resolv.conf

Class Exercise – Set the DNS nameserver



Lets configure the DNS settings on Celebrian to use Bubbles as the primary name server

- Edit **/etc/resolv.conf** to be:
  search cabrillo.edu
  nameserver 207.62.187.53
- Use **ping opus** to verify



Can Celebrian ping Sauron and Frodo now?

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)

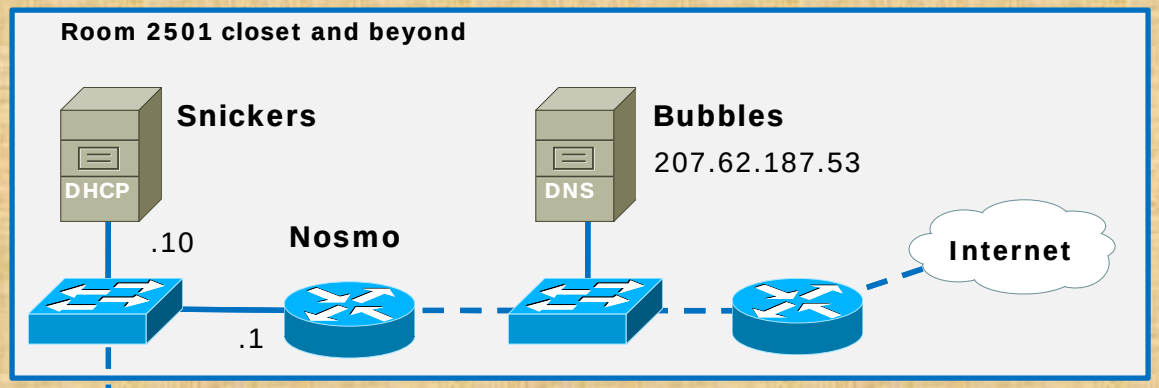
Permanent: Step 2 - edit **/etc/hosts**

```
[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

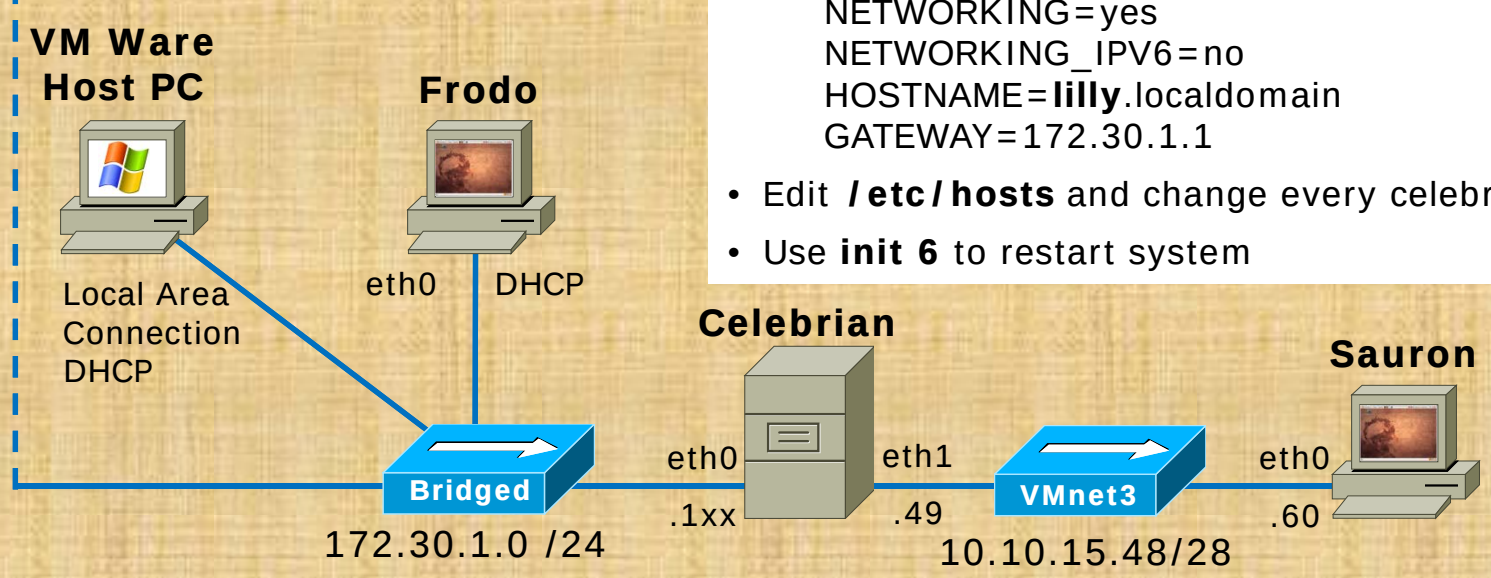
```
[root@homer ~]# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       homer.localdomain homer localhost.localdomain localhost
::1             localhost6.localdomain6 localhost6
[root@homer ~]#
```

Class Exercise – Change hostname



Rename Celebrian to Lilly. Note we are changing the system hostname, not the VM name!

- Edit **/etc/sysconfig/network** and change celebrian to lilly:
  NETWORKING=yes
  NETWORKING_IPV6=no
  HOSTNAME=**lilly**.localdomain
  GATEWAY=172.30.1.1
- Edit **/etc/hosts** and change every celebrian to lilly
- Use **init 6** to restart system



What happens to your prompt now after the Celebrian VM restarts?

/etc/hosts

ping frodo vs **ping 172.30.4.150**
ssh frodo vs **ssh 172.30.4.150**
scp myfile frodo: vs **ssh myfile 172.30.4.150:**

- 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@elrond ~]# 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
192.168.2.123   legolas
172.30.4.150    frodo
172.30.4.1      nosmo
192.168.3.200   sauron
[root@elrond ~]#
```

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

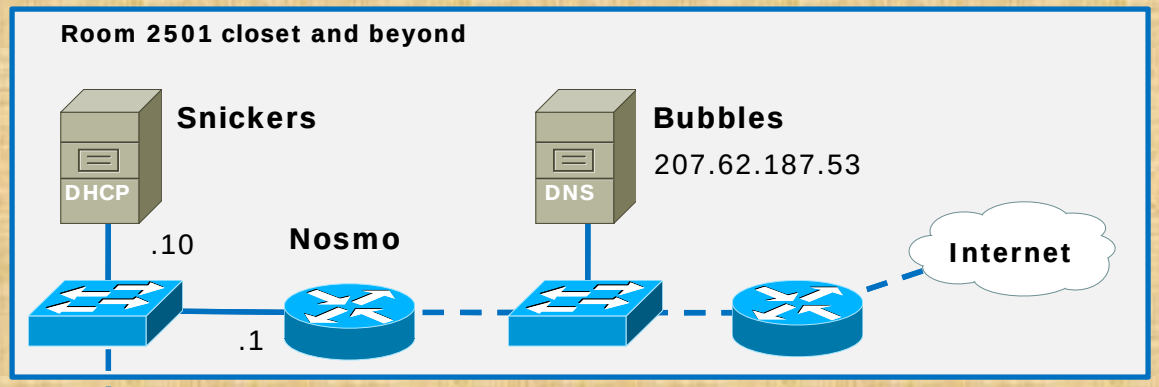
/etc/hosts

using a name rather than an IP address

```
[root@elrond ~]# ping nosmo
PING nosmo (172.30.4.1) 56(84) bytes of data.
64 bytes from nosmo (172.30.4.1): icmp_seq=1 ttl=255 time=1.55 ms

--- nosmo ping statistics ---
2 packets transmitted, 1 received, 50% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.554/1.554/1.554/0.000 ms
[root@elrond ~]# ping sauron
PING sauron (192.168.3.200) 56(84) bytes of data.
64 bytes from sauron (192.168.3.200): icmp_seq=1 ttl=63 time=1.61 ms

--- sauron ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.616/1.616/1.616/0.000 ms
[root@elrond ~]#
```

/etc/hosts was updated with nosmo and sauron and their IP addresses

Class Exercise – Add remote hostname / IP pairs

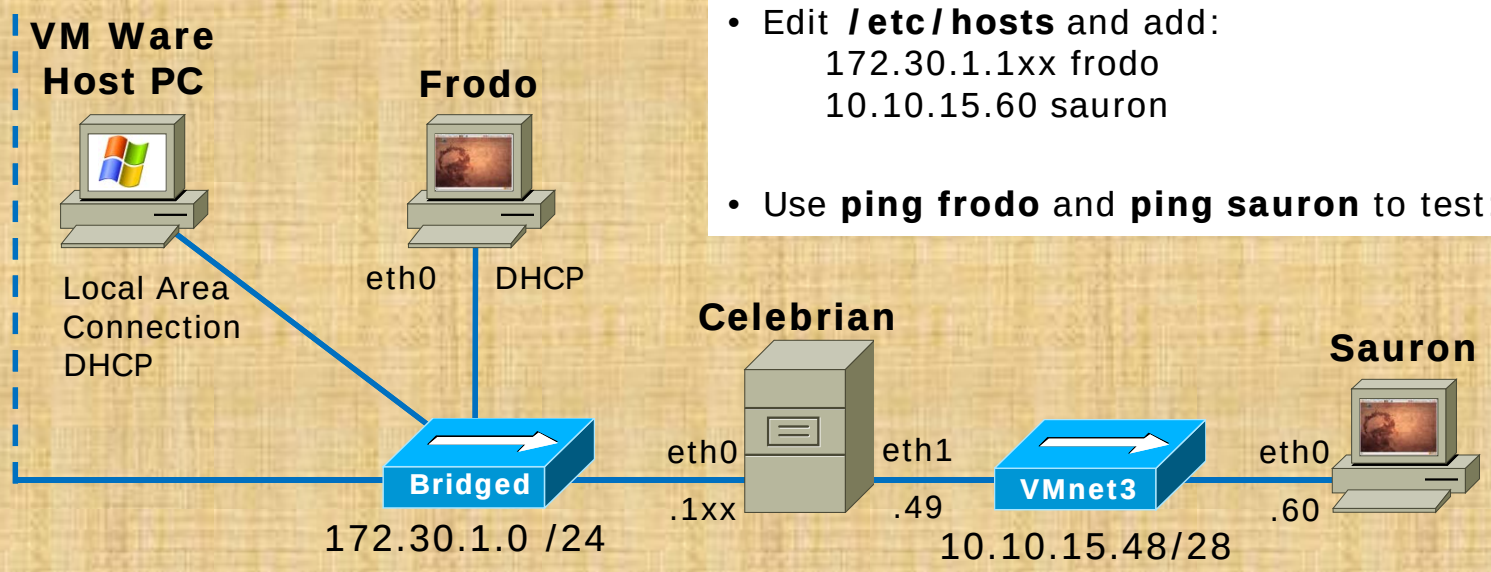


Lets add some remote hostnames so we don't have to type the IP address each time

On Celebrian (now Lilly):

- Edit **/etc/hosts** and add:
  172.30.1.1xx frodo
  10.10.15.60 sauron

- Use **ping frodo** and **ping sauron** to test:



What happens to your prompt now after the Celebrian VM restarts?

Debian/Ubuntu Network Config

Debian/Ubuntu
Permanent Network Configuration

Configuring DHCP on an interface

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

auto eth0
iface eth0 inet dhcp

root@sun:~#
```

Always need a loopback device on Linux systems

Setting a DHCP IP address

Debian/Ubuntu configuration differs from the Red Hat family. It is a little nicer in that there are fewer files to configure.

Debian/Ubuntu

Permanent Network Configuration

Configuring a static IP, mask and default gateway

```
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

root@sun:~#
```

Always need a loopback device on Linux systems

Setting a static IP address

Setting the default gateway

Debian/Ubuntu

Permanent Network Configuration

Configuring DNS Settings

```
root@sun:~# cat /etc/resolv.conf
search cabrillo.edu
nameserver 207.62.187.53
root@sun:~#
```

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

Same as the Red Hat family

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 Red Hat family

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

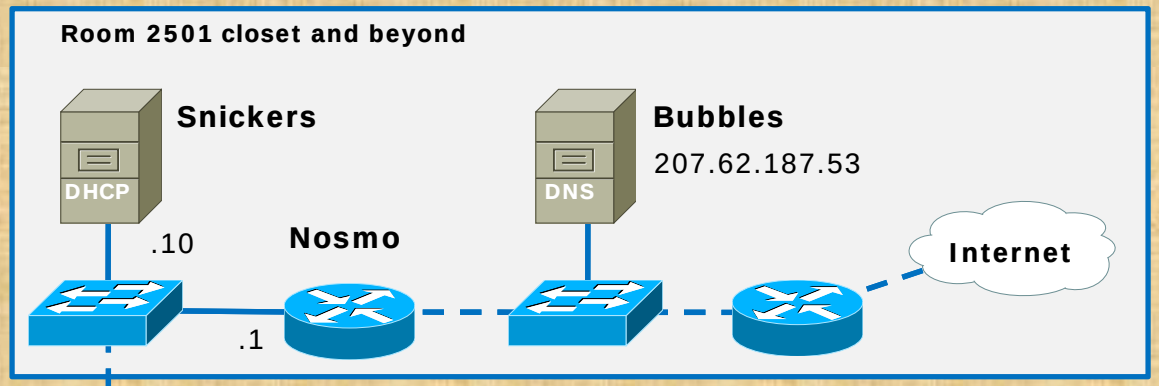
System hostname

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

To change the hostname, edit this file.

- *Use just the domain name, not the fully qualified domain name.*
- *The new name will take effect after the next system restart.*

Class Exercise – Configure Ubuntu interface settings permanently

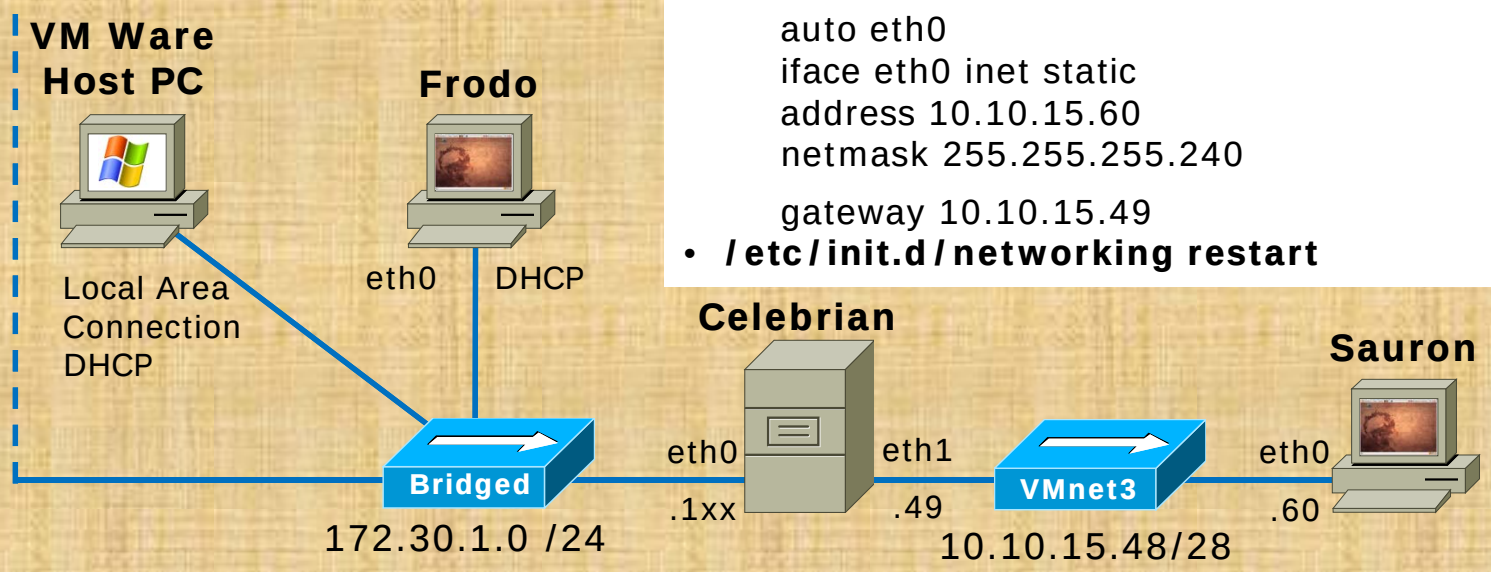


*Lets configure the **eth0** interface on Sauron*

- On Sauron, edit **/etc/network/interfaces** to be:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 10.10.15.60
netmask 255.255.255.240

gateway 10.10.15.49
```

- **/etc/init.d/networking restart**



Can Celebrian ping Frodo now?

