Cabrillo College est. 1959

CIS 192 - Lesson 2

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

Solomon   Sean C.   Chris   Corey   Bryan   Sean F.

Tony   David   Donna   Stephanie

Dave   Evan   Gabriel   Elia   Tajvia   Carlos   Adam

Ben   Laura

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

Lesson Module Checklist

- Slides
- Flashcards
- 1st minute quiz
- Web Calendar summary
- Web book pages
- Commands
- Howtos

- Lab tested
- Opus – lab template in depot
- Youtube Videos, if any, uploaded

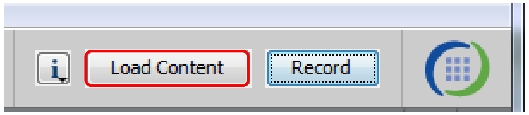
- Whiteboard updated with random order quiz questions

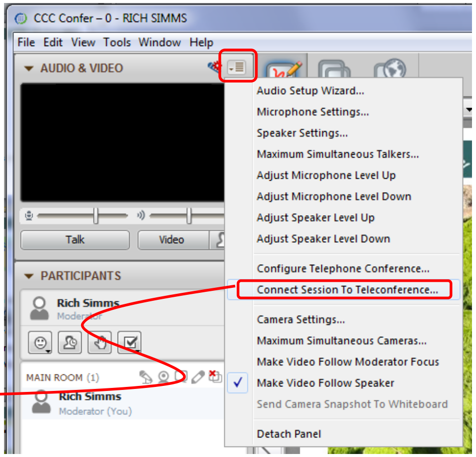
- Bring Add Codes
- Bring printed roster

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

CCC Confer

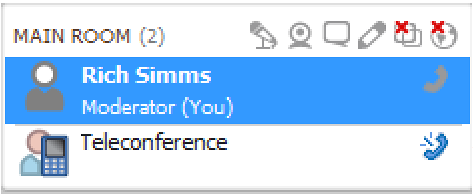
[ ] **Preload White Board with *cis*lesson??*-WB***



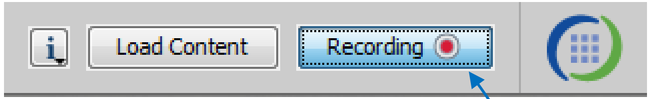

CCC Confer – 0 - RICH SIMMS
File Edit View Tools Window Help
AUDIO & VIDEO
Audio Setup Wizard...
Microphone Settings...
Speaker Settings...
Maximum Simultaneous Talkers...
Adjust Microphone Level Up
Adjust Microphone Level Down
Adjust Speaker Level Up
Adjust Speaker Level Down
Configure Telephone Conference...
Connect Session To Teleconference...
Camera Settings...
Maximum Simultaneous Cameras...
Make Video Follow Moderator Focus
Make Video Follow Speaker
Send Camera Snapshot To Whiteboard
Detach Panel
Talk
Video
PARTICIPANTS
Rich Simms
Moderator
MAIN ROOM (1)
Rich Simms
Moderator (You)

[ ] **Connect session to Teleconference**

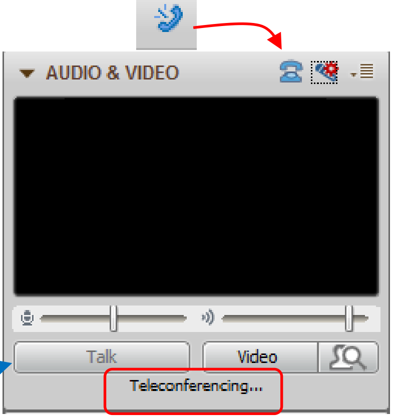
Session now connected to teleconference



[ ] **Is recording on?**



Red dot means recording

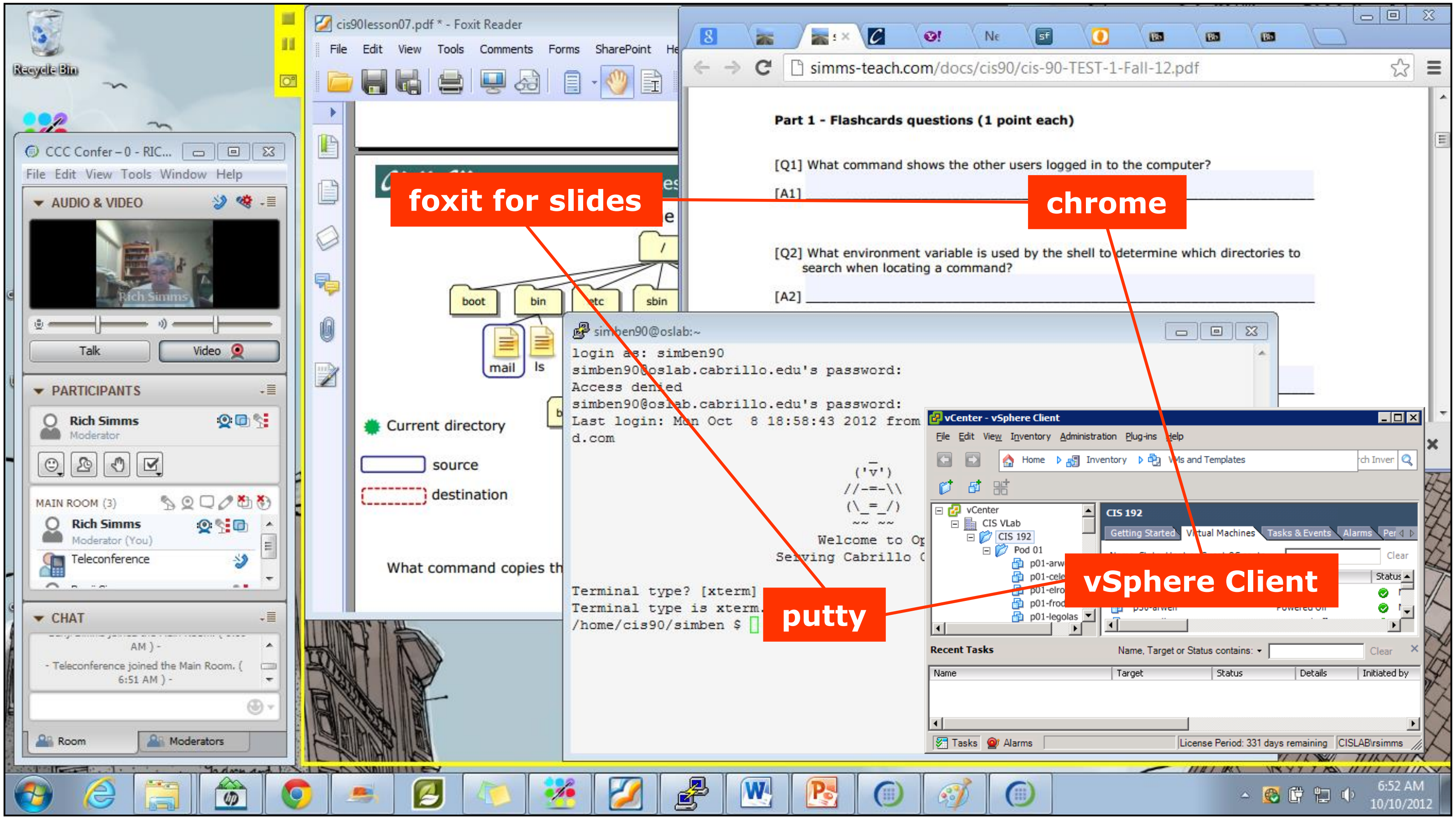


[ ] **Use teleconferencing, not mic**

Should be greyed out

[ ] **Video (webcam) optional**

[ ] **layout and share apps**



What command copies th

simms-teach.com/docs/cis90/cis-90-TEST-1-Fall-12.pdf

Part 1 - Flashcards questions (1 point each)

[Q1] What command shows the other users logged in to the computer?

[A1] ___

[Q2] What environment variable is used by the shell to determine which directories to search when locating a command?

[A2] ___

```
simben90@oslab:~
login as: simben90
simben90@oslab.cabrillo.edu's password:
Access denied
simben90@oslab.cabrillo.edu's password:
Last login: Mon Oct  8 18:58:43 2012 from
d.com

                                  ('v')
                                 //-=-\\
                                 (\_=_/)
                                  ~~ ~~
                              Welcome to Op
                           Serving Cabrillo C

Terminal type? [xterm]
Terminal type is xterm.
/home/cis90/simben $
```

foxit for slides

chrome

putty

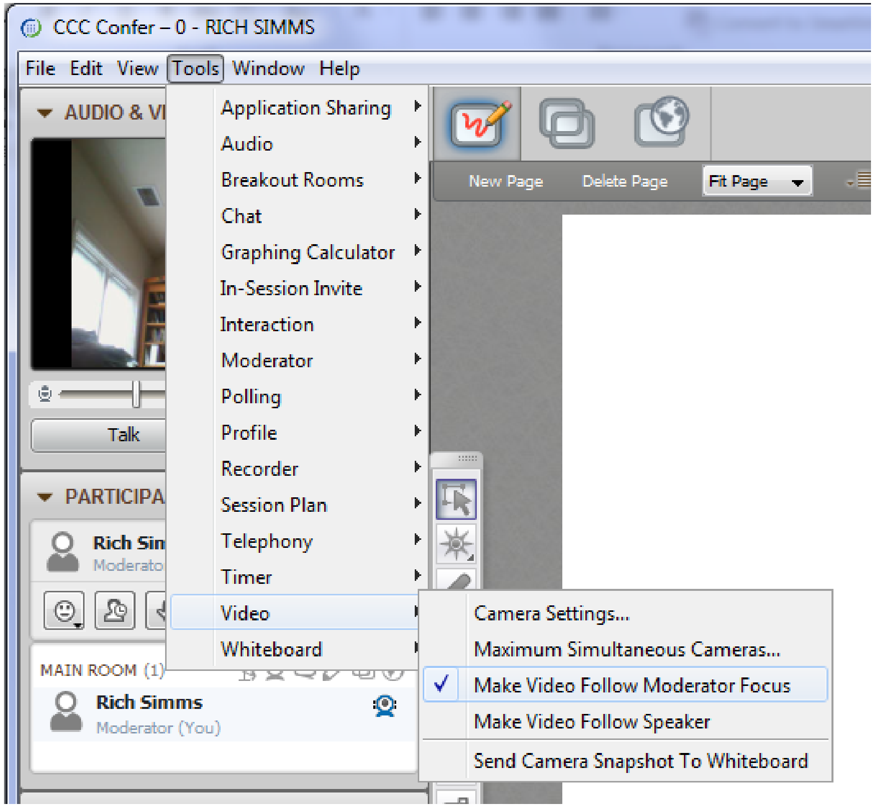
vSphere Client

CCC Confer

[ ] Video (webcam) optional

[ ] Follow moderator

[ ] Double-click on postages stamps

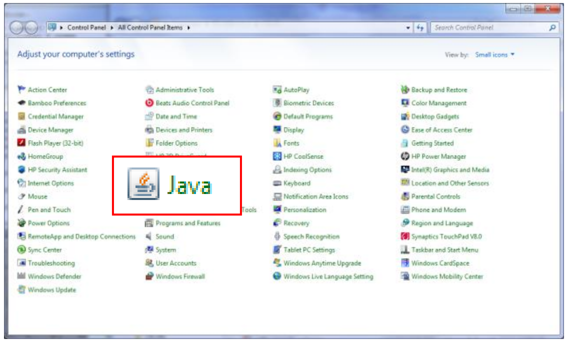


✓ Make Video Follow Moderator Focus

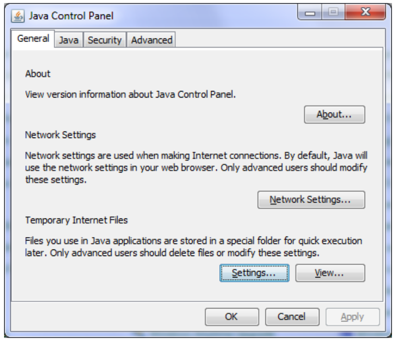
Universal Fix for CCC Confer:

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

Control Panel (small icons)



General Tab > Settings...



About

View version information about Java Control Panel.

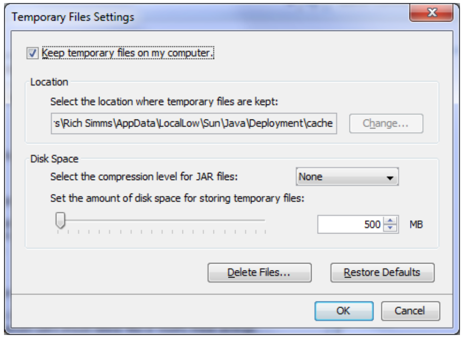
Network Settings

Network settings are used when making Internet connections. By default, Java will use the network settings in your web browser. Only advanced users should modify these settings.

Temporary Internet Files

Files you use in Java applications are stored in a special folder for quick execution later. Only advanced users should delete files or modify these settings.

500MB cache size



☑ Keep temporary files on my computer.

Location

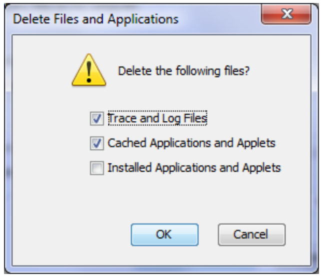
Select the location where temporary files are kept: s\Rich Simms\AppData\LocalLow\Sun\Java\Deployment\cache

Disk Space

Select the compression level for JAR files: None

Set the amount of disk space for storing temporary files: 500 MB

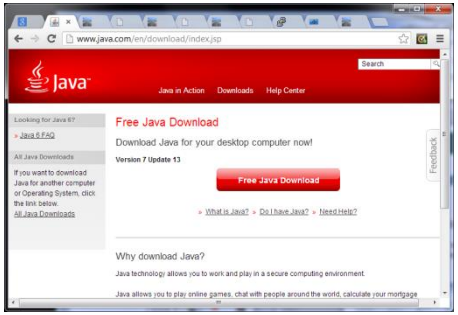
Delete these



Delete the following files?

☑ Trace and Log Files
☑ Cached Applications and Applets
☐ Installed Applications and Applets

Google Java download



Looking for Java 6?

Java 6 FAQ

All Java Downloads

If you want to download Java for another computer or Operating System, click the link below.

All Java Downloads

Free Java Download

Download Java for your desktop computer now!

Version 7 Update 13

Why download Java?

Java technology allows you to work and play in a secure computing environment.

Java allows you to play online games, chat with people around the world, calculate your mortgage

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/

Q1

First Minute Quiz

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

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

ARP and the Internet Layer

Related Course Objectives

- Use basic network terminology to describe the five layers of the TCP/IP Reference Model, and describe at least one major function of each layer.
- Use the arpwatch daemon to collect IP/hardware addresses, and manually add an address to the ARP table.
- Install the device drivers and configure the network interface card (NIC) of a Linux system so that it may join a network.
- Configure appropriate IP addresses, network and subnet masks, and broadcast addresses based on the size and number of network segments required.
- Use a network sniffer to analyze network traffic between two hosts.
- Identify, isolate, and correct malfunctions in a computer network.

Agenda

- Quiz
- Questions on previous material
- Housekeeping
- Cabling VMs
- Joining a network (temp)
- Joining a network (perm)
- Aliases
- ARP
- arpwatch
- Viewing packets
- Internet Layer
- IPv4 Addressing
- NAT/PAT and IPv6
- Traversing VMs using SSH
- Troubleshooting
- Lab
- Wrap

Questions

Questions

How this course works?

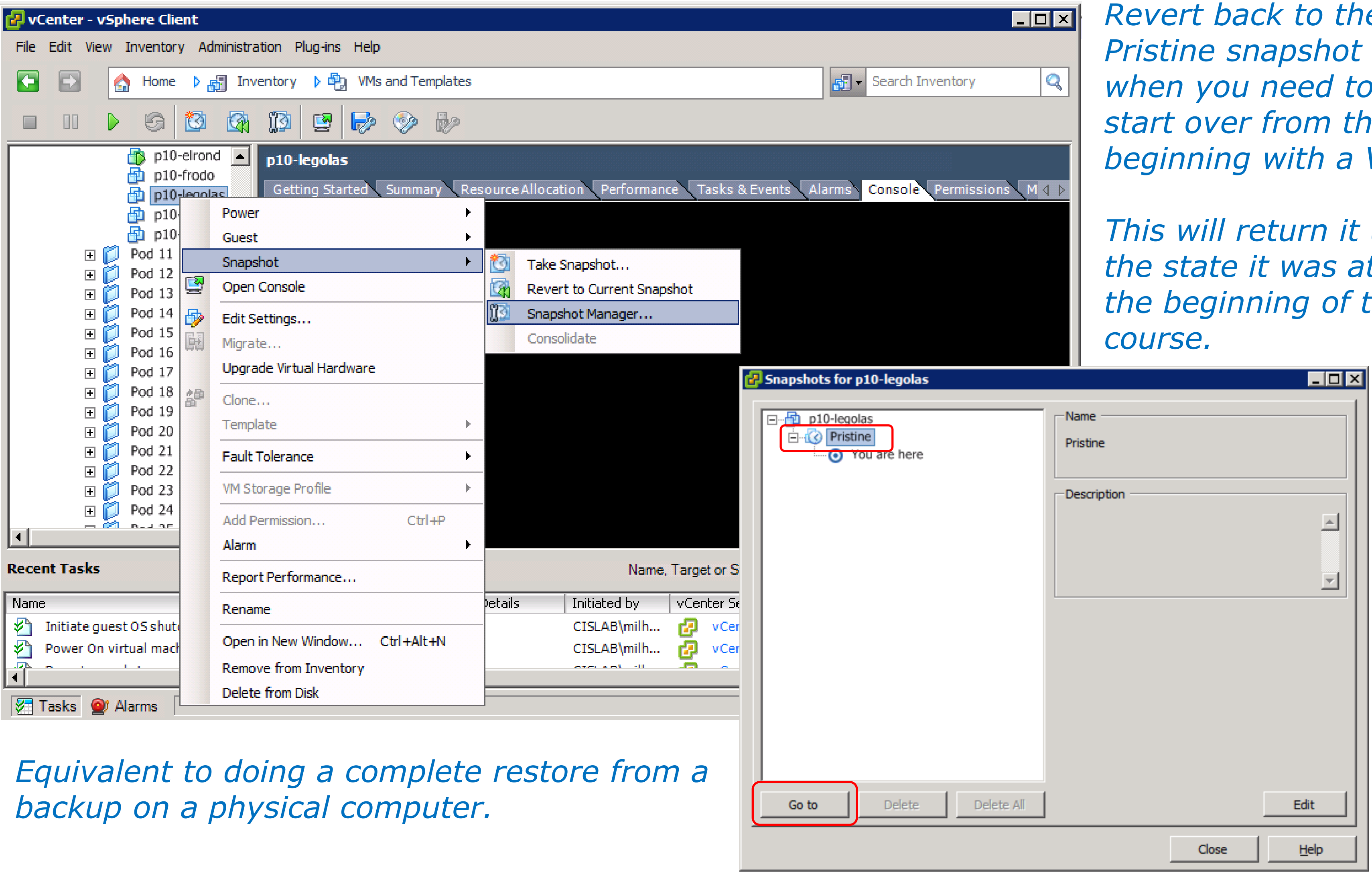
Lesson 1?

Lab 1?

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

VMware Tips

Revert to a "Pristine" snapshot

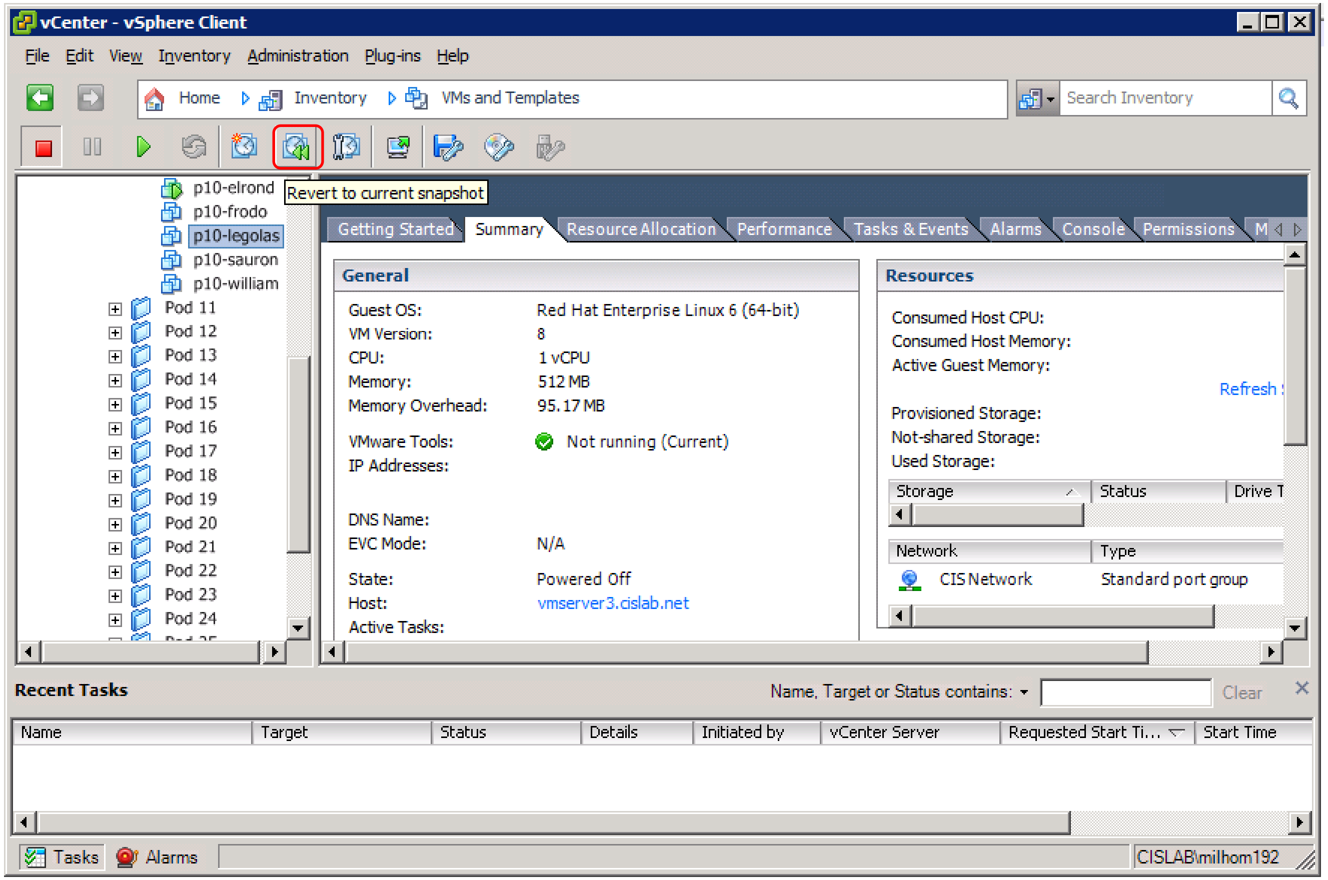


Revert back to the Pristine snapshot when you need to start over from the beginning with a VM.

This will return it to the state it was at the beginning of the course.

Equivalent to doing a complete restore from a backup on a physical computer.