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 lets looks at some of the details

Routers and the Network Layer (Layer 3)

Cisco commercial Router (IOS)

Linux Server with multiple NICs

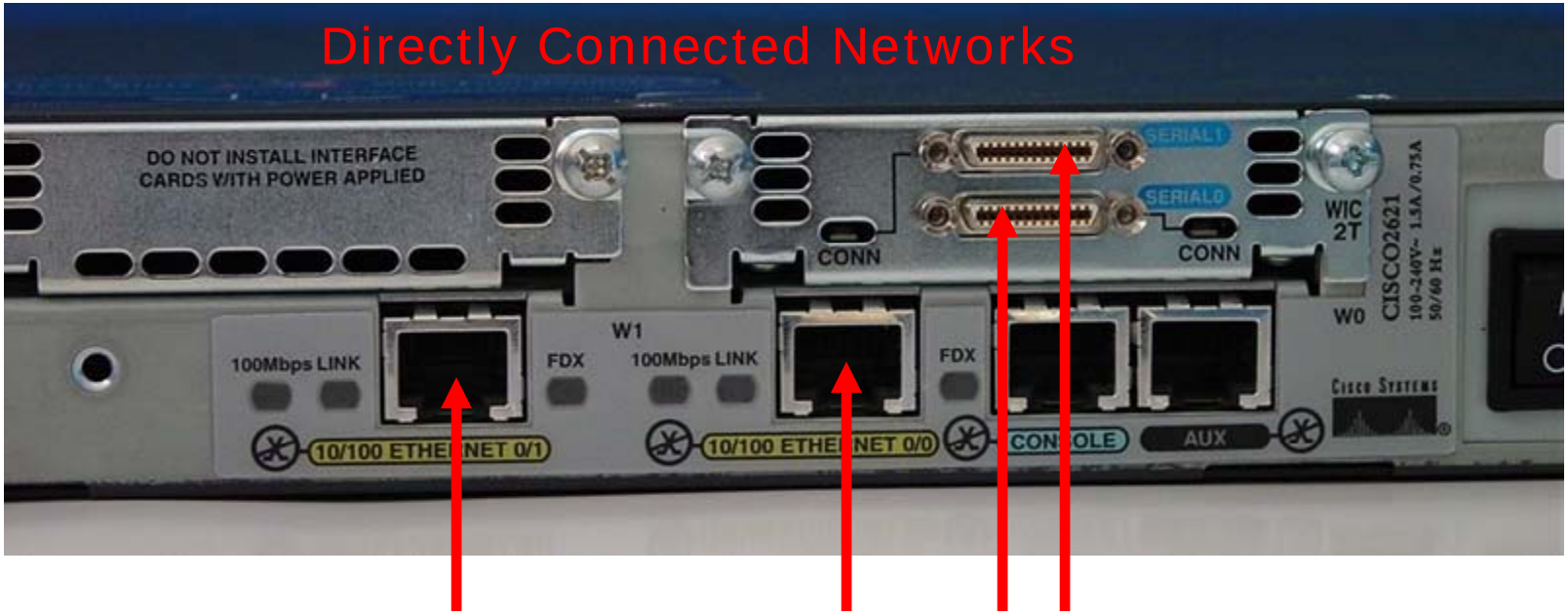
Linksys home Router

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.

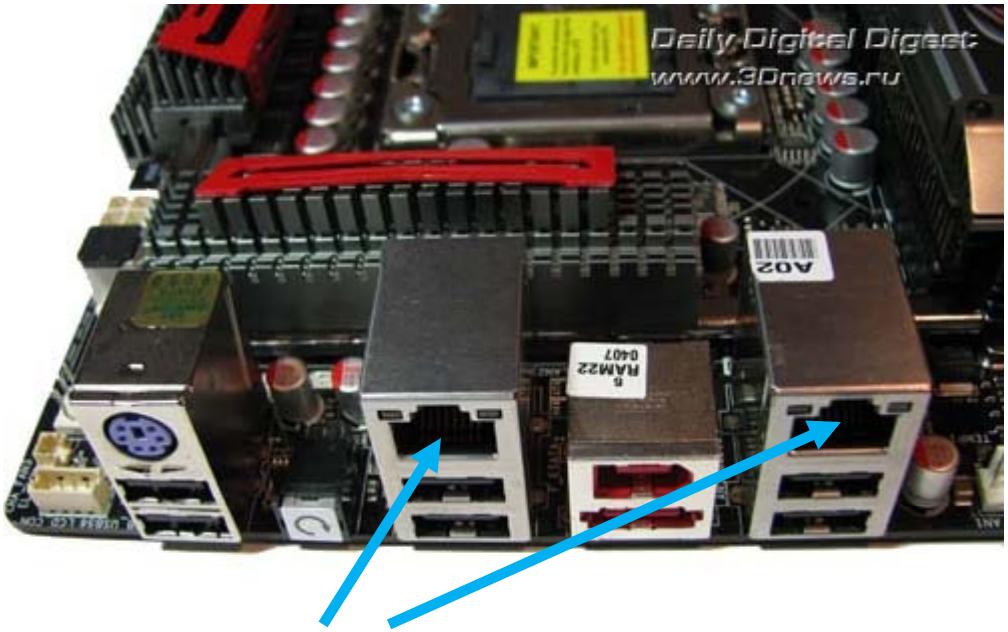


Each interface connects to a different network. Each interface has an IP address/mask for that network.

Routers and the Network Layer (Layer 3)

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

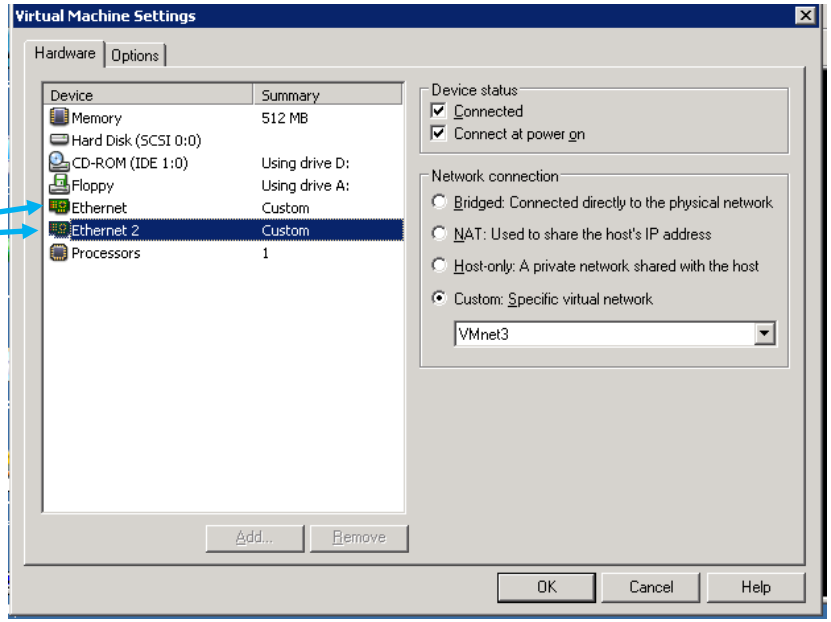
Each interface connects to a different network. Each interface has an IP address/mask for that network.

Routers and the Network Layer (Layer 3)

Virtual Linux routers

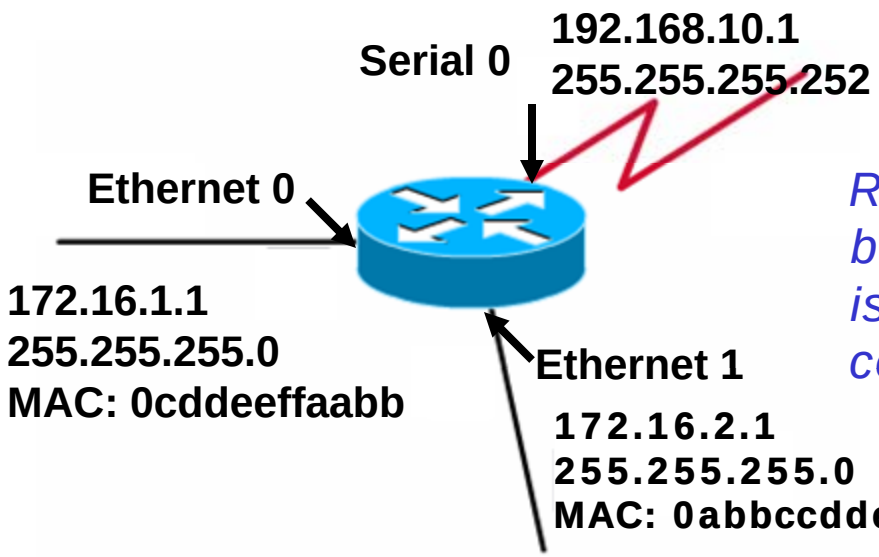
- Networking devices that make best path decisions (which interface to forward the IP packet) based in Layer 3 IP Destination Address.
- Virtual Linux routers connect multiple networks.

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



Directly Connected Networks

Routers belong to networks

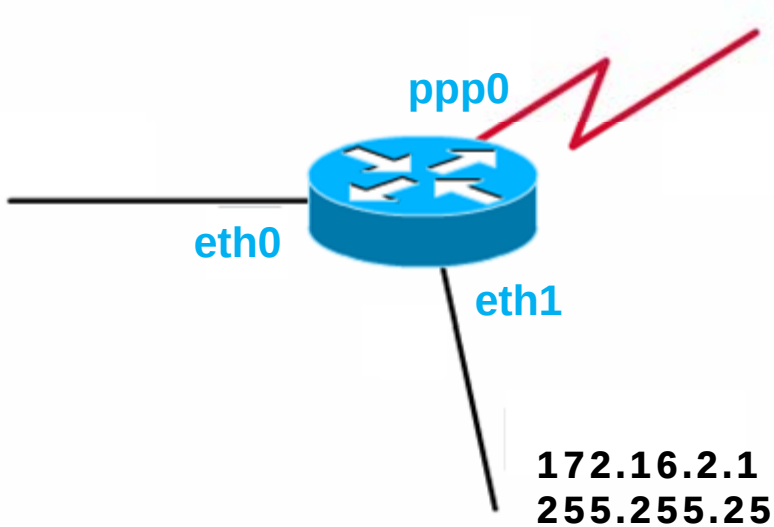


Router interfaces belonging to networks is a very important concept!

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



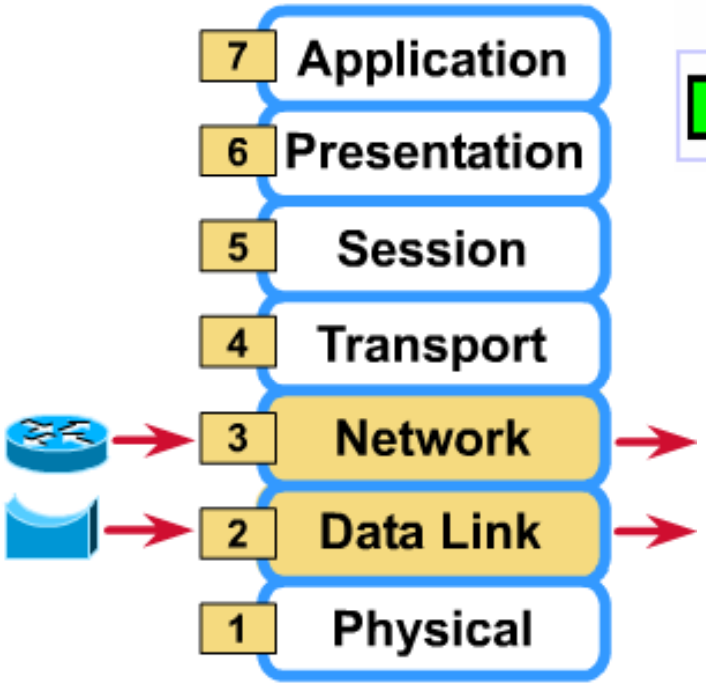
Same concept goes for Linux routers!

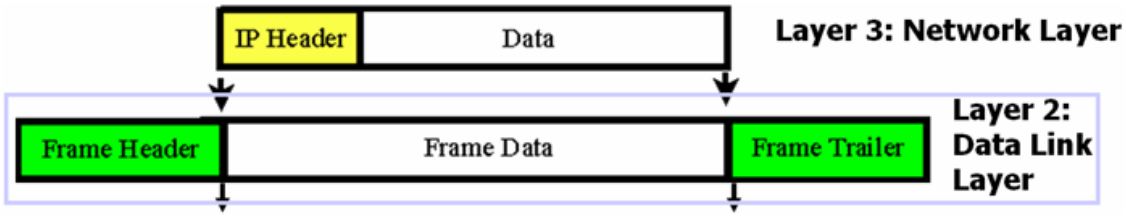
Directly Connected Networks

- 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.

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 **take the eth0 interface till you get to the 172.30.1.1 router and get more instructions there**

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

Routing Types

- 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.