Revert to a "Pristine" snapshot



General

Guest OS: Red Hat Enterprise Linux 6 (64-bit)
VM Version: 8
CPU: 1 vCPU
Memory: 512 MB
Memory Overhead: 95.17 MB

VMware Tools: ✓ Not running (Current)
IP Addresses:

DNS Name:
EVC Mode: N/A

State: Powered Off
Host: vmserver3.cislab.net
Active Tasks:

Resources

Consumed Host CPU:
Consumed Host Memory:
Active Guest Memory:

Provisioned Storage:
Not-shared Storage:
Used Storage:

Storage	Status	Drive T

Network	Type
CIS Network	Standard port group

Name	Target	Status	Details	Initiated by	vCenter Server	Requested Start Ti...	Start Time

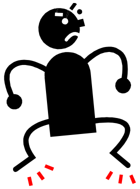
There is also a shortcut on the toolbar to revert to the current snapshot

Never delete a snapshot on one of the CIS 192 VMs

It will wipe out all other VMs in all pods using the same distro!

Really!

An auxiliary set of pods is available in case this ever happens. All configurations made to VMs in primary pods will be lost! The student security roles do not allow removing a snapshot so hopefully this will never happen!



Repeating Keystrokes in VLab

Details
When typing into a remote console, you see unintended repeated keystrokes.

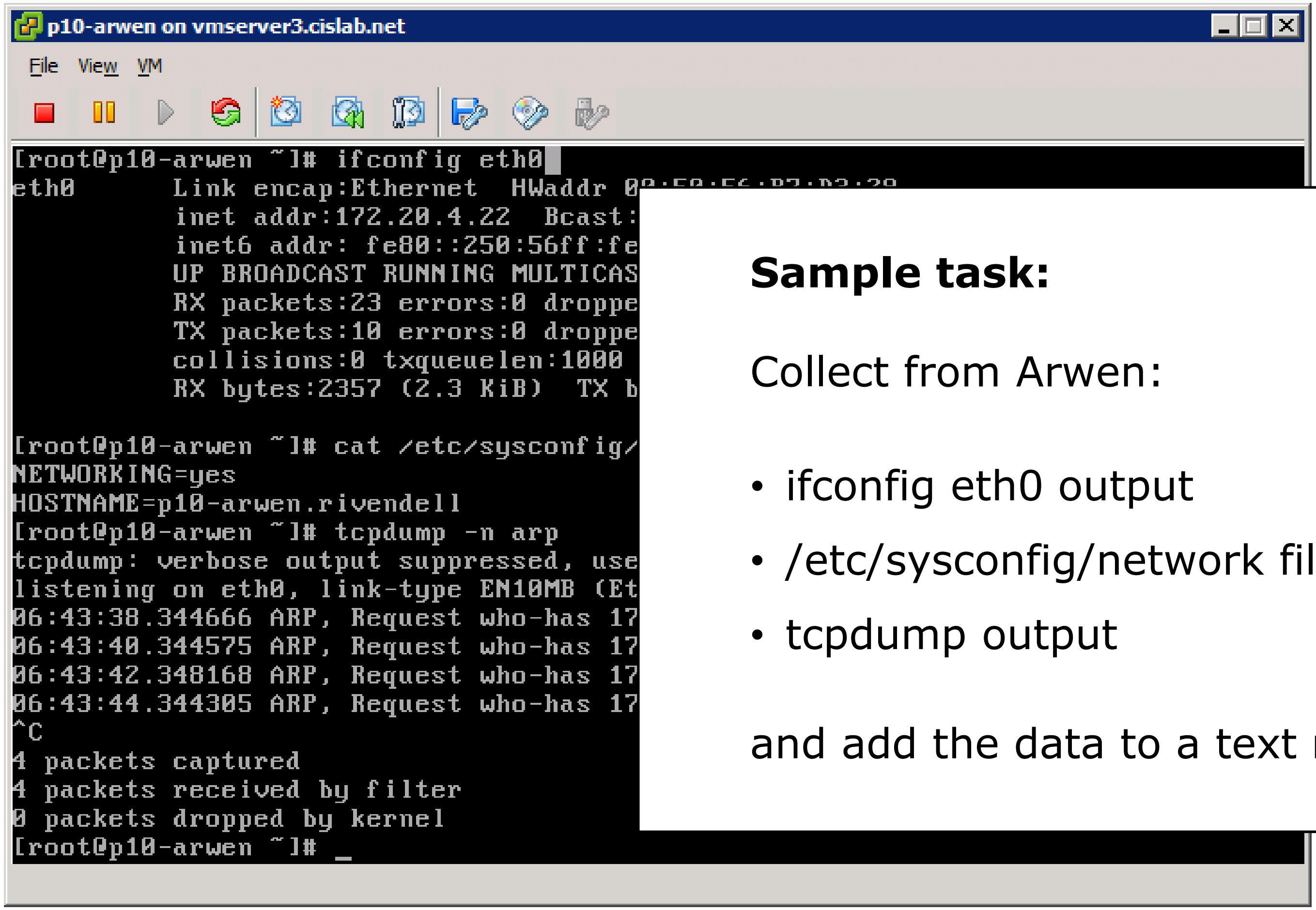
Solution
To make the changes using the vSphere Client:

1. Power off the virtual machine
2. Right click virtual machine select Edit Settings
3. Click Options > General > Configuration Parameters
4. Click Add Row
    - Under Name enter: **keyboard.typematicMinDelay**
    - Under Value enter: **2000000**
5. Click OK
6. Power on the virtual machine

http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=196

VLAB Tips

Collecting data for lab reports



```
[root@p10-arwen ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:B7:D3:28
          inet addr:172.20.4.22  Bcast:
          inet6 addr: fe80::250:56ff:fe
          UP BROADCAST RUNNING MULTICAS
          RX packets:23 errors:0 droppe
          TX packets:10 errors:0 droppe
          collisions:0 txqueuelen:1000
          RX bytes:2357 (2.3 KiB)  TX b

[root@p10-arwen ~]# cat /etc/sysconfig/
NETWORKING=yes
HOSTNAME=p10-arwen.rivendell
[root@p10-arwen ~]# tcpdump -n arp
tcpdump: verbose output suppressed, use
listening on eth0, link-type EN10MB (Et
06:43:38.344666 ARP, Request who-has 17
06:43:40.344575 ARP, Request who-has 17
06:43:42.348168 ARP, Request who-has 17
06:43:44.344305 ARP, Request who-has 17
^C
4 packets captured
4 packets received by filter
0 packets dropped by kernel
[root@p10-arwen ~]# _
```

Sample task:

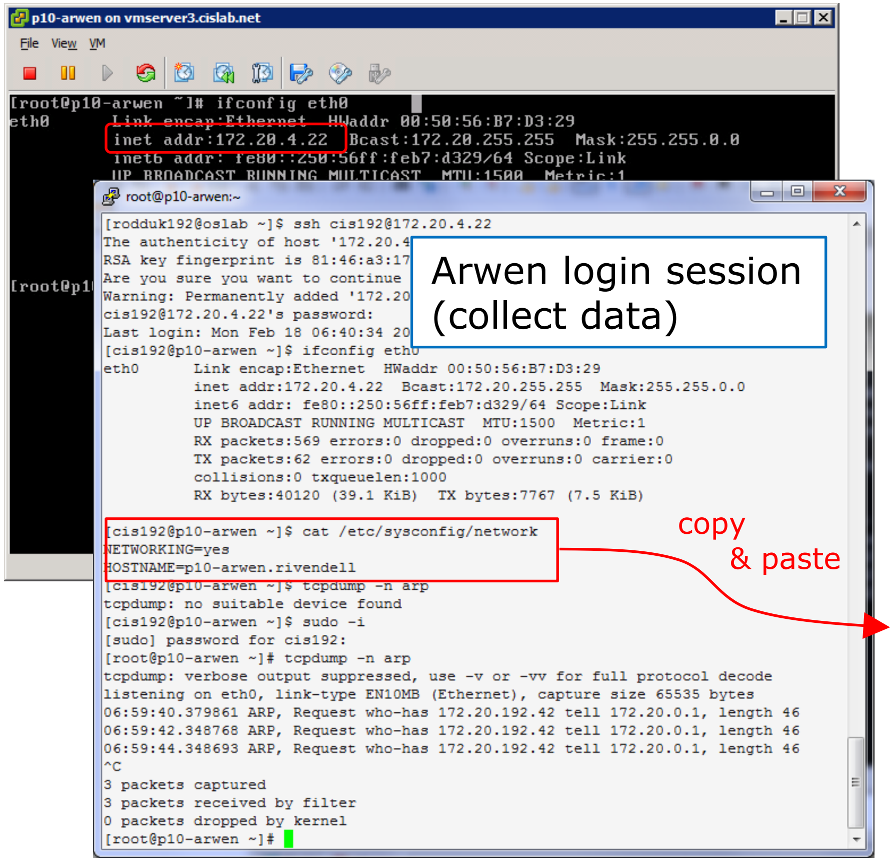
Collect from Arwen:

- ifconfig eth0 output
- /etc/sysconfig/network file contents
- tcpdump output

and add the data to a text report on Opus

Virtual terminals in VLab are copy/paste challenged

Method 1 - Two terminals, copy & paste



p10-arwen on vmserver3.cislab.net

```
[root@p10-arwen ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:B7:D3:29
          inet addr:172.20.4.22  Bcast:172.20.255.255  Mask:255.255.0.0
          inet6 addr: fe80::250:56ff:feb7:d329/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

[root@p1
```

root@p10-arwen:~

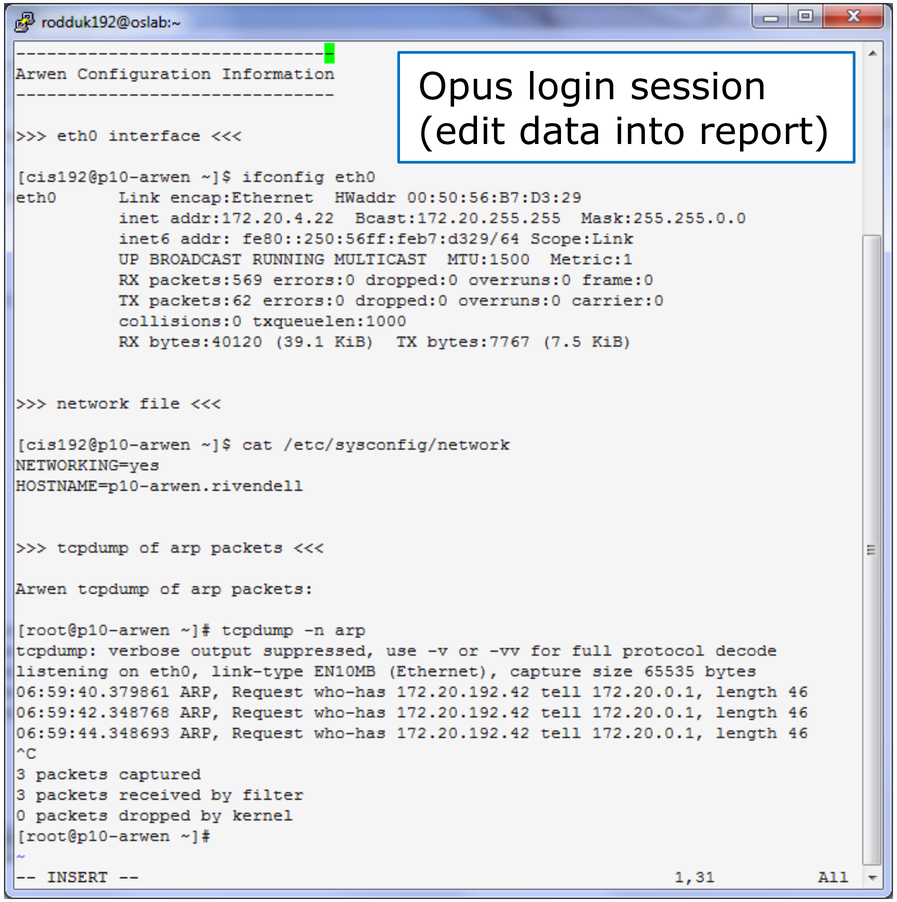
```
[rodduk192@oslab ~]$ ssh cis192@172.20.4.22
The authenticity of host '172.20.4
RSA key fingerprint is 81:46:a3:17
Are you sure you want to continue
Warning: Permanently added '172.20
cis192@172.20.4.22's password:
Last login: Mon Feb 18 06:40:34 20
[cis192@p10-arwen ~]$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:B7:D3:29
          inet addr:172.20.4.22  Bcast:172.20.255.255  Mask:255.255.0.0
          inet6 addr: fe80::250:56ff:feb7:d329/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:569 errors:0 dropped:0 overruns:0 frame:0
          TX packets:62 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:40120 (39.1 KiB)  TX bytes:7767 (7.5 KiB)

[cis192@p10-arwen ~]$ cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=p10-arwen.rivendell
[cis192@p10-arwen ~]$ tcpdump -n arp
tcpdump: no suitable device found
[cis192@p10-arwen ~]$ sudo -i
[sudo] password for cis192:
[root@p10-arwen ~]# tcpdump -n arp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
06:59:40.379861 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
06:59:42.348768 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
06:59:44.348693 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
^C
3 packets captured
3 packets received by filter
0 packets dropped by kernel
[root@p10-arwen ~]#
```

Arwen login session (collect data)

copy & paste

1) ssh into the VM, using its IP address, with a smarter terminal like Putty or the Mac terminal. Copy appropriate data to the clipboard.