Two types of routes … static or dynamic

Static

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

Dynamic

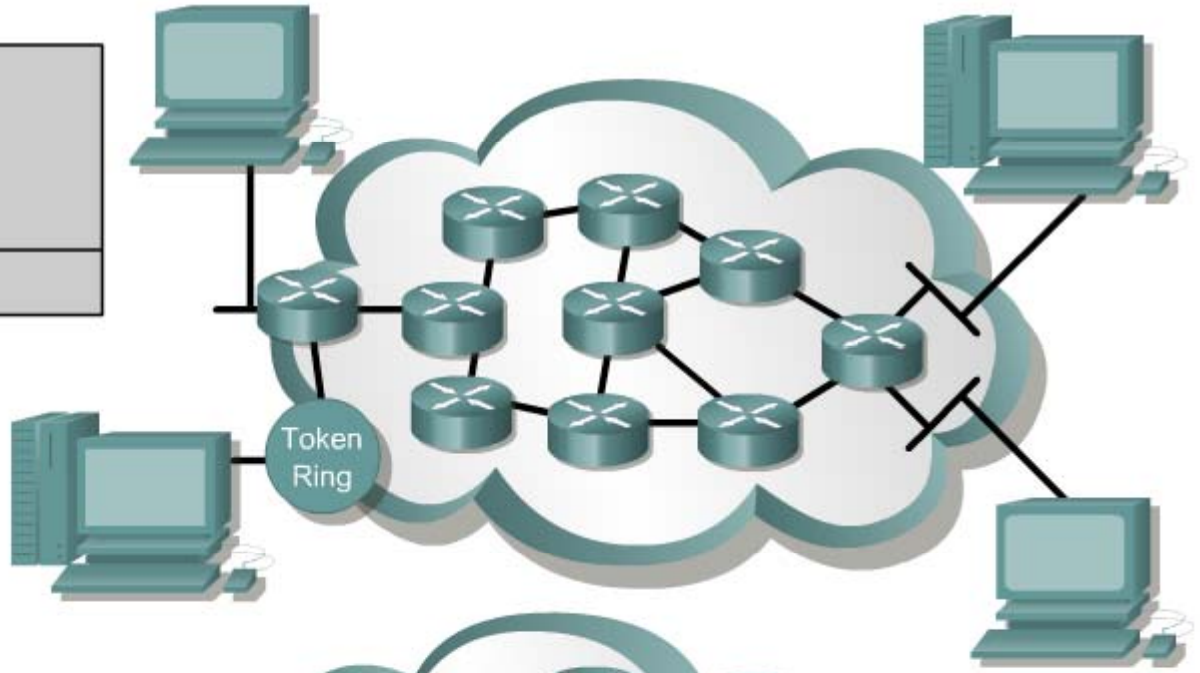
Uses a route that a routing protocol adjusts automatically for topology or traffic changes

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

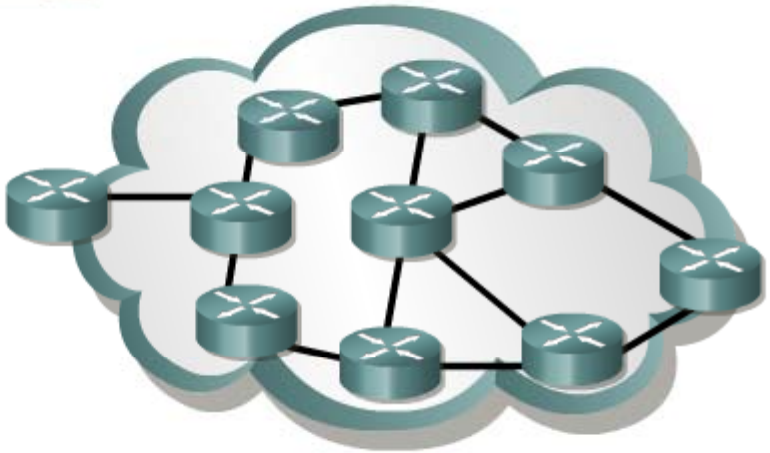
Routed Protocols vs. Routing Protocols

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

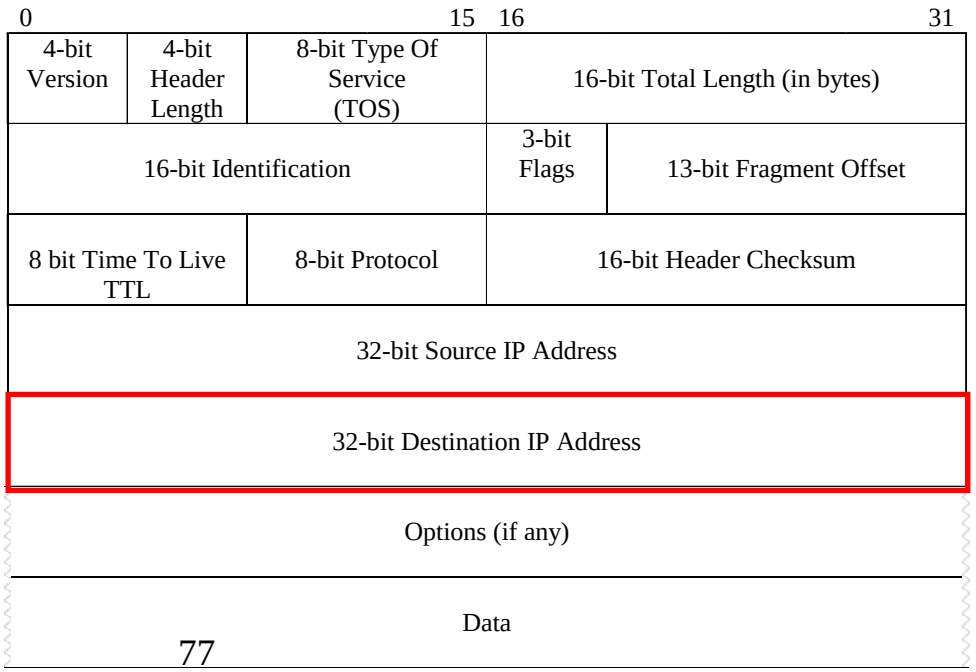
- IP is a routed protocol
- A routed protocol is a layer 3 protocol that contains network addressing information.
- This network addressing information is used by routers to determine the which interface, which next router, to forward this packet.

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!

Rick Graziani
graziani@cabrillo.edu

IP Header



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

Packet Forwarding

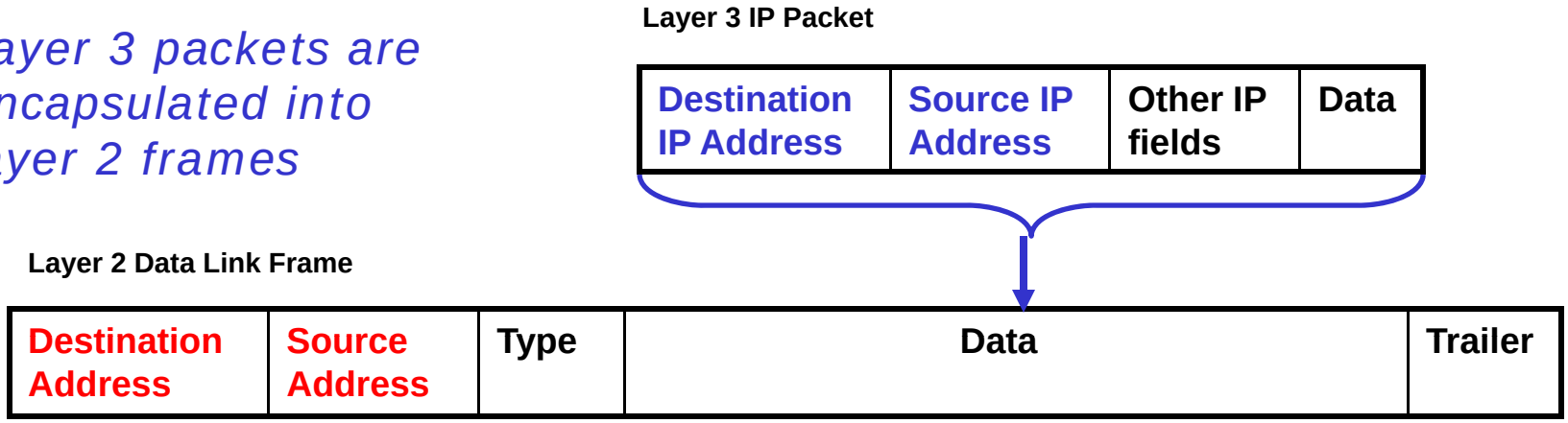
Packet Forwarding Summary

- 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 lets looks at some of the details

Encapsulation

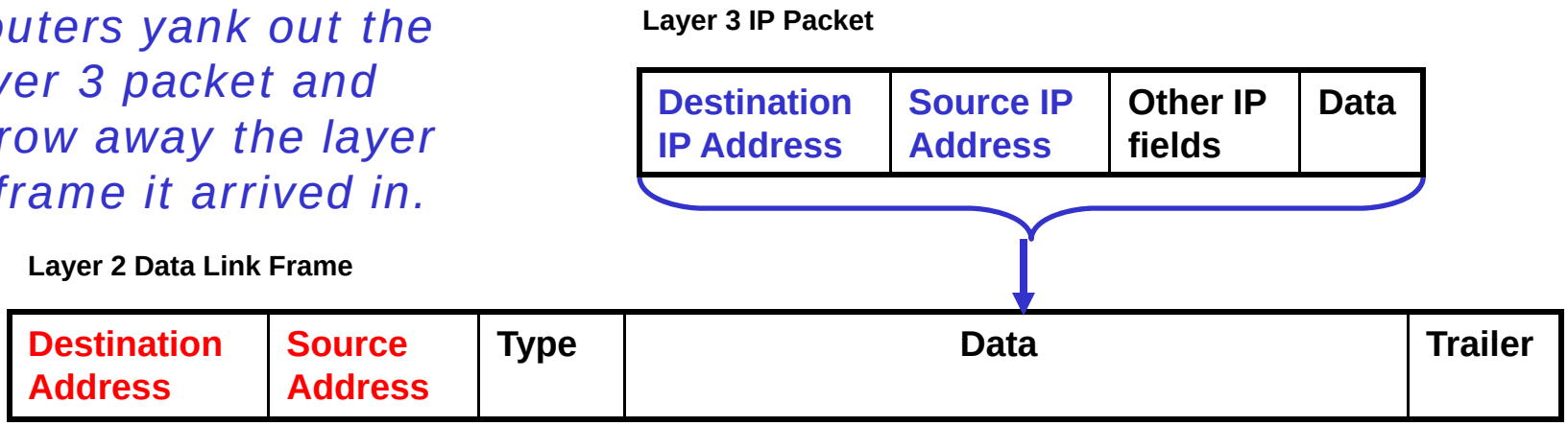
Layer 3 packets are encapsulated into layer 2 frames



- 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

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

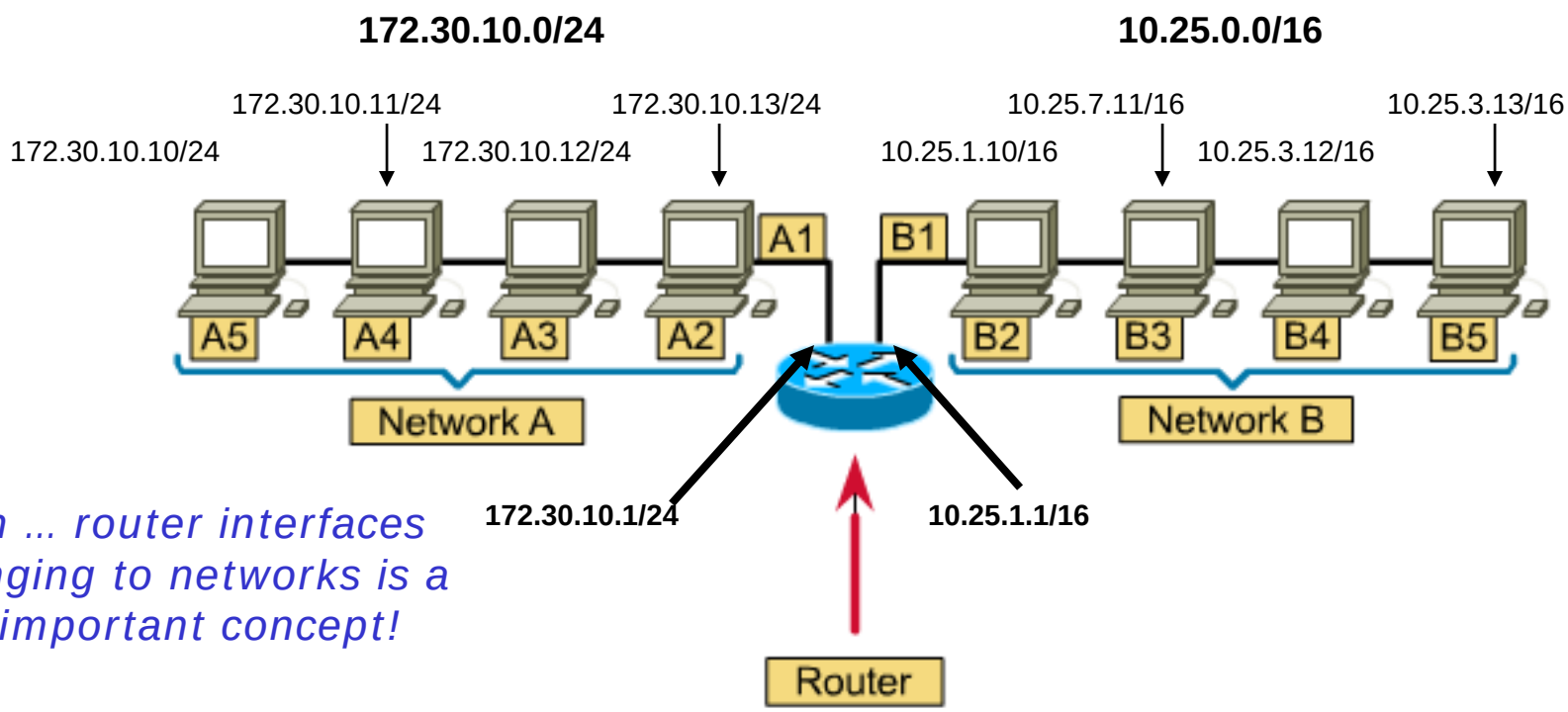
Destination Address	Source Address	Type	Data	Trailer

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 intensive purposes the rule remains the same, IP Addresses do NOT change.

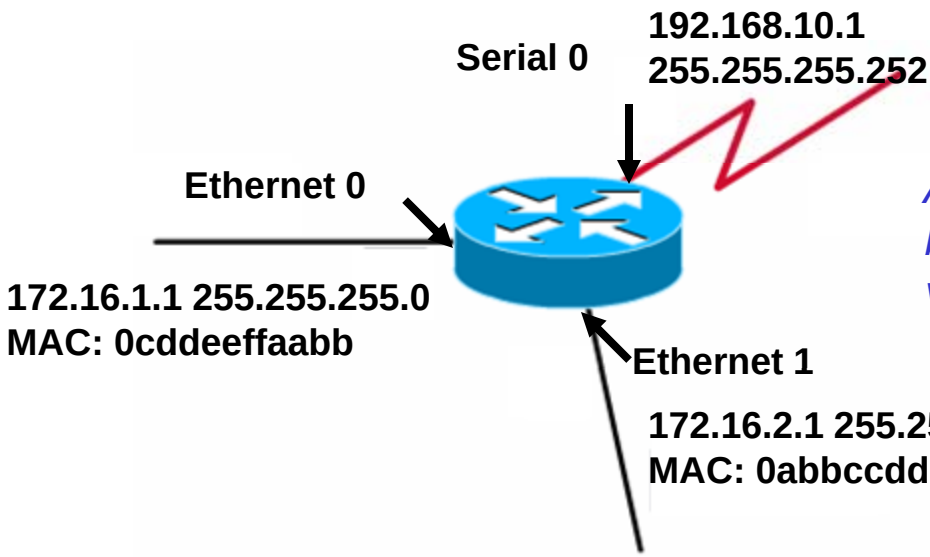
A router interface is a host on that network



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!