2) Paste the data into the report you are editing in the Opus login session

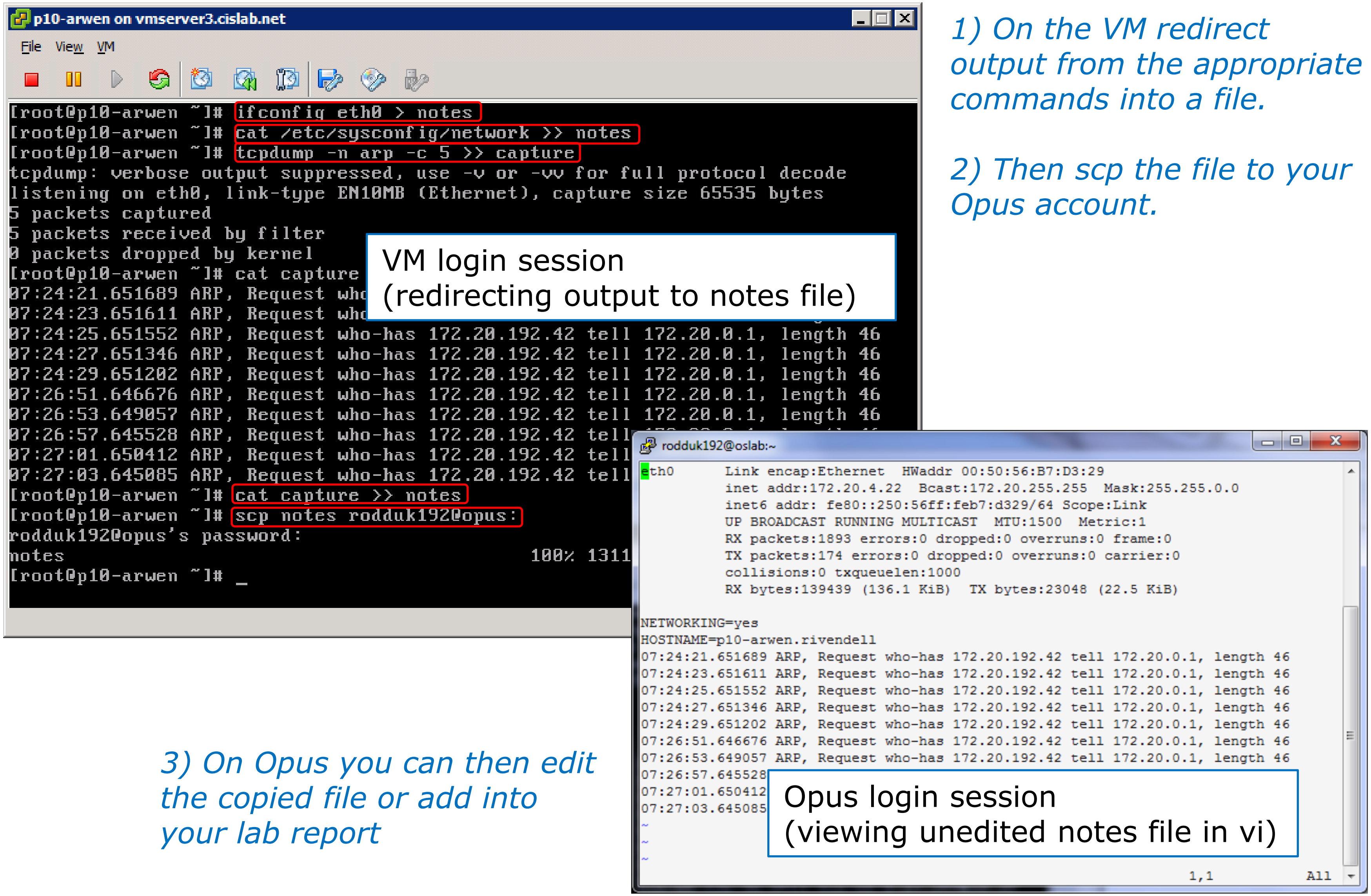


rodduk192@oslab:~

```
--------------------------------
Arwen Configuration Information
--------------------------------

>>> eth0 interface <<<

[cis192@p10-arwen ~]$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:B7:D3:29
          inet addr:172.20.4.22  Bcast:172.20.255.255  Mask:255.255.0.0
          inet6 addr: fe80::250:56ff:feb7:d329/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:569 errors:0 dropped:0 overruns:0 frame:0
          TX packets:62 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:40120 (39.1 KiB)  TX bytes:7767 (7.5 KiB)


>>> network file <<<

[cis192@p10-arwen ~]$ cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=p10-arwen.rivendell


>>> tcpdump of arp packets <<<

Arwen tcpdump of arp packets:

[root@p10-arwen ~]# tcpdump -n arp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
06:59:40.379861 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
06:59:42.348768 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
06:59:44.348693 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
^C
3 packets captured
3 packets received by filter
0 packets dropped by kernel
[root@p10-arwen ~]#
~
-- INSERT --                                                  1,31          All
```

Opus login session (edit data into report)

Method 2 - redirection and scp



```
[root@p10-arwen ~]# ifconfig eth0 > notes
[root@p10-arwen ~]# cat /etc/sysconfig/network >> notes
[root@p10-arwen ~]# tcpdump -n arp -c 5 >> capture
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
5 packets captured
5 packets received by filter
0 packets dropped by kernel
[root@p10-arwen ~]# cat capture
07:24:21.651689 ARP, Request who
07:24:23.651611 ARP, Request who
07:24:25.651552 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:24:27.651346 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:24:29.651202 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:26:51.646676 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:26:53.649057 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:26:57.645528 ARP, Request who-has 172.20.192.42 tell
07:27:01.650412 ARP, Request who-has 172.20.192.42 tell
07:27:03.645085 ARP, Request who-has 172.20.192.42 tell
[root@p10-arwen ~]# cat capture >> notes
[root@p10-arwen ~]# scp notes rodduk192@opus:
rodduk192@opus's password:
notes                                        100% 1311
[root@p10-arwen ~]# _
```

VM login session
(redirecting output to notes file)

```
eth0      Link encap:Ethernet  HWaddr 00:50:56:B7:D3:29
          inet addr:172.20.4.22  Bcast:172.20.255.255  Mask:255.255.0.0
          inet6 addr: fe80::250:56ff:feb7:d329/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1893 errors:0 dropped:0 overruns:0 frame:0
          TX packets:174 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:139439 (136.1 KiB)  TX bytes:23048 (22.5 KiB)

NETWORKING=yes
HOSTNAME=p10-arwen.rivendell
07:24:21.651689 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:24:23.651611 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:24:25.651552 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:24:27.651346 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:24:29.651202 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:26:51.646676 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:26:53.649057 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:26:57.645528
07:27:01.650412
07:27:03.645085
~
~
~
                                                             1,1           All
```

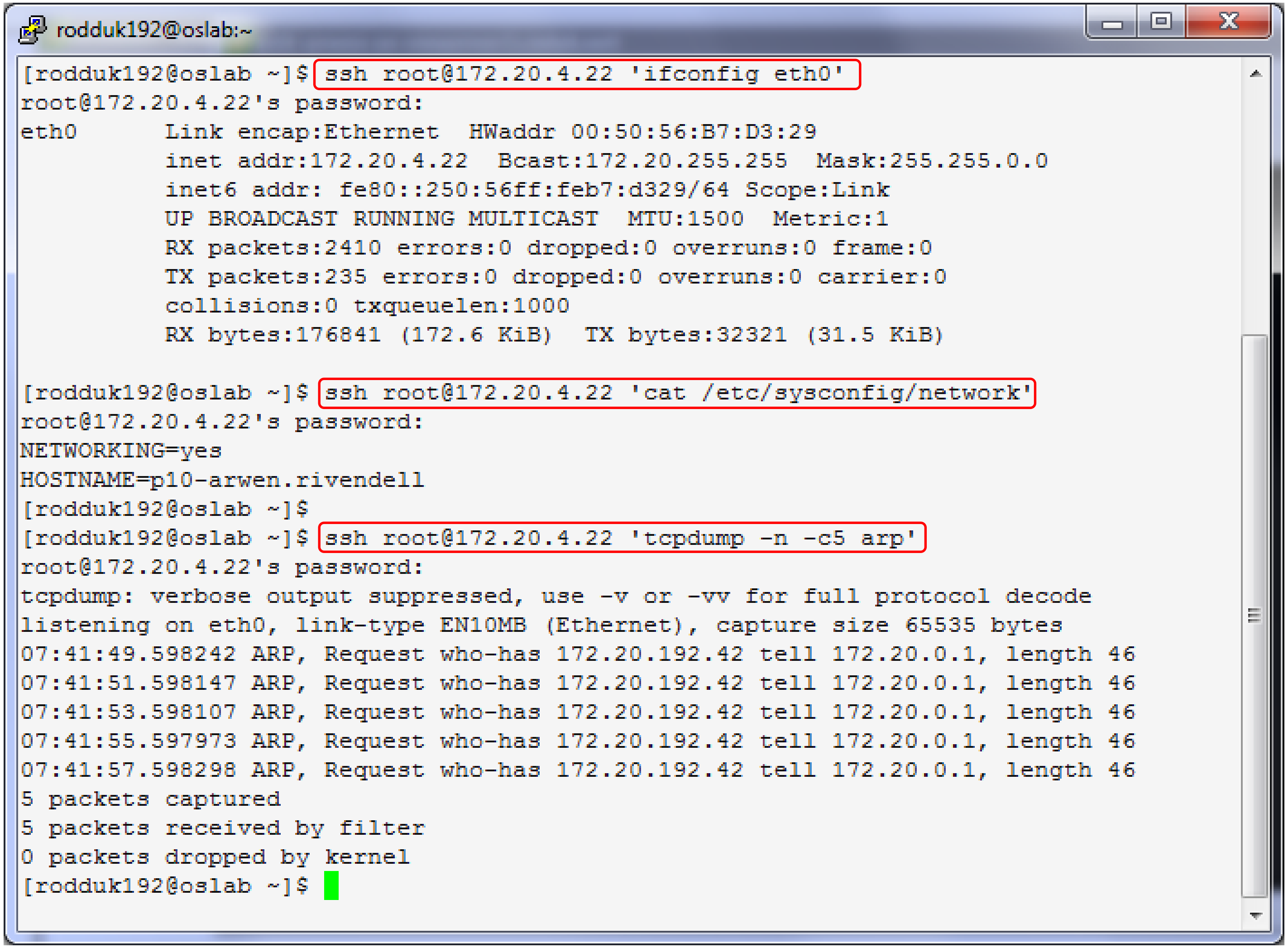
Opus login session
(viewing unedited notes file in vi)

1) On the VM redirect output from the appropriate commands into a file.