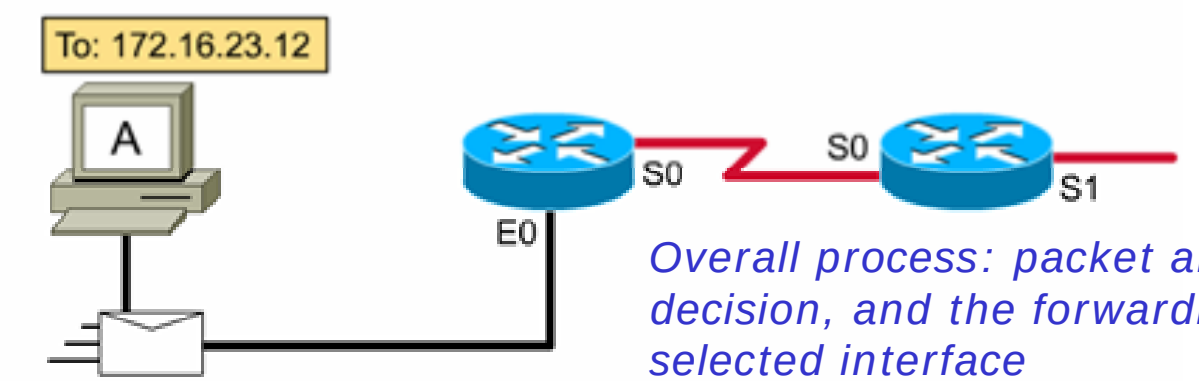
Routers belong to networks



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.

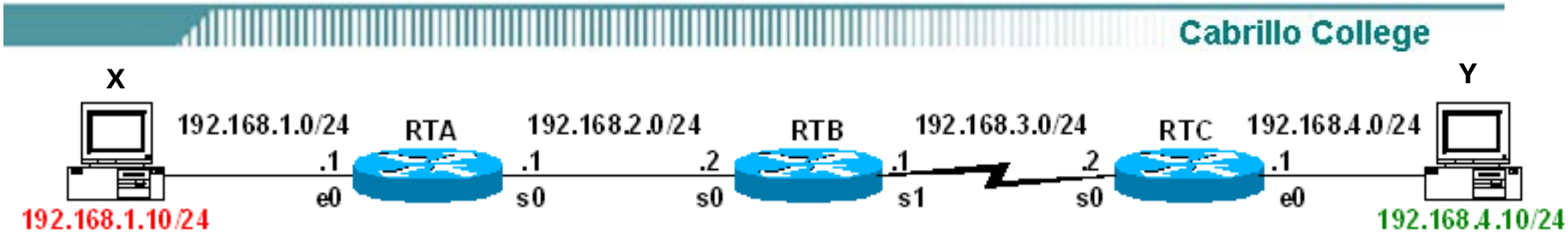
Router's Routing Table



Overall process: packet arrival, routing decision, and the forwarding out the selected interface

- The network layer provides best-effort end-to-end packet delivery across interconnected networks.
- Routers examine the Destination IP Address of a packet to determine where to send the packet next.
- After the router determines which path to use, it proceeds with forwarding the packet.
- It takes the packet that it accepted on one interface and forwards it to another interface or port that reflects the best path to the packet's destination.
- Much more information in the presentation on "The Routing Table Structure" (CIS 82 and CST 311)

Packet Forwarding



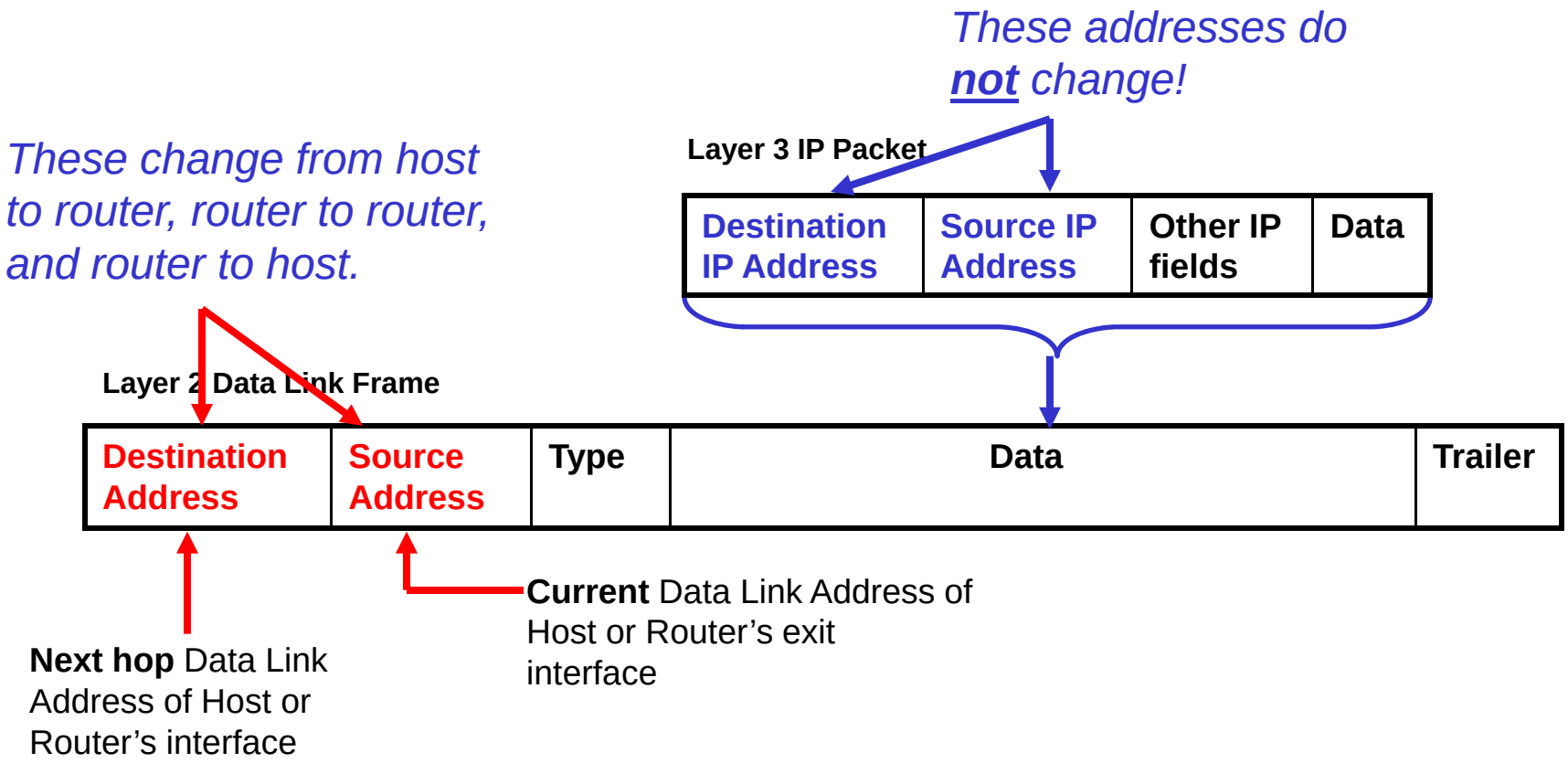
Let's do an example showing how a IP packet travels from network to network to get to its eventual destination.

Packet Forwarding

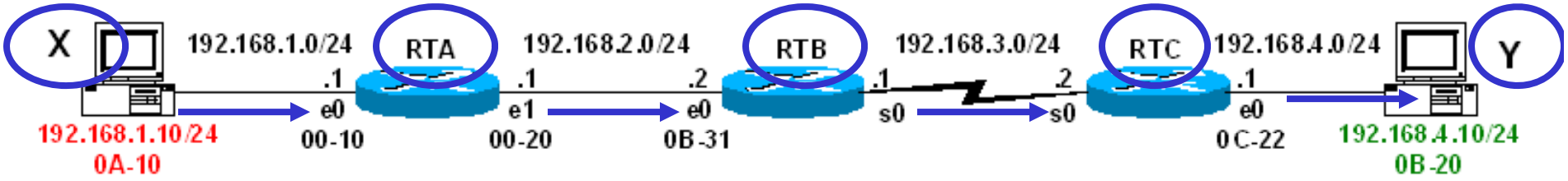
- Host X has a packet(s) to send to Host Y
- A router generally relays a packet from one data link to another, using two basic functions:
  1. a path determination function - **Routing**
  2. a switching function – **Packet Forwarding**

- 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…**



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**

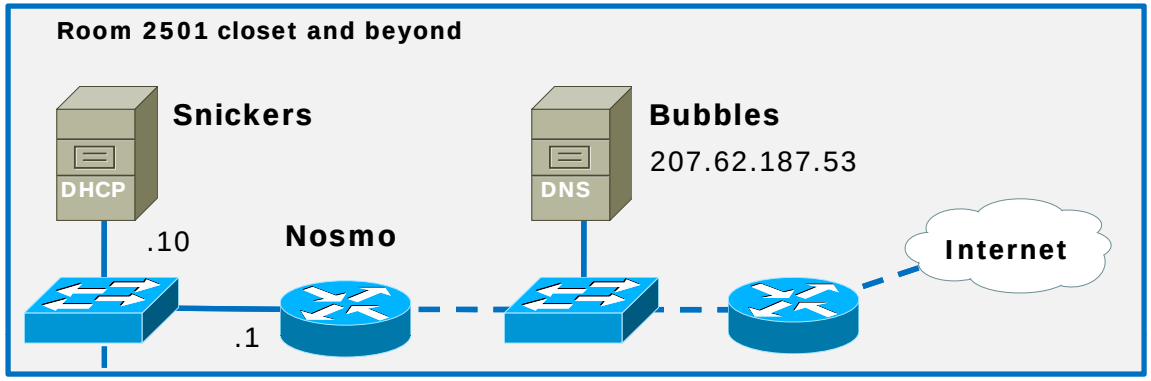
```
[root@elrond ~]# cat /etc/sysctl.conf
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# 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

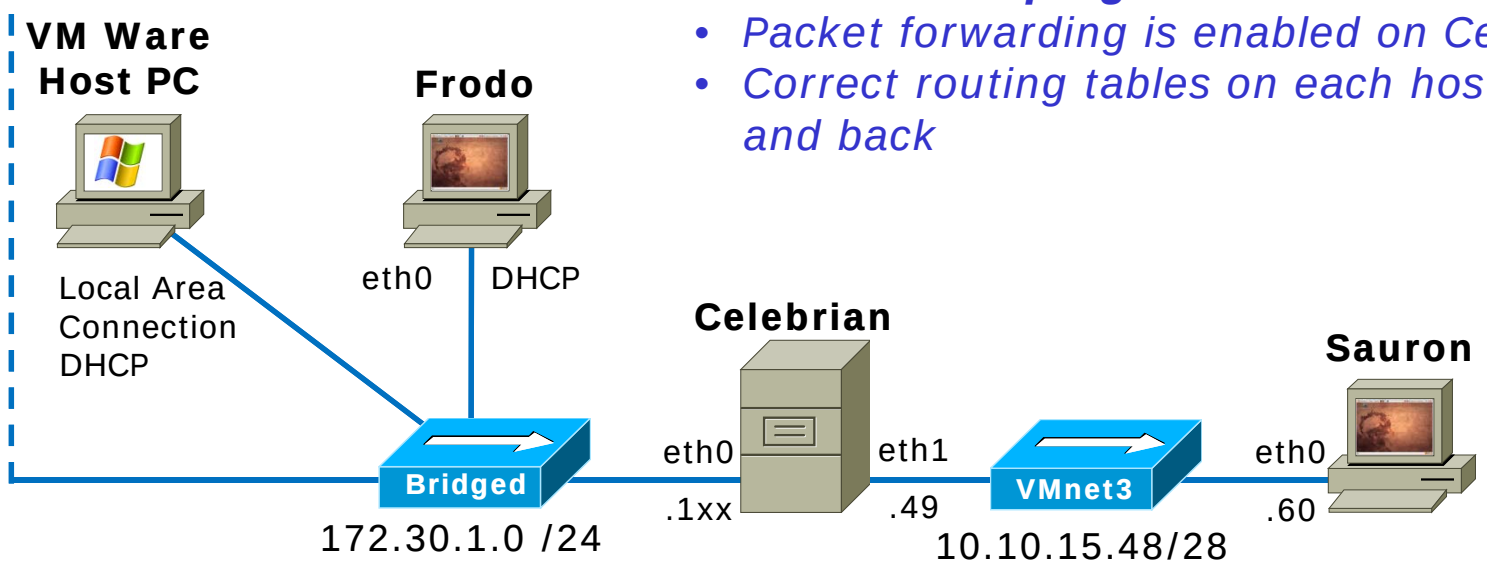
Packet Forwarding Summary

- 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 Celebrian*
- *Correct routing tables on each host out and back*



To ping Sauron (10.10.15.60) from Frodo (172.30.1.193) we need:

- Packet forwarding enabled on Celebrian (Lilly)
- Correct routing tables out and back for the ping packets

Frodo has no route to 10.10.15.48/28 network, and default gateway will send packets in the wrong direction! ☹

```
root@frodo:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         172.30.1.1      0.0.0.0         UG    0      0        0 eth0
```

Celebrian packet forwarding is not enabled! ☹

```
[root@lilly ~]# cat /proc/sys/net/ipv4/ip_forward
0
```

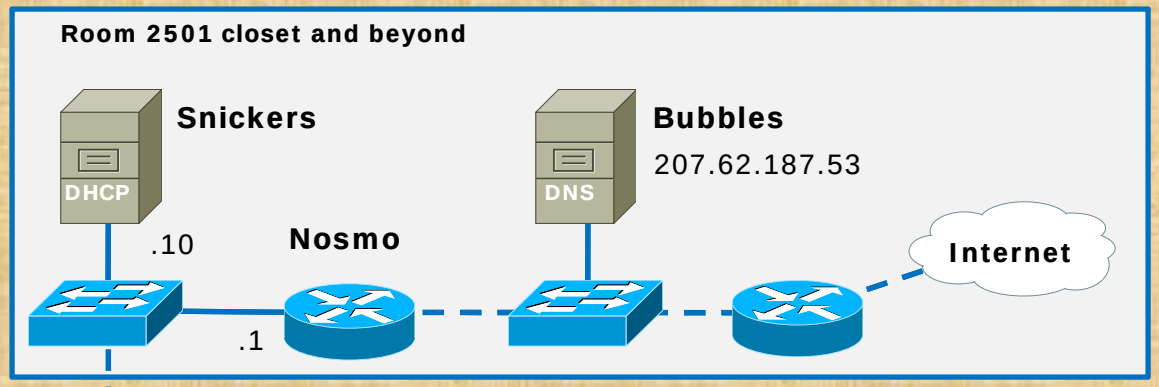
Celebrian routing table has needed network information ☺

```
[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
```

Sauron routing table does not have information on 172.30.1.0/24 network but default gateway will send packets in the right direction ☺

```
root@sauron:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.15.0      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     1000   0        0 eth0
0.0.0.0         10.10.15.49     0.0.0.0         UG    100    0        0 eth0
```

Class Exercise – packet forwarding

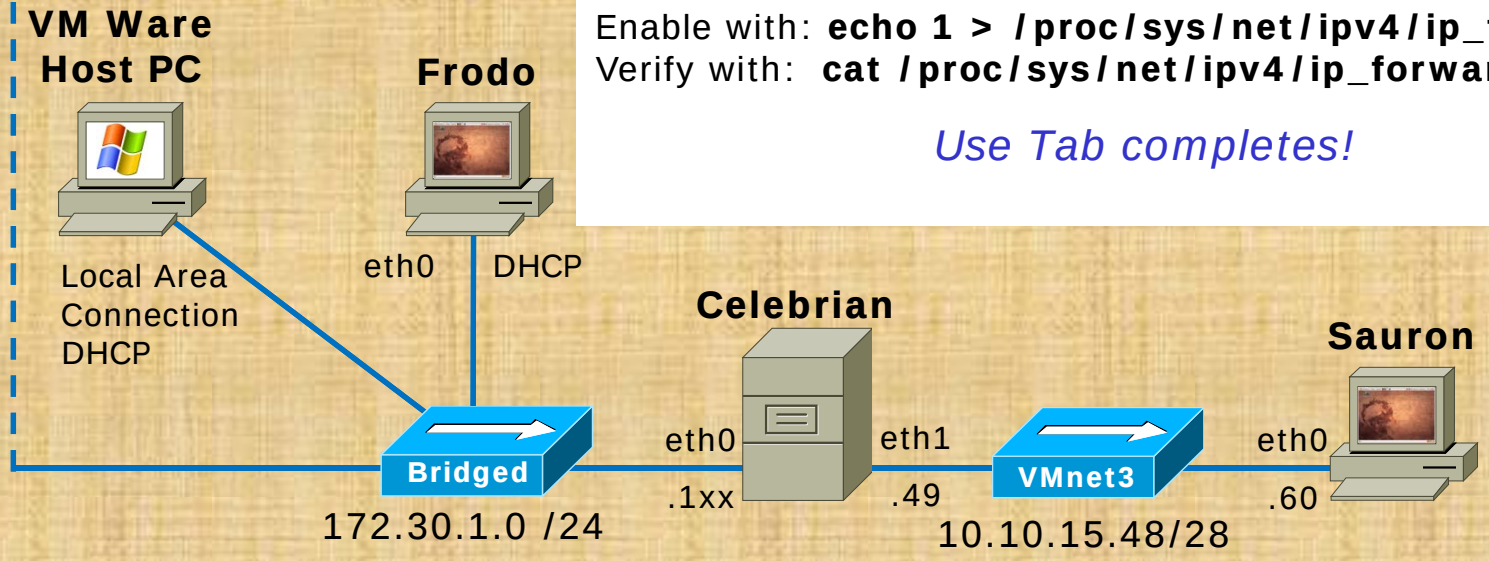


Enable packet forwarding on Celebrian (Lilly)

On the Celebrian VM:

Enable with: **echo 1 > /proc/sys/net/ipv4/ip_forward**
Verify with: **cat /proc/sys/net/ipv4/ip_forward**

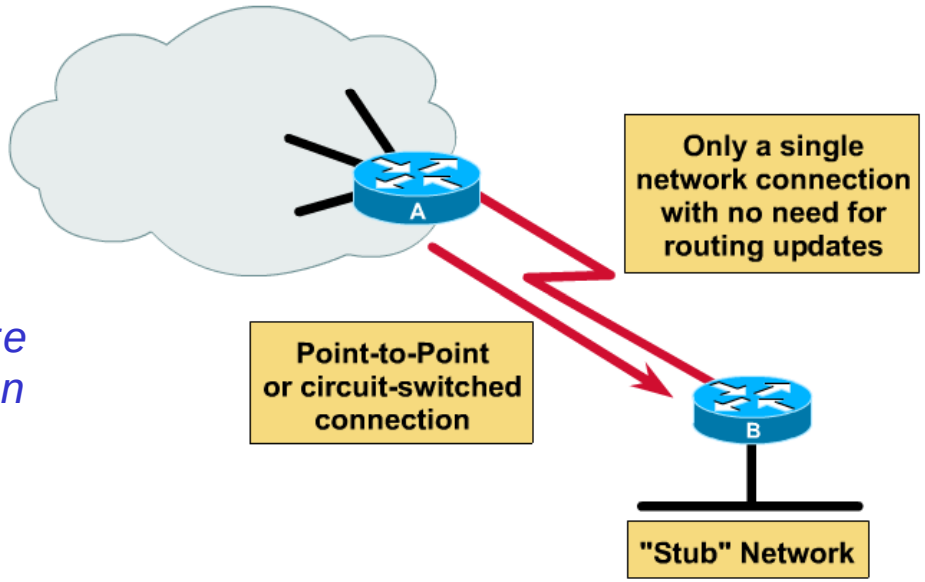
Use Tab completes!



Static Routes

Common uses for Static Routes

Static Routing Example

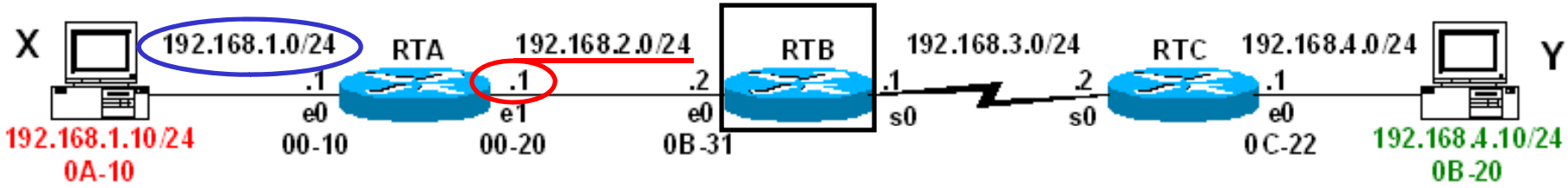


Static routes are very useful even in a dynamic routing world

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:**

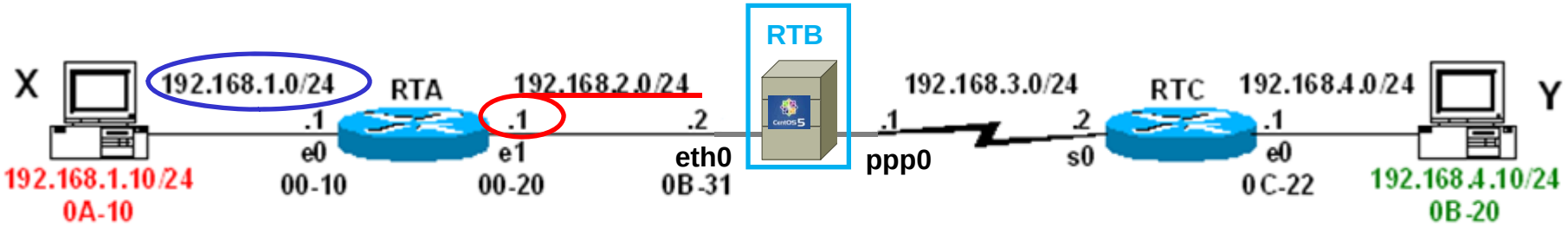
RTB(config)# ip route *network-address mask next-hop-ip-address*

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:

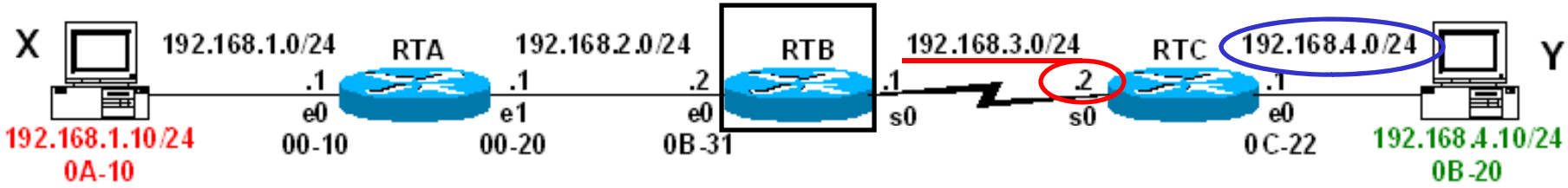
[root@RTB ~#] route add -net *network* *netmask* *mask* *gw* *next-hop*

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

[root@RTB ~#] route add –net 192.168.1.0 netmask 255.255.255.0 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

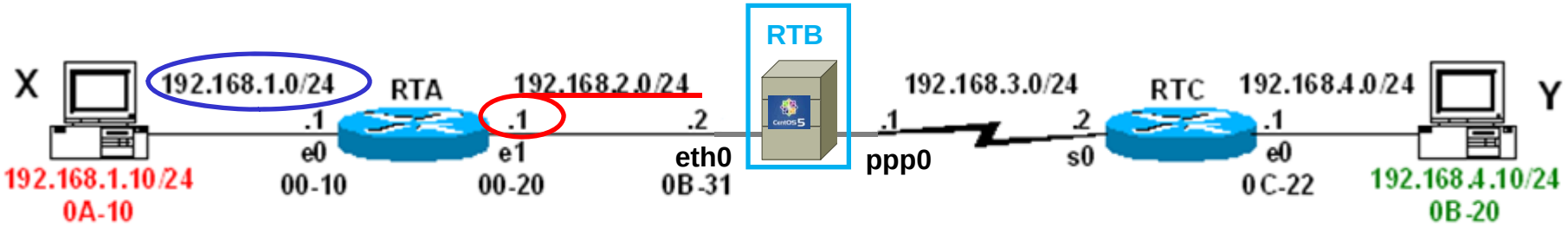


RTB(config)# ip route *network-address mask next-hop-ip-address*

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

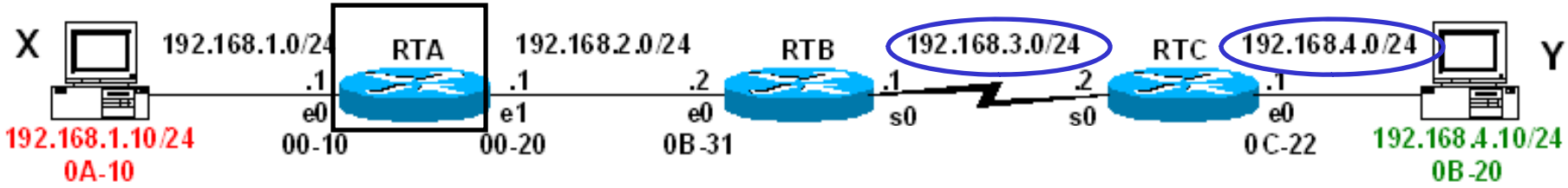


[root@RTB ~#] route add -net *network netmask mask gw next-hop*

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

[root@RTB ~#] route add –net 192.168.4.0 netmask 255.255.255.0 gw 192.168.3.2

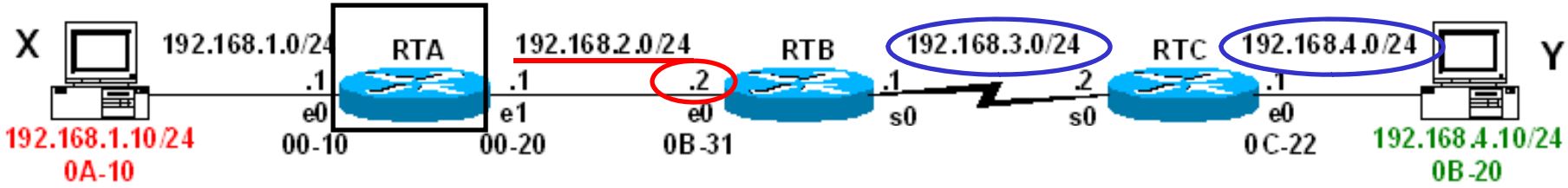
Static Route Examples



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



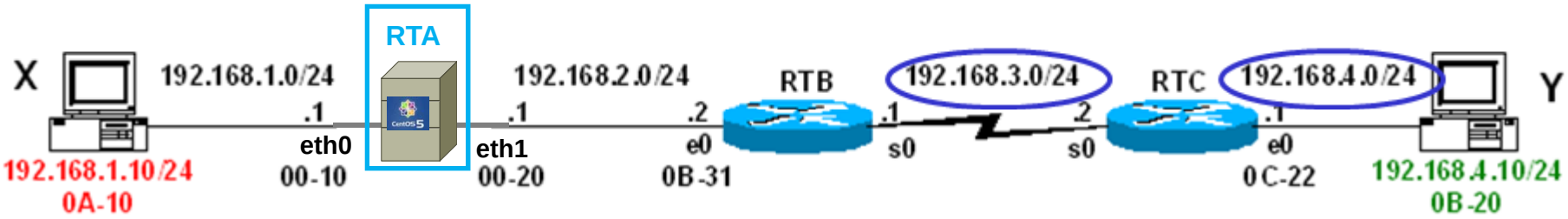
RTB(config)# ip route *network-address mask next-hop-ip-address*

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

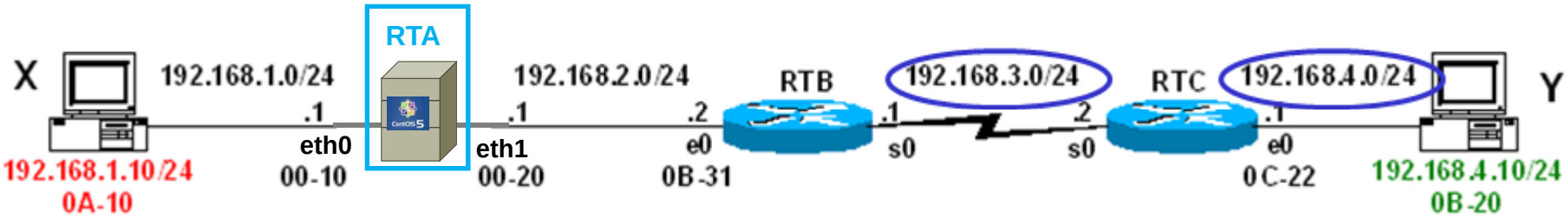
Static Route Examples



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* *netmask* *mask* *gw* *next-hop*

Static Route Examples



[root@RTA ~#] route add -net *network* *netmask* *mask* *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 netmask 255.255.255.0 gw 192.168.2.2
[root@RTA ~#] route add –net 192.168.4.0 netmask 255.255.255.0 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***