2) Then scp the file to your Opus account.

3) On Opus you can then edit the copied file or add into your lab report

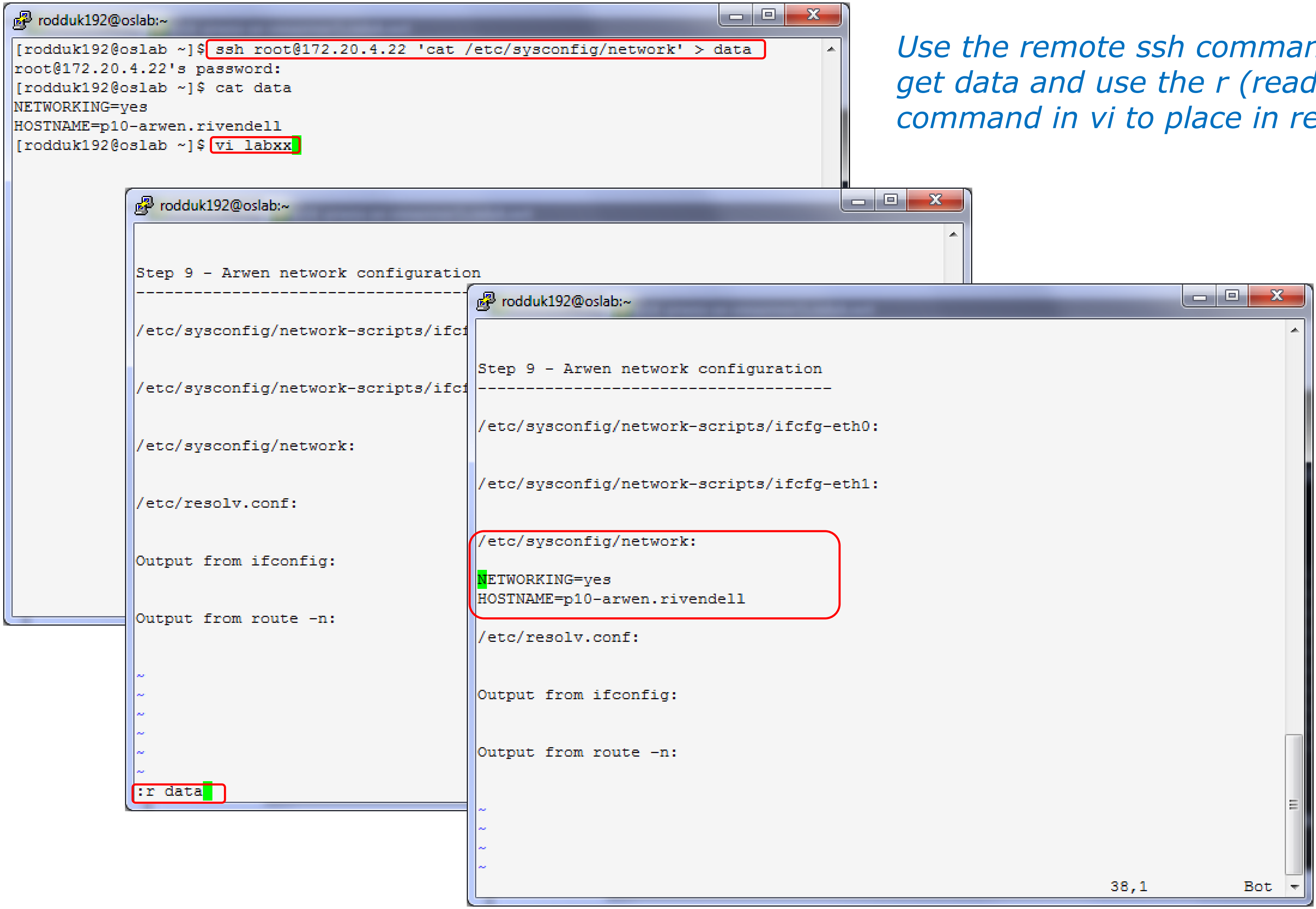
Method 3 - run remote ssh commands from Opus



```
[rodduk192@oslab ~]$ ssh root@172.20.4.22 'ifconfig eth0'
root@172.20.4.22's password:
eth0      Link encap:Ethernet  HWaddr 00:50:56:B7:D3:29
          inet addr:172.20.4.22  Bcast:172.20.255.255  Mask:255.255.0.0
          inet6 addr: fe80::250:56ff:feb7:d329/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2410 errors:0 dropped:0 overruns:0 frame:0
          TX packets:235 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:176841 (172.6 KiB)  TX bytes:32321 (31.5 KiB)

[rodduk192@oslab ~]$ ssh root@172.20.4.22 'cat /etc/sysconfig/network'
root@172.20.4.22's password:
NETWORKING=yes
HOSTNAME=p10-arwen.rivendell
[rodduk192@oslab ~]$
[rodduk192@oslab ~]$ ssh root@172.20.4.22 'tcpdump -n -c5 arp'
root@172.20.4.22's password:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
07:41:49.598242 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:41:51.598147 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:41:53.598107 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:41:55.597973 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
07:41:57.598298 ARP, Request who-has 172.20.192.42 tell 172.20.0.1, length 46
5 packets captured
5 packets received by filter
0 packets dropped by kernel
[rodduk192@oslab ~]$
```

You could now use copy & paste to paste data into your report

Method 4 - No copy & paste whatsoever



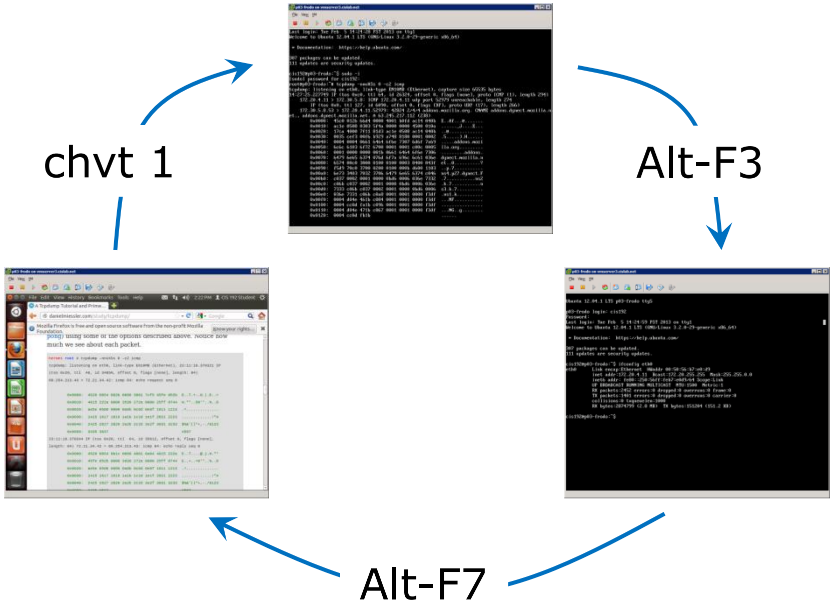
```
[rodduk192@oslab ~]$ ssh root@172.20.4.22 'cat /etc/sysconfig/network' > data
root@172.20.4.22's password:
[rodduk192@oslab ~]$ cat data
NETWORKING=yes
HOSTNAME=p10-arwen.rivendell
[rodduk192@oslab ~]$ vi labxx
```

```
Step 9 - Arwen network configuration
-----------------------------------

/etc/sysconfig/network-scripts/ifcf

/etc/sysconfig/network-scripts/ifcf

/etc/sysconfig/network:

/etc/resolv.conf:

Output from ifconfig:

Output from route -n:

~
~
~
~
~
~
:r data
```

```
Step 9 - Arwen network configuration
-------------------------------------

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

/etc/sysconfig/network-scripts/ifcfg-eth1:

/etc/sysconfig/network:

NETWORKING=yes
HOSTNAME=p10-arwen.rivendell

/etc/resolv.conf:

Output from ifconfig:

Output from route -n:

~
~
~
~
                                                                38,1          Bot
```

Use the remote ssh command to get data and use the r (read) command in vi to place in report

Changing Virtual Terminals Part II



```
Last login: Tue Feb  5 14:24:28 PST 2013 on tty1
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

307 packages can be updated.
111 updates are security updates.

cis192@p03-frodo:~$ sudo -i
[sudo] password for cis192:
root@p03-frodo:~# tcpdump -vvvXs 0 -c2 icmp
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
14:27:25.227749 IP (tos 0xc0, ttl 64, id 26324, offset 0, flags [none], proto ICMP (1), length 294)
    172.20.4.11 > 172.30.5.8: ICMP 172.20.4.11 udp port 52979 unreachable, length 274
        IP (tos 0x0, ttl 127, id 6090, offset 0, flags [DF], proto UDP (17), length 266)
    172.30.5.8.53 > 172.20.4.11.52979: 42024 2/4/4 addons.mozilla.org. CNAME addons.dynect.mozilla.n
et., addons.dynect.mozilla.net. A 63.245.217.112 (238)
        0x0000:  45c0 0126 66d4 0000 4001 b0fd ac14 040b  E..&f...@.......
        0x0010:  ac1e 0508 0303 5f4a 0000 0000 4500 010a  ......_J....E...
        0x0020:  17ca 4000 7f11 0543 ac1e 0508 ac14 040b  ..@....C........
        0x0030:  0035 cef3 00f6 b929 a748 8180 0001 0002  .5.....).H......
        0x0040:  0004 0004 0661 6464 6f6e 7307 6d6f 7a69  .....addons.mozi
        0x0050:  6c6c 6103 6f72 6700 0001 0001 c00c 0005  lla.org.........
        0x0060:  0001 0000 0000 001b 0661 6464 6f6e 7306  .........addons.
        0x0070:  6479 6e65 6374 076d 6f7a 696c 6c61 036e  dynect.mozilla.n
        0x0080:  6574 00c0 3000 0100 0100 0003 0400 043f  et..0..........?
        0x0090:  f5d9 70c0 3700 0200 0100 000b d600 1503  ..p.7...........
        0x00a0:  6e73 3403 7032 3706 6479 6e65 6374 c046  ns4.p27.dynect.F
        0x00b0:  c037 0002 0001 0000 0bd6 0006 036e 7332  .7...........ns2
        0x00c0:  c06b c037 0002 0001 0000 0bd6 0006 036e  .k.7...........n
        0x00d0:  7333 c06b c037 0002 0001 0000 0bd6 0006  s3.k.7..........
        0x00e0:  036e 7331 c06b c0a0 0001 0001 0000 f3df  .ns1.k..........
        0x00f0:  0004 d04e 461b c0b4 0001 0001 0000 f3df  ...NF...........
        0x0100:  0004 cc0d fa1b c096 0001 0001 0000 f3df  ................
        0x0110:  0004 d04e 471b c067 0001 0001 0000 f3df  ...NG..g........
        0x0120:  0004 cc0d fb1b                           ......
```

```
Ubuntu 12.04.1 LTS p03-frodo tty5

p03-frodo login: cis192
Password:
Last login: Tue Feb  5 14:24:59 PST 2013 on tty1
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

307 packages can be updated.
111 updates are security updates.

cis192@p03-frodo:~$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:b7:e0:d9
          inet addr:172.20.4.11  Bcast:172.20.255.255  Mask:255.255.0.0
          inet6 addr: fe80::250:56ff:feb7:e0d9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2452 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1401 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2874799 (2.8 MB)  TX bytes:151204 (151.2 KB)

cis192@p03-frodo:~$
```

Some cool new ways to change virtual terminals contributed by forum posters:

- Use the **chvt** command. For example **chvt 3** changes to tty3.
- If you are in one of the tty virtual terminals like tty1 you only need to type **Alt-F*n*** to change. For example, if you are in tty1, **Alt-F3** changes to tty3.