```
[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***

alternate syntax

```
[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
```

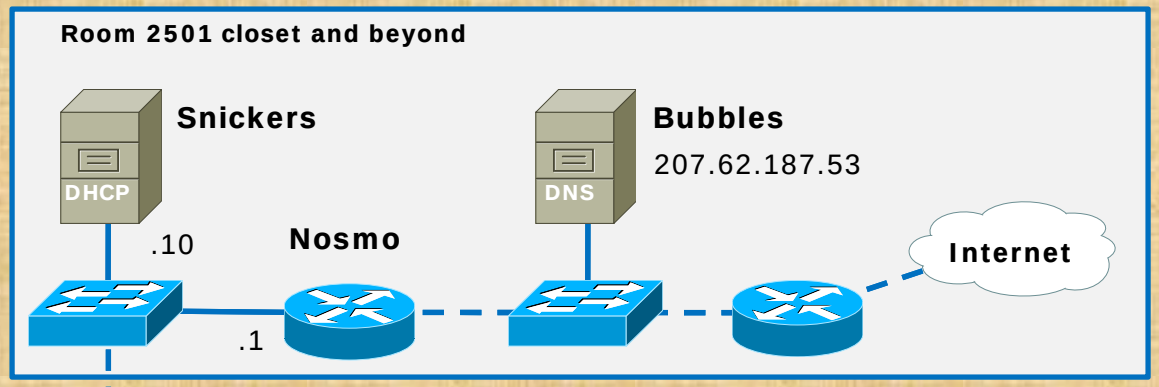
Debian/Ubuntu
Permanent NIC Configuration

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:~#
```

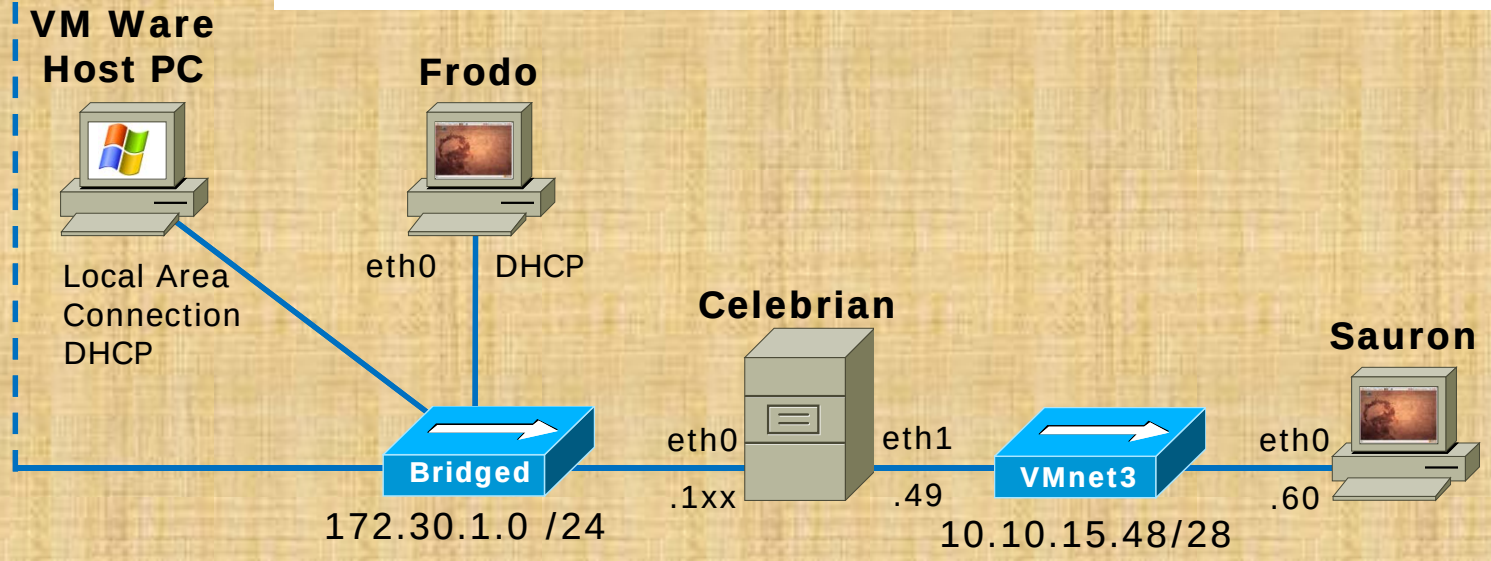
Use up and down to configure what will happen when interface is brought up or down

Class Exercise – static routes



Add a static route to Frodo's routing table for the 10.10.15.48/28 network

On Frodo:
route add -net 10.10.15.48 netmask 255.255.255.240 gw 172.30.1.1xx
Verify with: **route -n**



Can Frodo ping Sauron now? It should!

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

-n shows IP addresses instead of names (faster)

```
[root@elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.4.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.123   255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.4.1      0.0.0.0         UG    0      0        0 eth0
[root@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@elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.4.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.123   255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.4.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond ~]#
```

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

The Routing Table

Routing Table

```
[root@elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.4.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.123   255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.4.1      0.0.0.0         UG    0      0        0 eth0
[root@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@elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.4.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.123   255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.4.1      0.0.0.0         UG    0      0        0 eth0
[root@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@elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.4.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.123   255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.4.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond ~]#
```

Possible flags include:

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

The Routing Table

Routing Table

```
[root@elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.4.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.123   255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.4.1      0.0.0.0         UG    0      0        0 eth0
[root@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@elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.4.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.123   255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.4.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond ~]#
```

Ref:
Number of references to this route.
(Not used in the Linux kernel.)

The Routing Table

Routing Table

```
[root@elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.4.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.123   255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.4.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond ~]#
```

Use: Count of lookups for the route when using the –C option (show cache based information)

The Routing Table

Routing Table

```
[root@elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.4.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.123   255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.4.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond ~]#
```

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

The Routing Table
Supernetting

Routing Table

```
root@frodo:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.3.0     172.30.1.125    255.255.255.0   UG    0      0        0 eth0
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.2.0     172.30.1.125    255.255.255.0   UG    0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth0
0.0.0.0         172.30.1.1      0.0.0.0         UG    100    0        0 eth0
root@frodo:~#
```

*Note: these two routes could be replaced with a single route for **192.168.0.0 /16**. This is super-netting (the reverse of sub-netting)*

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 number 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

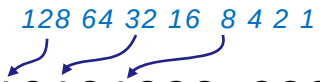
Network Number

Compute the network number

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

Example: 192.168.3.200 with genmask 255.255.255.0 is 192.168.3.0

- By hand




```
110000 10101000 00000011 11001000  192.168.3.200
111111 11111111 11111111 00000000  255.255.255.0
110000 10101000 00000011 00000000  192.168.3.0
```

- With ipcalc

```
[root@elrond ~]# ipcalc -n 192.168.3.200 255.255.255.0
NETWORK=192.168.3.0
```

The Routing Algorithm

Network Number

Compute the network number

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

Example: 192.168.30.100 with genmask 255.255.240.0 is 192.168.16.0

- By hand


```
110000 10101000 00011110 01100100  192.168.30.100
111111 11111111 11110000 00000000  255.255.240.0
110000 10101000 00010000 00000000  192.168.16.0
```

- With ipcalc

```
[root@elrond ~]# ipcalc -n 192.168.30.100 255.255.240.0
NETWORK=192.168.16.0
[root@elrond ~]#
```

route command –n option

show route table with names

```
[root@elrond ~]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.4.0      *               255.255.255.0   U     0      0        0 eth0
192.168.3.0     legolas         255.255.255.0   UG    0      0        0 eth1
192.168.2.0     *               255.255.255.0   U     0      0        0 eth1
169.254.0.0     *               255.255.0.0     U     0      0        0 eth1
default         nosmo           0.0.0.0         UG    0      0        0 eth0
```

show route table with IP addresses

```
[root@elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.4.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.123   255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.4.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond ~]#
```

route command for viewing cache

show route table cache with names

```
[root@elrond ~]# route -C
Kernel IP routing cache
Source          Destination     Gateway         Flags Metric Ref    Use Iface
192.168.2.125   sauron          legolas                0      0        0 eth1
172.30.4.125    nosmo           nosmo                  0      0        0 eth0
172.30.4.125    nosmo           nosmo                  0      0        6 eth0
sauron          192.168.2.125   192.168.2.125   l      0      0        1 lo
frodo           172.30.4.125    172.30.4.125    il     0      0        1 lo
172.30.4.108    172.30.4.255    172.30.4.255    ibl    0      0        0 lo
172.30.4.103    172.30.4.125    172.30.4.125    il     0      0      105 lo
nosmo           172.30.4.125    172.30.4.125    il     0      0        5 lo
172.30.4.125    172.30.4.103    172.30.4.103           0      1        0 eth0
legolas         192.168.2.125   192.168.2.125   il     0      0        0 lo
172.30.4.125    frodo           frodo                  0      0        0 eth0
172.30.4.125    frodo           frodo                  0      0        1 eth0
172.30.4.10     172.30.4.255    172.30.4.255    ibl    0      0       10 lo
192.168.2.125   sauron          legolas                0      0        2 eth1
172.30.4.12     255.255.255.255 255.255.255.255 ibl    0      0        3 lo
172.30.4.10     172.30.4.255    172.30.4.255    ibl    0      0       10 lo
192.168.2.125   sauron          legolas                0      0        2 eth1
172.30.4.12     255.255.255.255 255.255.255.255 ibl    0      0        3 lo
[root@elrond ~]#
```

route command for viewing cache

show route table cache with IP addresses

```
[root@elrond ~]# route -Cn
Kernel IP routing cache
Source          Destination     Gateway         Flags Metric Ref    Use Iface
192.168.2.125   192.168.3.200   192.168.2.123           0      0        0 eth1
172.30.4.125    172.30.4.1      172.30.4.1              0      0        0 eth0
172.30.4.125    172.30.4.1      172.30.4.1              0      0        6 eth0
192.168.3.200   192.168.2.125   192.168.2.125   l       0      0        1 lo
172.30.4.150    172.30.4.125    172.30.4.125    il      0      0        1 lo
172.30.4.108    172.30.4.255    172.30.4.255    ibl     0      0        0 lo
172.30.4.103    172.30.4.125    172.30.4.125    il      0      0      119 lo
172.30.4.125    207.62.187.53   172.30.4.1              0      0        7 eth0
172.30.4.1      172.30.4.125    172.30.4.125    il      0      0        5 lo
172.30.4.106    172.30.4.255    172.30.4.255    ibl     0      0        0 lo
172.30.4.110    172.30.4.255    172.30.4.255    ibl     0      0        0 lo
207.62.187.53   172.30.4.125    172.30.4.125    l       0      0        7 lo
172.30.4.125    172.30.4.103    172.30.4.103            0      1        0 eth0
192.168.2.123   192.168.2.125   192.168.2.125   il      0      0        0 lo
172.30.4.125    172.30.4.150    172.30.4.150            0      0        0 eth0
172.30.4.125    207.62.187.53   172.30.4.1              0      0        7 eth0
172.30.4.125    172.30.4.150    172.30.4.150            0      0        1 eth0
172.30.4.10     172.30.4.255    172.30.4.255    ibl     0      0       14 lo
192.168.2.125   192.168.3.200   192.168.2.123           0      0        2 eth1
172.30.4.12     255.255.255.255 255.255.255.255 ibl     0      0        5 lo
[root@elrond ~]#
```

route command
flushing the cache

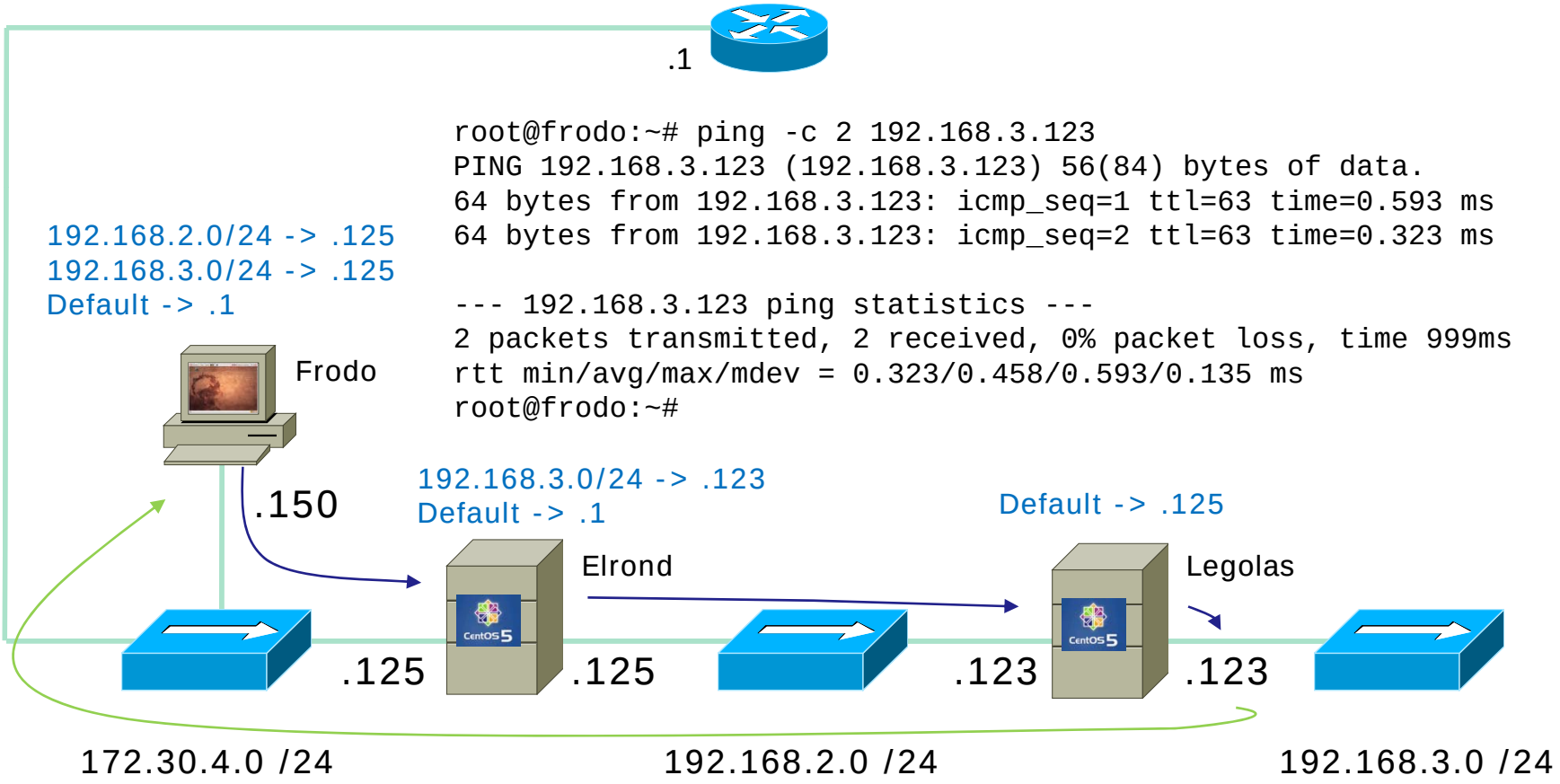
Flush the route cache