Network Configuration

(Joining a network)

Joining a network

1) With only a loopback interface active we can only communicate with ourselves.

2) Adding an **IP address** and **subnet mask** enables us to communicate with other hosts on the same LAN segment.

3) Adding a **default gateway** enables us to communicate with hosts anywhere on the Internet.

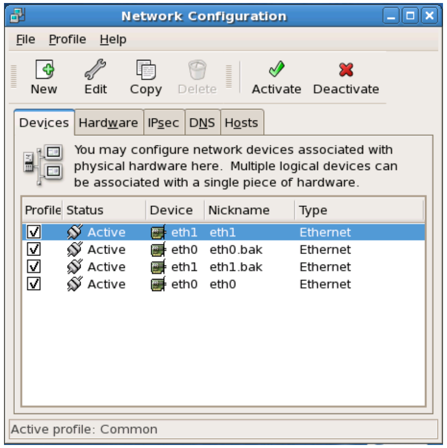
4) Adding one or more **DNS name servers** enables us to specify hosts on the Internet by name.

5) Setting a hostname gives our local system a name to go by.

GUI method

*The **GUI** (Graphical User Interface) tools are easy to use but they are different with each distribution.*

CentOS 5.4



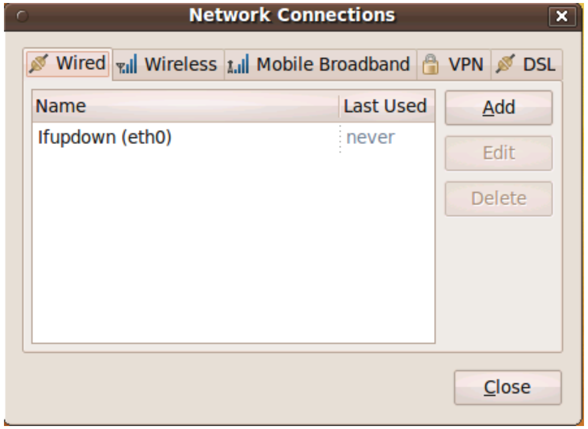
You may configure network devices associated with physical hardware here. Multiple logical devices can be associated with a single piece of hardware.

Profile	Status	Device	Nickname	Type
☑	Active	eth1	eth1	Ethernet
☑	Active	eth0	eth0.bak	Ethernet
☑	Active	eth1	eth1.bak	Ethernet
☑	Active	eth0	eth0	Ethernet

Active profile: Common

System
> Administration
> Network

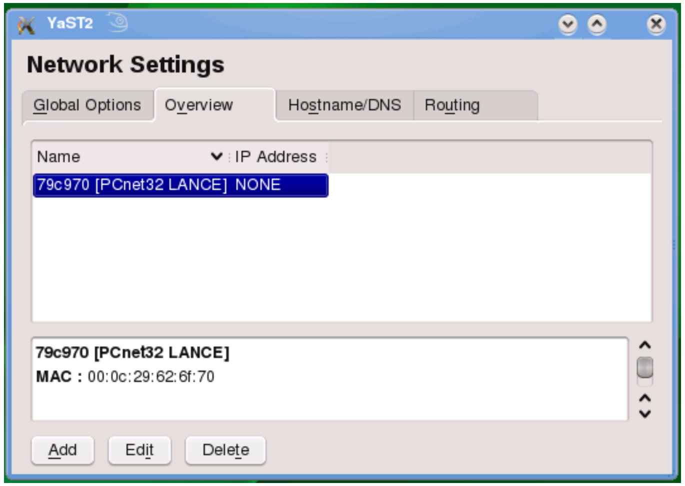
Ubuntu 9.10



Name	Last Used
Ifupdown (eth0)	never

System
> Preferences
> Network Connections

OpenSUSE 11.2



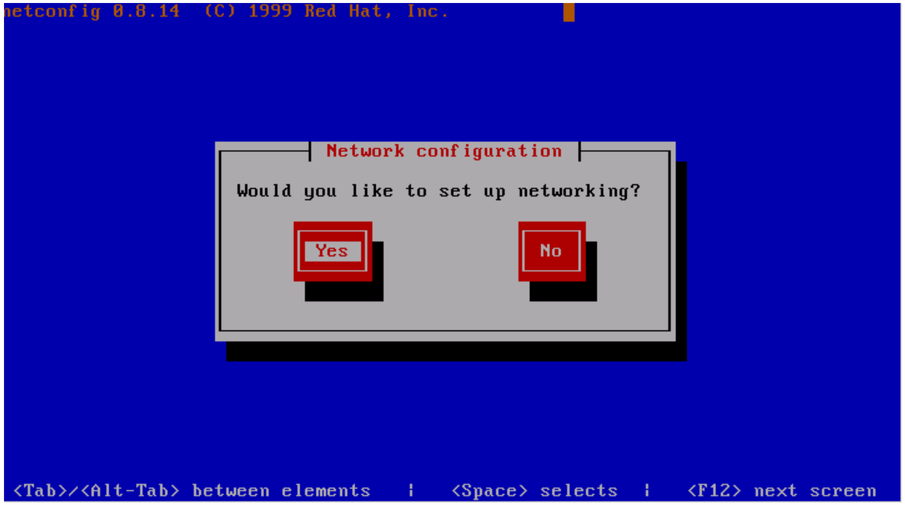
Name	IP Address
79c970 [PCnet32 LANCE]	NONE

79c970 [PCnet32 LANCE]
MAC : 00:0c:29:62:6f:70

Application Launcher
> Computer
> YaST
> YaST Control Center
> Network Devices
> Network Settings

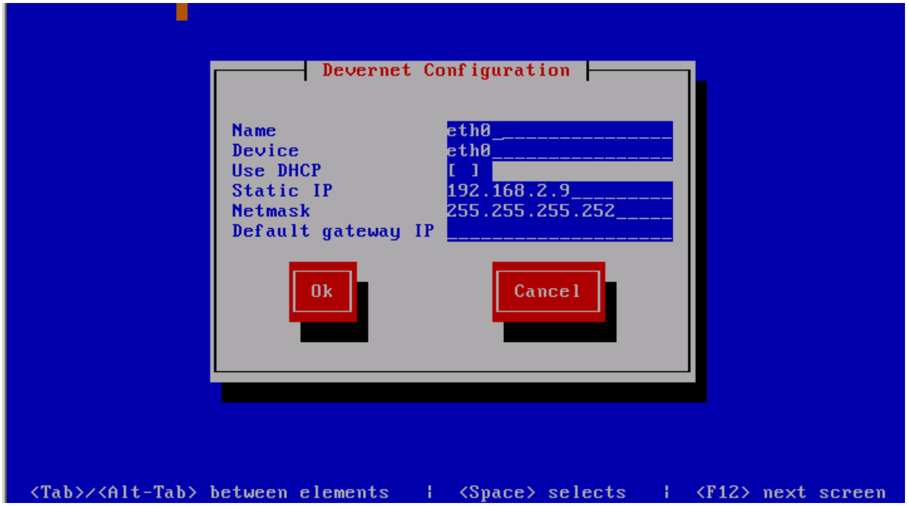
The UNIX/Linux customers first question was always: That a very pretty interface but I need to know exactly what commands you are calling underneath!

TUI (Red Hat Family) method



```
netconfig 0.8.14  (C) 1999 Red Hat, Inc.

              Network configuration

       Would you like to set up networking?

 <Tab>/<Alt-Tab> between elements   |   <Space> selects   |   <F12> next screen
```

*The **netconfig** command on Red Hat 9 provides a TUI interface to set up the basic network settings.*



```
              Devernet Configuration

   Name                  eth0_________________
   Device                eth0_________________
   Use DHCP              [ ]
   Static IP             192.168.2.9__________
   Netmask               255.255.255.252______
   Default gateway IP    _____________________

  <Tab>/<Alt-Tab> between elements   |   <Space> selects   |   <F12> next screen
```

*The **system-config-network** command replaces **netconfig** on CentOS 5.4.*

Command/Configuration File methods

Temporary (Commands)

- ifconfig
- route
- dhclient

These commands work across all distros.

However they are ***temporary*** *in that they only stay in effect till the system or the network service is restarted.*

Permanent (Configuration files)

- /etc/hosts
- /etc/resolv.conf

- Red Hat family:
    - /etc/sysconfig/network
    - /etc/sysconfig/network-scripts/ifcfg-eth*
    - **service network restart**

- Ubuntu family:
    - /etc/hostname
    - /etc/network/interfaces
    - **/etc/init.d/networking restart**

- OpenSUSE family
    - /etc/HOSTNAME
    - /etc/sysconfig/network/ifcfg-eth*
    - **rcnetwork restart**

These settings are ***permanent.***

However they don't take effect until the system or the network service is restarted

Review

Joining a network (temporarily via DHCP)

Command Line Method

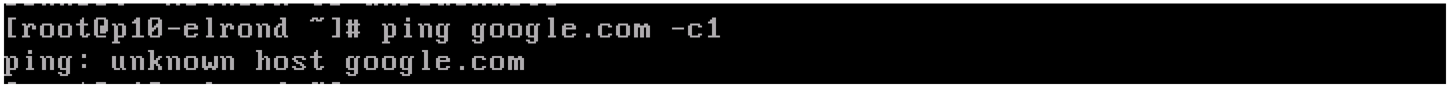
(Temporary - all distros)

	Dynamic (via DHCP server)
IP and subnet mask	**dhclient -v eth***n* *to obtain network settings* **dhclient -r eth***n* *to release network settings*
Default gateway	
DNS	

This is a quick and easy way to join a network as long as there is a DHCP server available

Joining a network temporarily via DHCP example

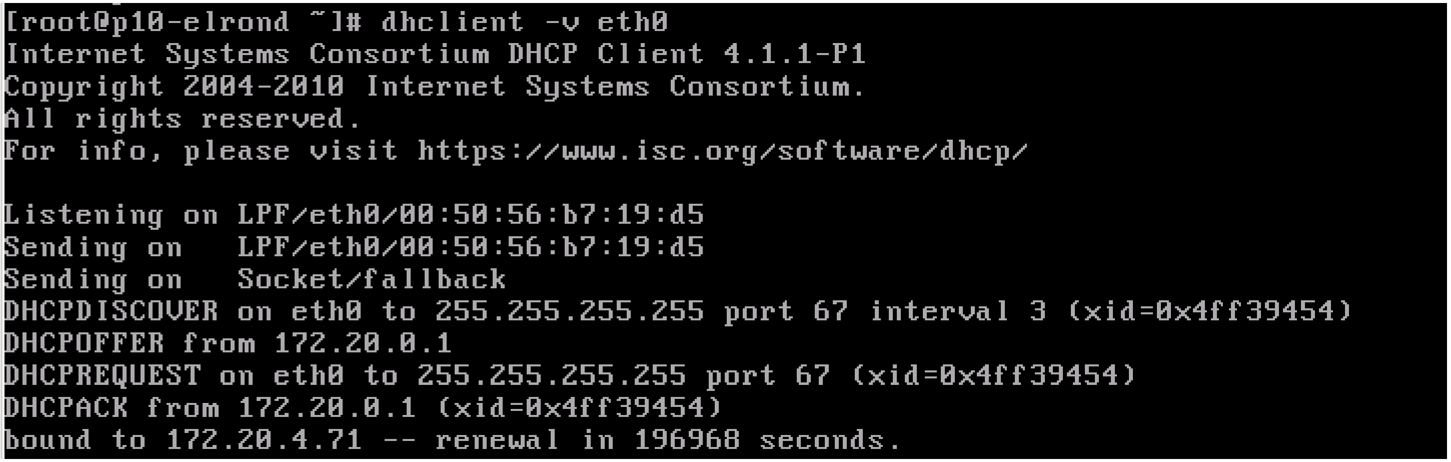
ping google.com -c1



```
[root@p10-elrond ~]# ping google.com -c1
ping: unknown host google.com
```

No connectivity!