```
[root@elrond ~]# ip route flush cache
[root@elrond ~]# route -C
Kernel IP routing cache
Source          Destination     Gateway         Flags Metric Ref    Use Iface
172.30.4.103    172.30.4.125    172.30.4.125    il    0      0        3 lo
172.30.4.125    172.30.4.103    172.30.4.103          0      1        0 eth0
buttercup.cabri 172.30.4.125    172.30.4.125    l     0      0        1 lo
172.30.4.103    172.30.4.125    172.30.4.125    il    0      0        4 lo
172.30.4.125    172.30.4.103    172.30.4.103          0      1        0 eth0
[root@elrond ~]#
```

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

ICMP Redirect

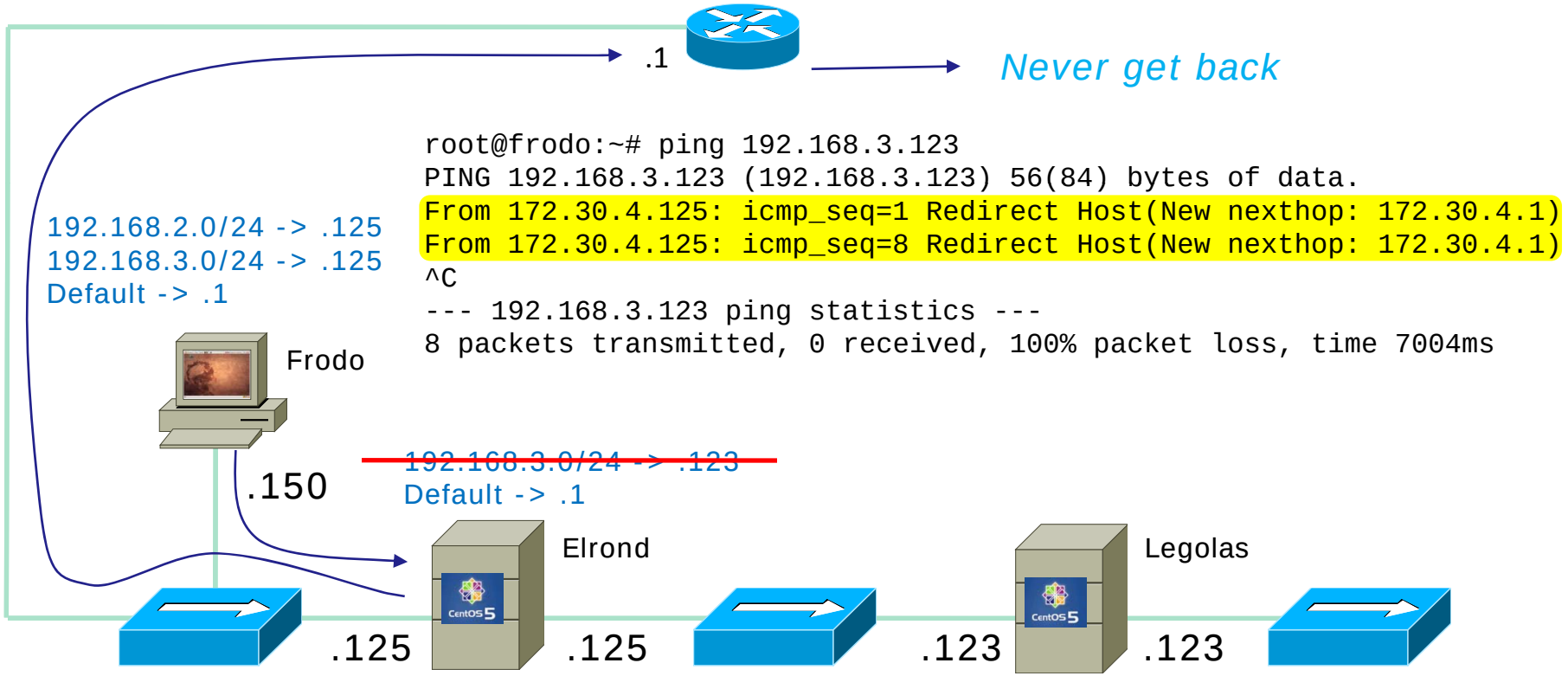
Routers will updates each others caches when they discover an inefficient route



When doing Lab 3, what happens if we left off the static route on Elrond?

ICMP Redirect

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



Elrond tells Frodo there is a shorter route

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.

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 or does not exist. 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 207.62.187.53
PING 207.62.187.53 (207.62.187.53) 56(84) bytes of data.

--- 207.62.187.53 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!

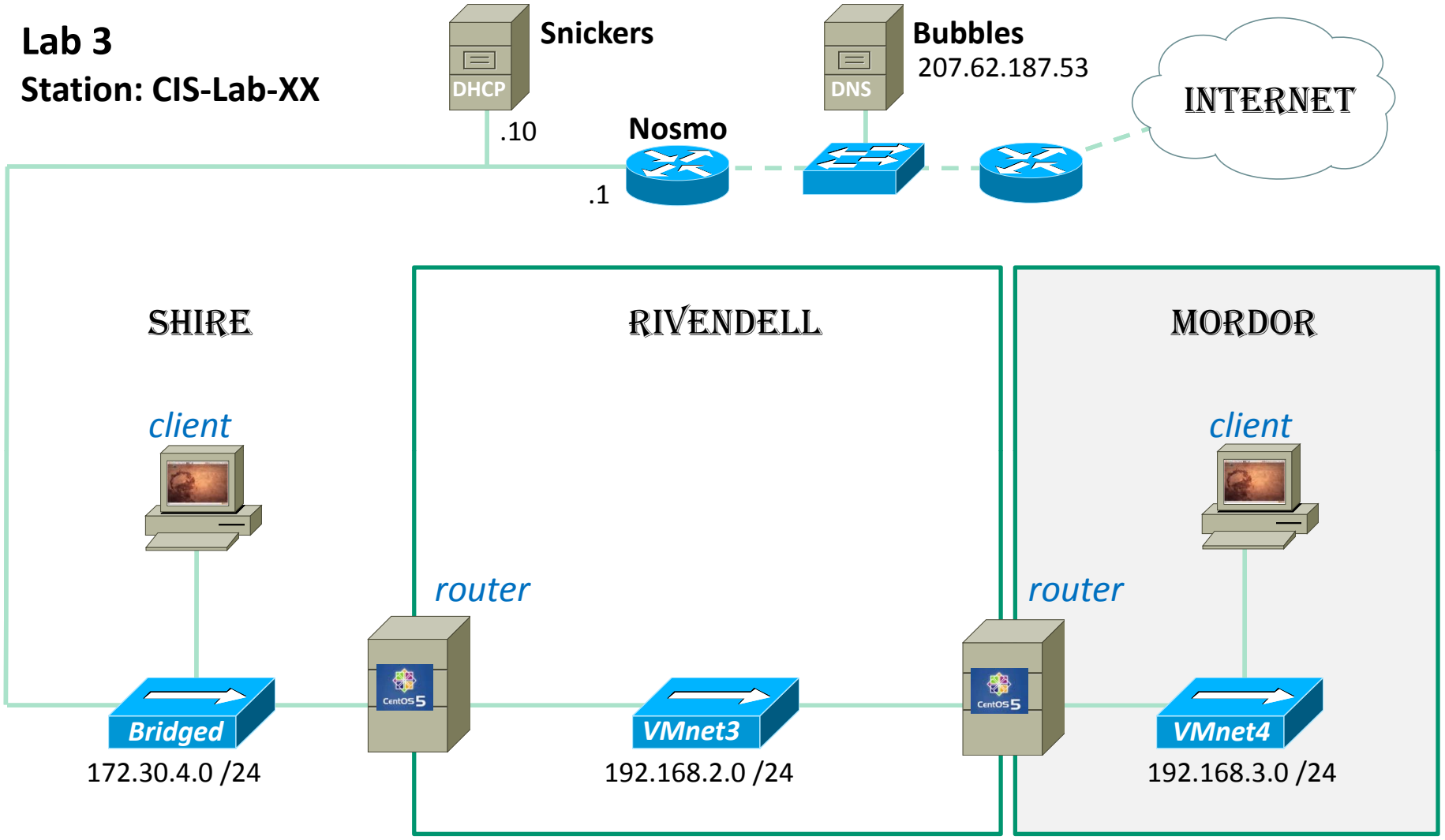
Lab

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

Three overall steps:

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

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


Lab 3
Station: CIS-Lab-XX
Snickers
DHCP
.10
Nosmo
.1
Bubbles
DNS
207.62.187.53
INTERNET
SHIRE
RIVENDELL
MORDOR
client
client
router
router
Bridged
172.30.4.0 /24
CentOS 5
VMnet3
192.168.2.0 /24
CentOS 5
VMnet4
192.168.3.0 /24

Some essentials for doing labs

- Becoming root:
  - sudo *command*
  - su –

The "–" is very important as this gets you root's environment

- To try again for a DHCP address: dhclient

If Frodo's DHCP interface fails to get an IP address after booting up use this command

- Use Google to research error messages
  - Google *network is unreachable*

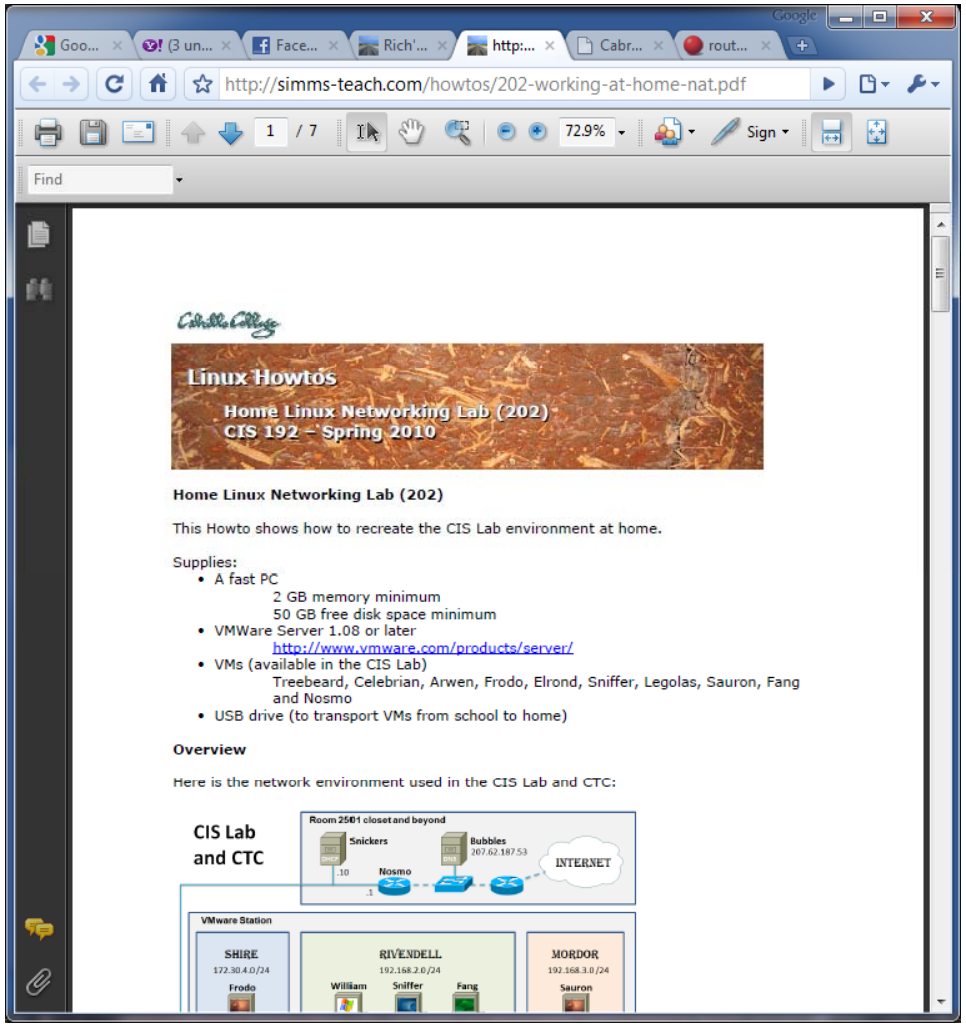
You will need to login as root to do most labs.
Be careful as root can do anything !!

Some essentials for doing labs

The "I've tried everything and it still won't work" problem

- Use the forum to ask questions and to clarify things
- Review Lesson Powerpoints which usually have examples aimed at doing the lab assignments
- Make a network diagram with all interfaces labeled. Confirm your configuration matches the diagram.
- Go back and methodically verify each step was completed. For example, if you modified /etc/hosts then cat it out and review your changes. If you set the default gateway, use route –n command to verify. If you configured an IP address, use ifconfig to verify.
- If your VM is completely "hosed": Use **Revert to snapshot** to restore to a pristine version.

Home Network



Except for masquerading you have learned just about everything you need to configure the Nosmo VM to work in your home now.

Wrap

New commands, tools and services:

- chkconfig
- dhclient
- ip
- route

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
- service network restart

New Files and Directories (Ubuntu):

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

VMware:

Next Class

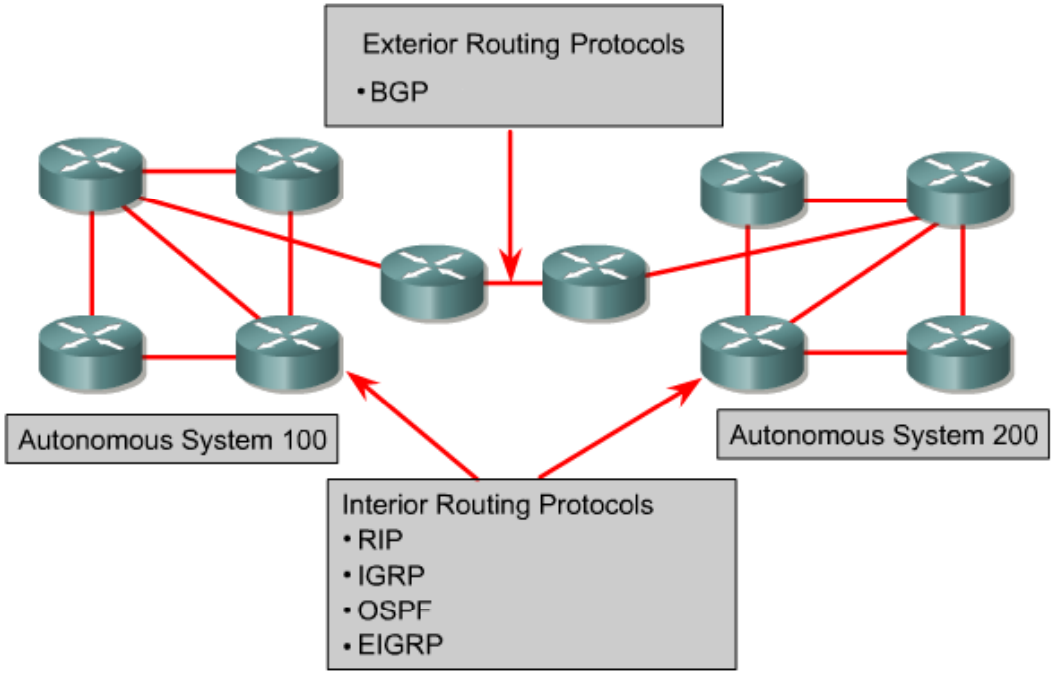
Assignment: Check Calendar Page
http://simms-teach.com/cis192calendar.php

Quiz questions for next class:

- If frodo has IP address 172.30.4.193 what line would be added to elrond's /etc/hosts file so elrond users could ping frodo by name?
- If two routes in the routing table match a destination IP address, which route is chosen – the one with the shorter or longer prefix?
- What does a router do with an incoming packet that has a destination IP address that matches no entries in the routing table?

Backup

Routing Protocols



- **RIP** – A distance vector interior routing protocol
- **IGRP** – Cisco's distance vector interior routing protocol
- **OSPF and IS-IS** – A link-state interior routing protocol
- **EIGRP** – Cisco's advanced distance vector interior routing protocol
- **BGP** – A distance vector exterior routing protocol

Rick Graziani
graziani@cabrillo.edu

Routing Protocols – CIS 82 / CST 312

Routing Information Protocol (RIP) was originally specified in RFC 1058.

- It is a **distance vector** routing protocol.
- **Hop count** is used as the metric for path selection.
- If the hop count is **greater than 15, the packet is discarded**.
- Routing updates are broadcast **every 30 seconds**, by default.

Interior Gateway Routing Protocol (IGRP) is a proprietary protocol developed by Cisco.

- It is a **distance vector** routing protocol.
- **Bandwidth, load, delay and reliability** are used to create a composite metric.
- Routing updates are broadcast **every 90 seconds**, by default.

EIGRP is a Cisco proprietary enhanced distance vector routing protocol.

- It is an **enhanced distance vector routing protocol**.
- Uses **unequal-cost and equal-cost** load balancing.
- Uses a combination of distance vector and link-state features.
- Uses **Diffused Update Algorithm (DUAL)** to calculate the shortest path.

Routing Protocols – CIS 82 / CST 312

Open Shortest Path First (OSPF) is a nonproprietary link-state routing protocol.

- It is a **link-state** routing protocol.
- **Open standard** routing protocol described in RFC 2328.
- Uses the **SPF algorithm** to calculate the lowest cost to a destination.
- **Routing updates are flooded** as topology changes occur.

Intermediate System to Intermediate System (IS-IS)

- IS-IS is an Open System Interconnection (OSI) routing protocol originally specified by International Organization for Standardization (ISO) 10589.
- It is a **link-state** routing protocol.

Border Gateway Protocol (BGP) is an exterior routing protocol.

- It is a **distance vector** (or path vector) exterior routing protocol
- Used between **ISPs or ISPs and clients**.
- Used to **route Internet traffic** between autonomous systems.

New commands, tools and services:

arp
ifconfig
netstat -i
netconfig
ipcalc
ping –cIR
traceroute

service arpwatch restart (Red Hat)
/etc/init.d/arpwatch start (Ubuntu)

wireshark

New Files and Directories:
/etc/resolv.conf
/var/arpwatch/arp.dat
/var/lib/arpwatch/arp.dat

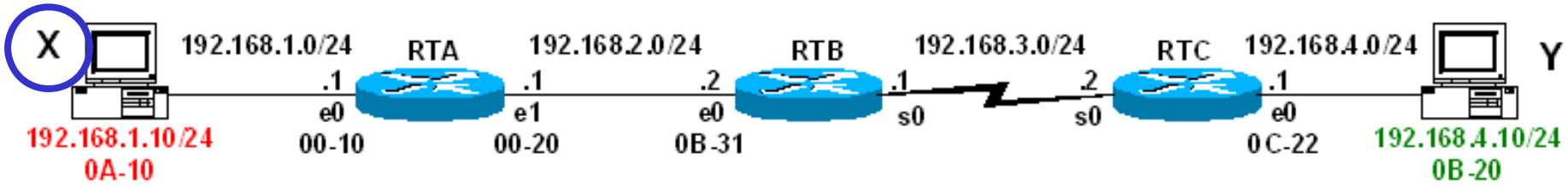
VMware:

IP addresses for VM's in the classroom

Station	IP	Static 1
Instructor	172.30.1.100	172.30.1.125
Station-01	172.30.1.101	172.30.1.126
Station-02	172.30.1.102	172.30.1.127