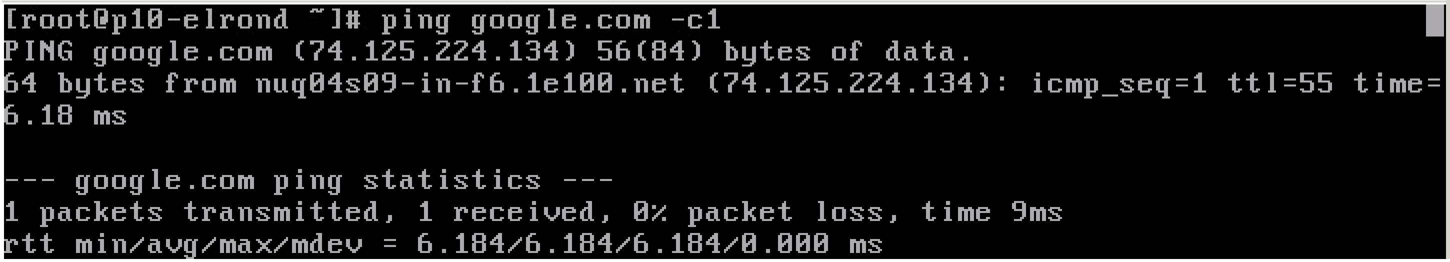
dhclient -v eth0



```
[root@p10-elrond ~]# dhclient -v eth0
Internet Systems Consortium DHCP Client 4.1.1-P1
Copyright 2004-2010 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/00:50:56:b7:19:d5
Sending on   LPF/eth0/00:50:56:b7:19:d5
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 3 (xid=0x4ff39454)
DHCPOFFER from 172.20.0.1
DHCPREQUEST on eth0 to 255.255.255.255 port 67 (xid=0x4ff39454)
DHCPACK from 172.20.0.1 (xid=0x4ff39454)
bound to 172.20.4.71 -- renewal in 196968 seconds.
```

Broadcasting a request and getting network settings from a DHCP server

ping google.com -c1

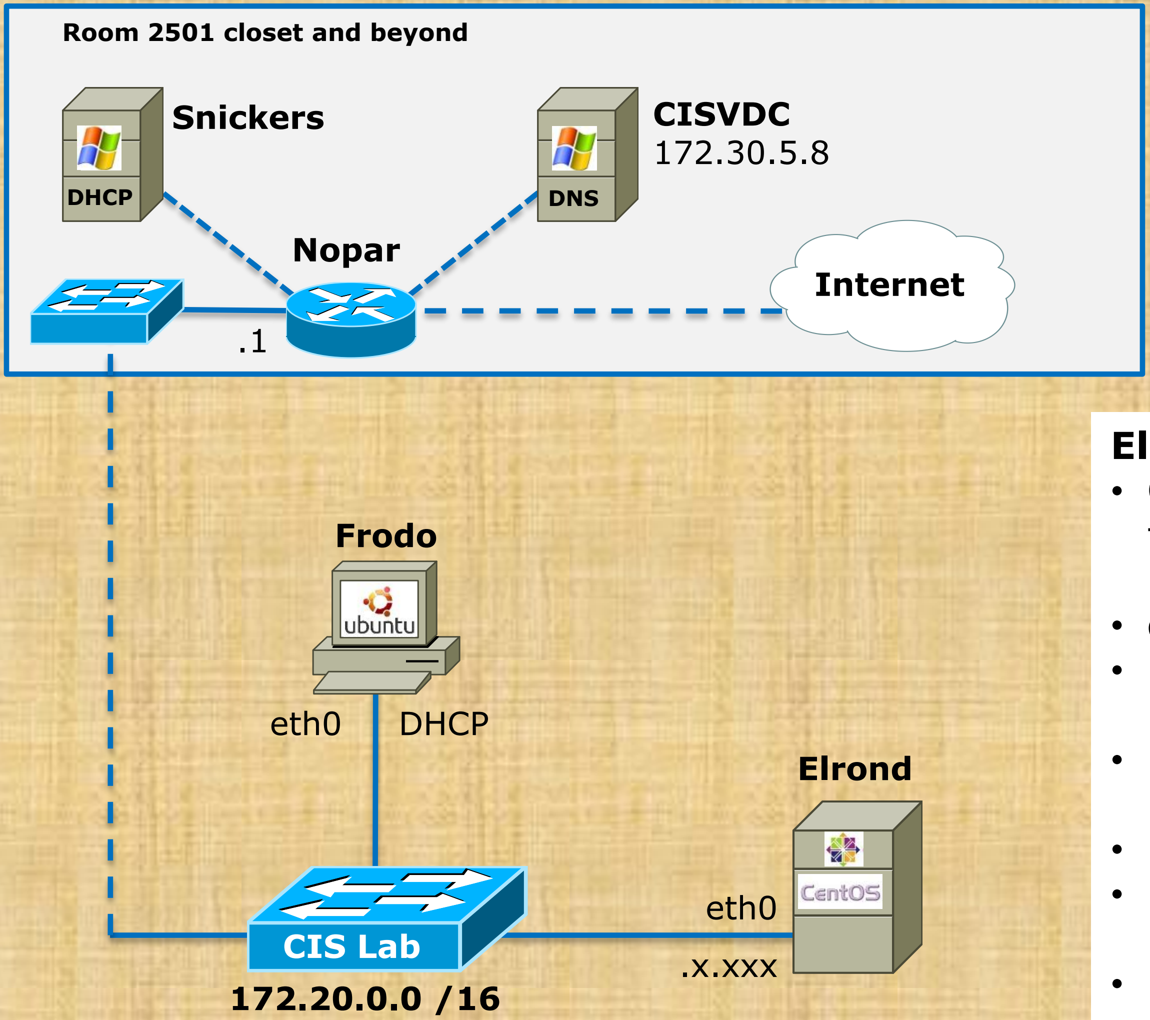


```
[root@p10-elrond ~]# ping google.com -c1
PING google.com (74.125.224.134) 56(84) bytes of data.
64 bytes from nuq04s09-in-f6.1e100.net (74.125.224.134): icmp_seq=1 ttl=55 time=
6.18 ms

--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 9ms
rtt min/avg/max/mdev = 6.184/6.184/6.184/0.000 ms
```

We have connectivity!

Class Activity

Join Elrond temporarily to the CIS Lab network using DHCP



Elrond

- Cable Elrond's eth0 interface to the CIS Lab network

- **dhclient -v eth0**
- **ifconfig eth0**

- **ping google.com**

- **init 6**
- Login again

- **ping google.com**

Review

Joining a network (temporarily via static IP)

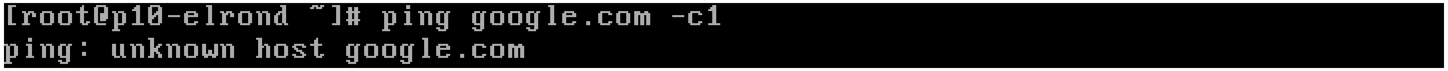
Command Line Method
(Temporary - all distros)

	Static IP Address
IP and subnet mask	**ifconfig eth***n xxx.xxx.xxx.xxx/pp*
Default gateway	**route add default gw** *xxx.xxx.xxx.xxx* **route del default gw** *xxx.xxx.xxx.xxx*
DNS	*add nameservers to* **/etc/resolv.conf**
Hostname	**hostname** *xxxxxxx*

If you manually configure a static IP address you must make sure it is not a duplicate!

Joining a network temporarily via static IP example

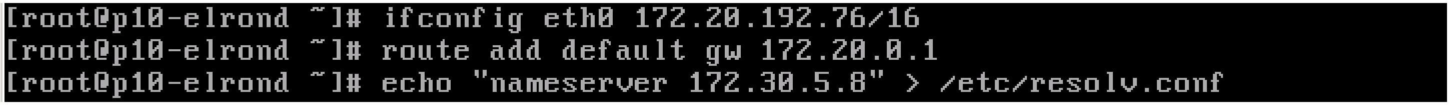
ping google.com -c1



```
[root@p10-elrond ~]# ping google.com -c1
ping: unknown host google.com
```

No connectivity!

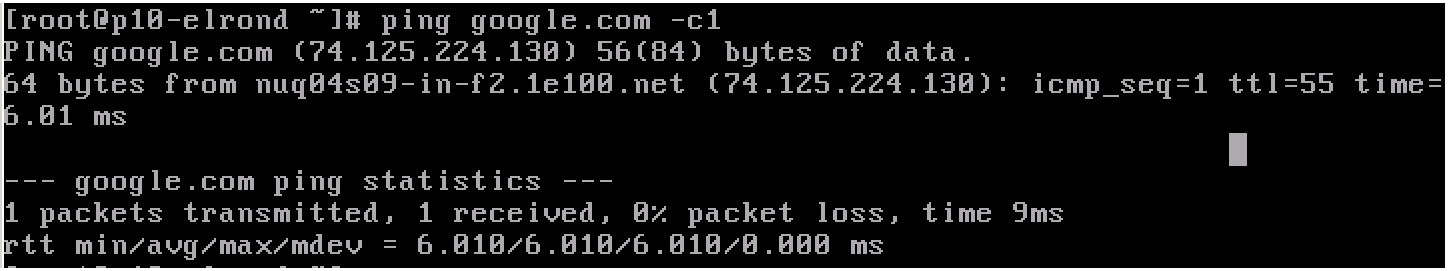
ifconfig eth0 172.20.192.76/16
route add default gw 172.20.0.1
echo "nameserver 172.30.5.8" > /etc/resolv.conf



```
[root@p10-elrond ~]# ifconfig eth0 172.20.192.76/16
[root@p10-elrond ~]# route add default gw 172.20.0.1
[root@p10-elrond ~]# echo "nameserver 172.30.5.8" > /etc/resolv.conf
```