Station-03	172.30.1.103	172.30.1.128
Station-04	172.30.1.104	172.30.1.129
Station-05	172.30.1.105	172.30.1.130
Station-06	172.30.1.106	172.30.1.131
Station-07	172.30.1.107	172.30.1.132
Station-08	172.30.1.108	172.30.1.133
Station-09	172.30.1.109	172.30.1.134
Station-10	172.30.1.110	172.30.1.135
Station-11	172.30.1.111	172.30.1.136
Station-12	172.30.1.112	172.30.1.137

Station	IP	Static 1
Station-13	172.30.1.113	172.30.1.138
Station-14	172.30.1.114	172.30.1.139
Station-15	172.30.1.115	172.30.1.140
Station-16	172.30.1.116	172.30.1.141
Station-17	172.30.1.117	172.30.1.142
Station-18	172.30.1.118	172.30.1.143
Station-19	172.30.1.119	172.30.1.144
Station-20	172.30.1.120	172.30.1.145
Station-21	172.30.1.121	172.30.1.146
Station-22	172.30.1.122	172.30.1.147
Station-23	172.30.1.123	172.30.1.148
Station-24	172.30.1.124	172.30.1.149

Note the static IP address for your station to use in the next class exercise

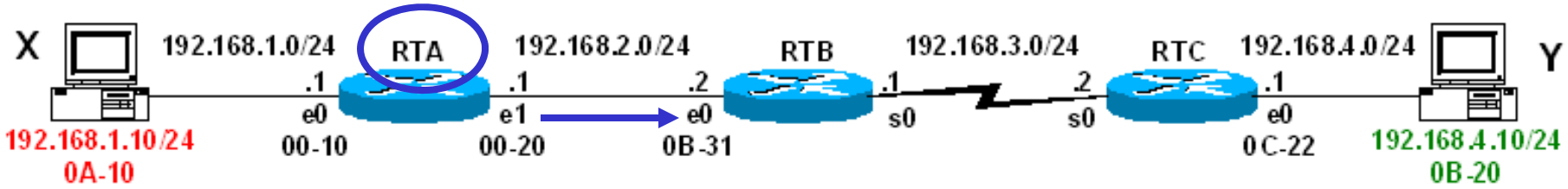


Layer 2 Data Link Frame **Layer 3 IP Packet**

Dest. MAC 00-10	Source MAC 0A-10	Type 800	Dest. IP 192.168.4.10	Source IP 192.168.1.10	IP fields	Data	Trailer

From Host X to Router RTA

- Host X begins by encapsulating the IP packet into a data link frame (in this case Ethernet) with RTA's Ethernet 0 interface's MAC address as the data link destination address.
- How does Host X know to forward to packet to RTA and not directly to Host Y?
  - IP Source and IP Destination Addresses are on different networks
- How does Host X know or get RTA's Ethernet address?
  - Checks ARP Table for Default Gateway IP Address and associated MAC Address.
- What if it there is not an entry in the ARP Table?
  - Host X sends an ARP Request and RTA sends an ARP Reply



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

Dest. MAC 0B-31	Source MAC 0A-10	Type 800	Dest. IP 192.168.4.10	Source IP 192.168.1.10	IP fields	Data	Trailer

RTA ARP Cache

IP Address	MAC Address
192.168.2.2	0B-31

RTA Routing Table

Network	Hops	Next-hop-ip	Exit-interface
192.168.1.0/24	0	Dir.Conn.	e0
192.168.2.0/24	0	Dir.Conn	e1
192.168.3.0/24	1	192.168.2.2	e1
192.168.4.0/24	2	192.168.2.2	e1

RTA

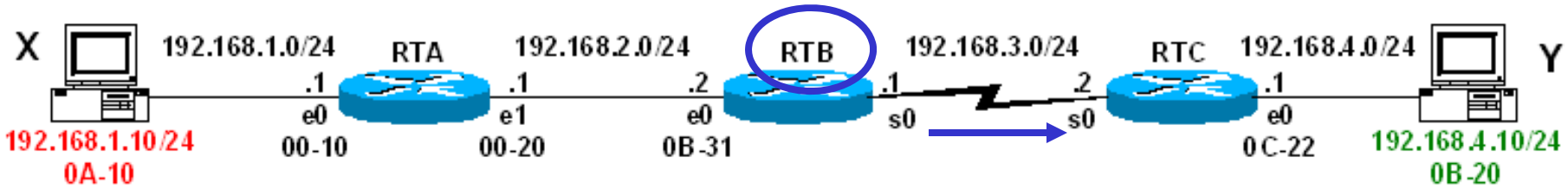
1. RTA examines Destination MAC address, which matches the E0 MAC address, so it copies in the frame.
2. RTA sees the Type field is 0x800, IP packet in the data field, a packet which needs to be routed.
3. RTA strips off the Ethernet frame.

RTA looks up the **Destination IP Address** in its routing table.

- 192.168.4.0/24 has next-hop-ip address of 192.168.2.2 and an exit-interface of e1.
- Since the exit interface is on an Ethernet network, RTA must resolve the next-hop-ip address with a destination MAC address.

4. RTA looks up the next-hop-ip address of 192.168.2.2 in its ARP cache.

- If the entry was not in the ARP cache, the RTA would need to send an ARP request out e1. RTB would send back an ARP reply, so RTA can update its ARP cache with an entry for 192.168.2.2. 5. Packet is encapsulated into a new data link (Ethernet) frame.



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

Dest. MAC 0B-31	Source MAC 00-20	Type 800	Dest. IP 192.168.4.10	Source IP 192.168.1.10	IP fields	Data	Trailer

RTB Routing Table

Network	Hops	Next-hop-ip	Exit-interface
192.168.1.0/24	1	192.168.2.1	e0
192.168.2.0/24	0	Dir.Conn	e0
192.168.3.0/24	0	Dir.Conn	s0
192.168.4.0/24	1	192.168.3.2	s0

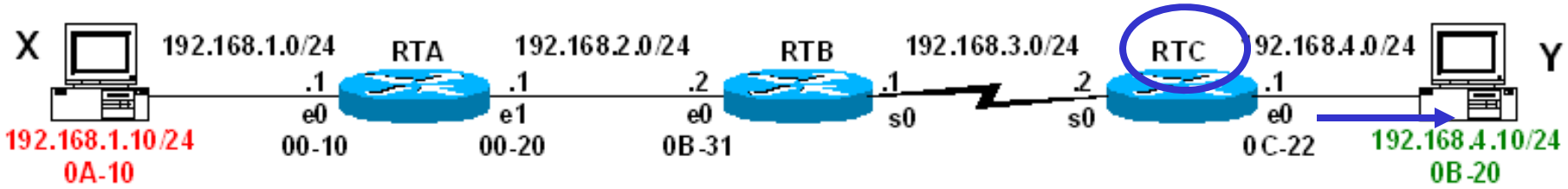
RTB

1. RTB examines Destination MAC address, which matches the E0 MAC address, and copies in the frame.
2. RTB sees Type field, 0x800, IP packet in the data field, a packet which needs to be routed.
3. RTB strips off the Ethernet frame.

RTB looks up the **Destination IP Address** in its routing table.

- 192.168.4.0/24 has next-hop-ip address of 192.168.3.2 and an exit-interface of Serial0.
- Since the exit interface is **not** an Ethernet network, RTB does **not** have to resolve the next-hop-ip address with a destination MAC address.
- When the interface is a **point-to-point serial connection**, (like a pipe), RTB encapsulates the IP packet into the proper data link frame, using the proper serial encapsulation (HDLC, PPP, etc.).
- The data link destination address is set to a **broadcast** (there's only one other end of the pipe).

5. Packet is encapsulated into a new data link (serial, PPP) frame and sent out the link.



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

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

RTC ARP Cache

IP Address	MAC Address
192.168.4.10	0B-20

RTC Routing Table

Network	Hops	Next-hop-ip	Exit-interface
192.168.1.0/24	2	192.168.3.1	s0
192.168.2.0/24	1	192.168.3.1	s0
192.168.3.0/24	0	Dir.Conn	s0
192.168.4.0/24	0	Dir.Conn	e0

RTC

1. RTC copies in the data link (serial, PPP) frame.
2. RTC sees the Type field is 0x800, IP packet in the data field, a packet which needs to be routed.
3. RTC strips off the data link, serial, frame.

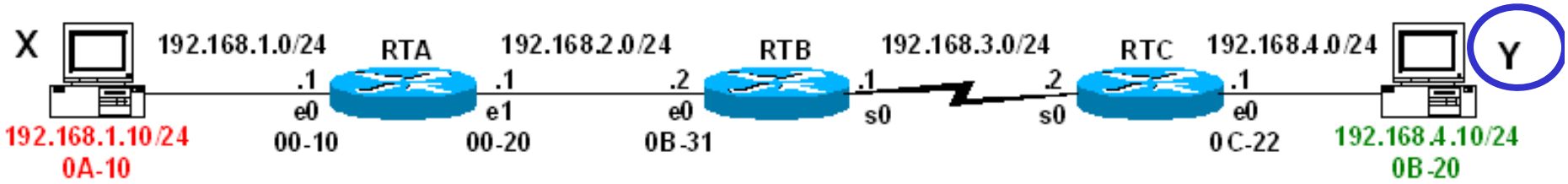
RTC looks up the **Destination IP Address** in its routing table.

- RTC realizes that this Destination IP Address is on the same network as one of its interfaces and it can sent the packet directly to the destination and not another router.
- Since the exit interface is on an directly connected Ethernet network, RTC must resolve the destination ip address with a destination MAC address.

2. RTC looks up the destination ip address of 192.168.4.10 in its ARP cache.

- If the entry was not in the ARP cache, the RTC would need to send an ARP request out e0. Host Y would send back an ARP reply, so RTC can update its ARP cache with an entry for 192.168.4.10.

5. Packet is encapsulated into a new data link (Ethernet) frame and sent out the interface.



Layer 2 Data Link Frame

Dest. MAC 0B-20	Source MAC 0C-22	Type 800	Data	Trailer

Host Y

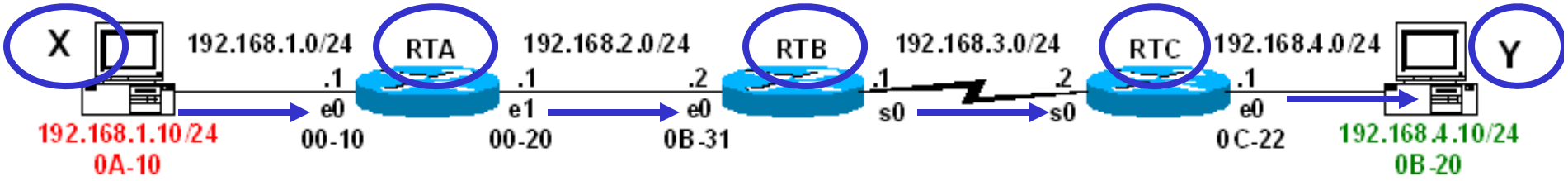
Layer 2: Data Link Frame

1. Host Y examines Destination MAC address, which matches its Ethernet interface MAC address, and copies in the frame.
2. Host Y sees the Type field is 0x800, IP packet in the data field, which needs to be sent to its IP process.
3. Host Y strips off the data link, Ethernet, frame and sends it to its IP process.

Layer 3: IP Packet

4. Host Y's IP process examines the **Destination IP Address** to make sure it matches its own IP Address..
   - If it does not, the packet will be dropped.
5. The packet's protocol field is examined to see where to send the data portion of this IP packet: TCP, UDP or other?

Layer 4: TCP, UDP or other?



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

- The summary once again!