Configure static IP address, default gateway and DNS name server

ping google.com -c1

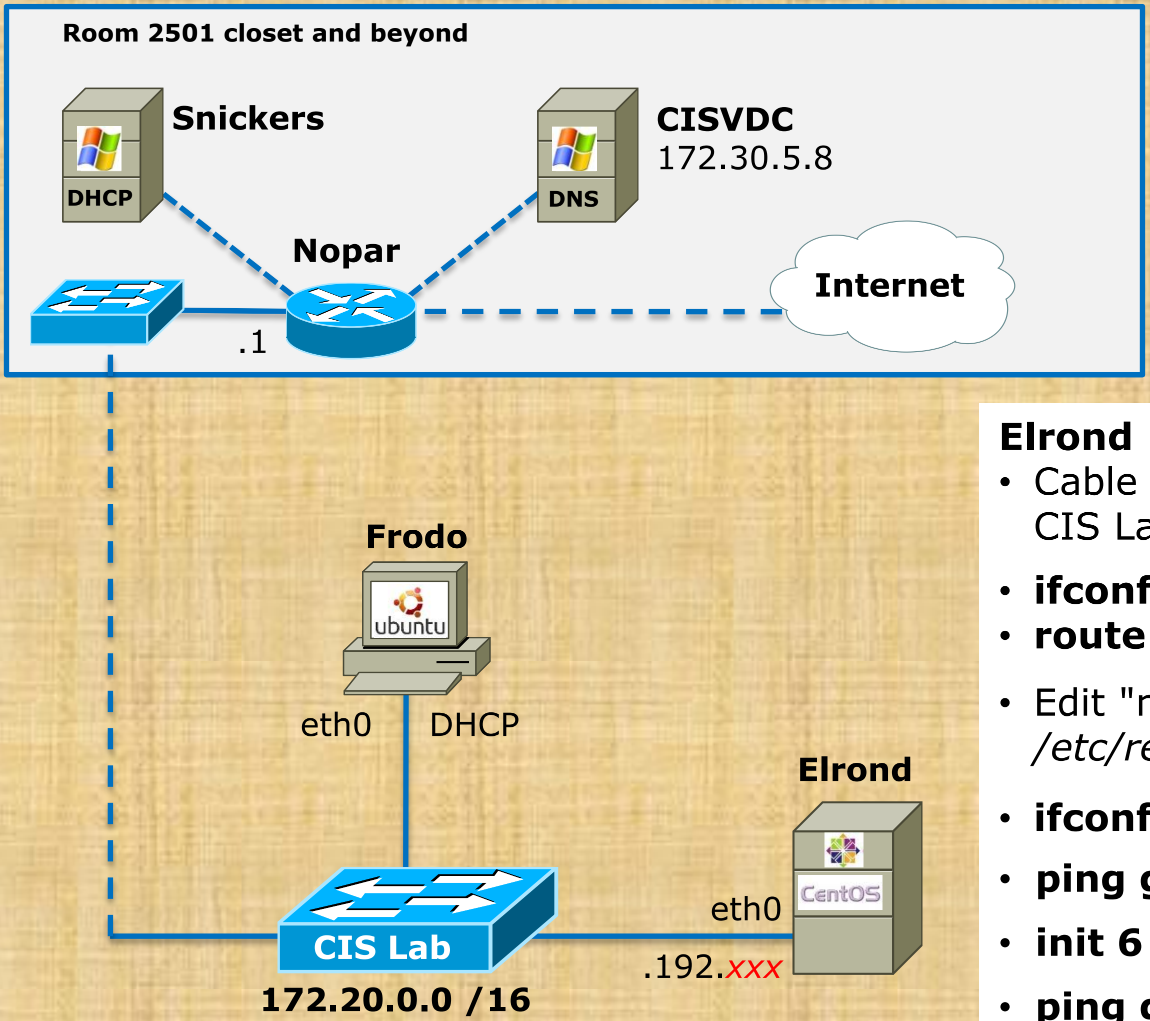


```
[root@p10-elrond ~]# ping google.com -c1
PING google.com (74.125.224.130) 56(84) bytes of data.
64 bytes from nuq04s09-in-f2.1e100.net (74.125.224.130): icmp_seq=1 ttl=55 time=
6.01 ms

--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 9ms
rtt min/avg/max/mdev = 6.010/6.010/6.010/0.000 ms
```

We have connectivity!

Class Activity

Join Elrond temporarily to the CIS Lab network using static IP



VLab RDP file

CIS 90 VLab VM Assignements

CIS 192 VLab Pod Assignements

No DUPS please!

Elrond

- Cable Elrond's eth0 to the CIS Lab network
- **ifconfig eth0 172.20.192.xxx/16**
- **route add default gw 172.20.0.1**
- Edit "nameserver 172.30.5.8" into */etc/resolv.conf*
- **ifconfig eth0**
- **ping google.com**
- **init 6** (and Login again)
- **ping google.com**

Joining a network (permanently via DHCP)

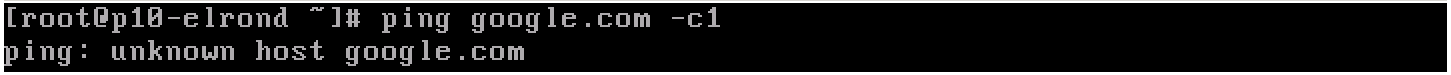
Joining a Network
(Permanent - Red Hat Family)

Area	Dynamic (permanent)
IP and subnet mask	**/etc/sysconfig/network-scripts/ifcfg-eth***n* **DEVICE="eth***n***"** **TYPE="Ethernet"** **NM_CONTROLLED="no"** **ONBOOT="yes"** **BOOTPROTO="dhcp"**
Default gateway	
DNS	

Use **service network restart** for changes to take effect

Joining a network permanently via DHCP example

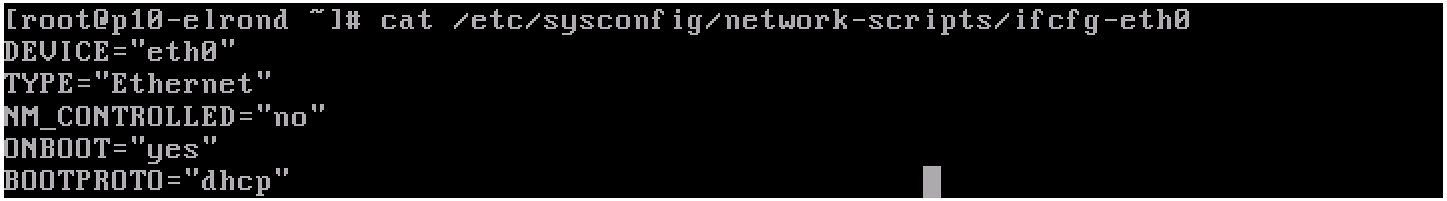
ping google.com -c1



```
[root@p10-elrond ~]# ping google.com -c1
ping: unknown host google.com
```

No connectivity!

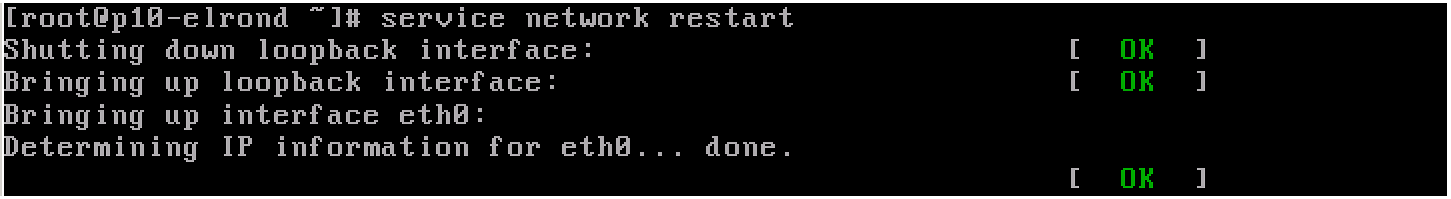
Edit */etc/sysconfig/network-scripts/ifcfg-eth0* to contain:



```
[root@p10-elrond ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
TYPE="Ethernet"
NM_CONTROLLED="no"
ONBOOT="yes"
BOOTPROTO="dhcp"
```

Configure DHCP at startup time and disable network manager

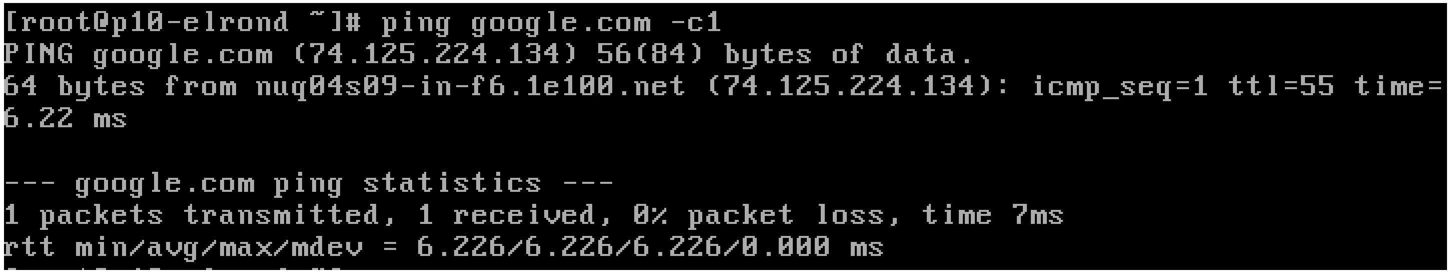
service network restart



```
[root@p10-elrond ~]# service network restart
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:
Determining IP information for eth0... done.
                                                           [  OK  ]
```

Restart the network service

ping google.com -c1

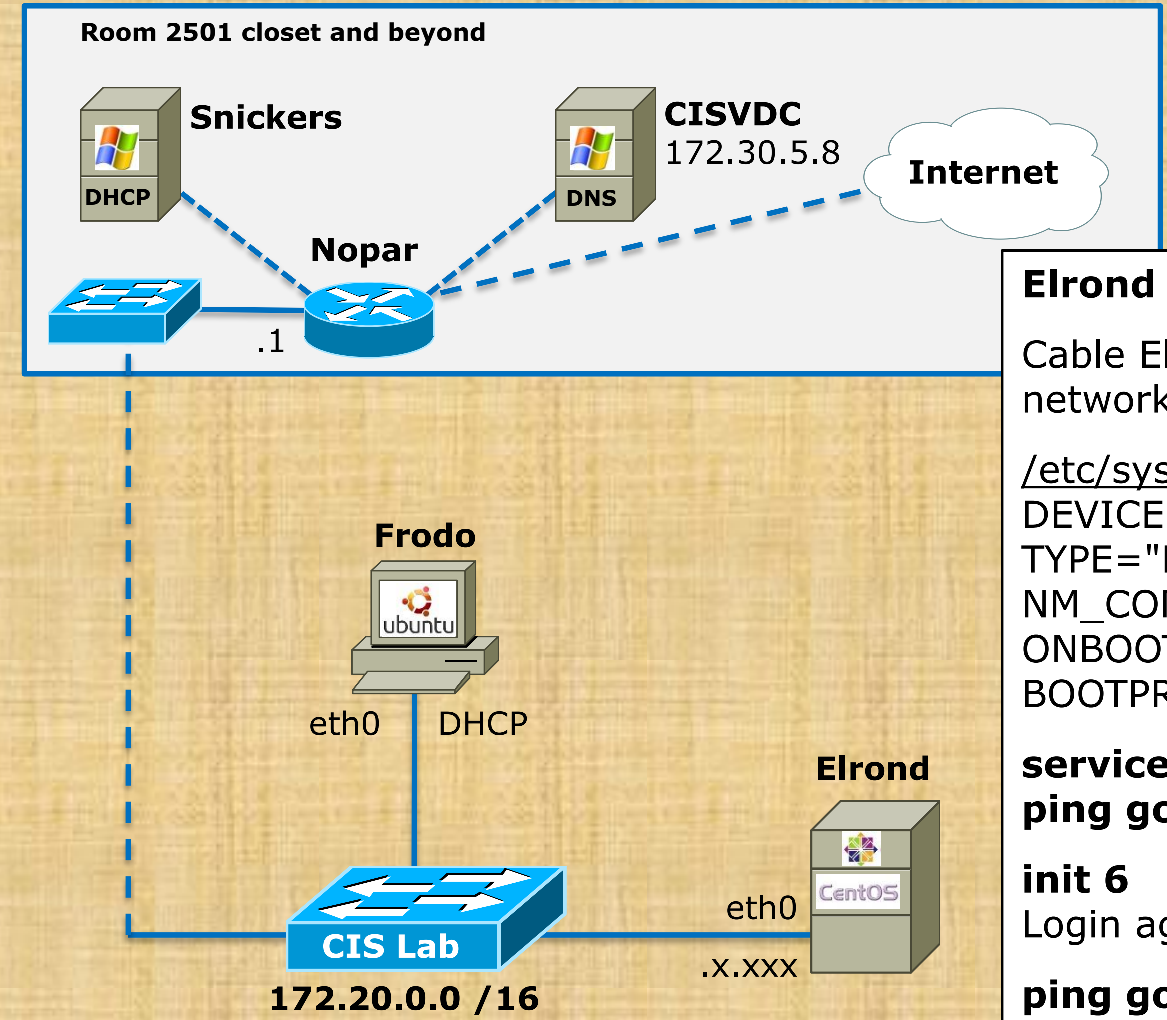


```
[root@p10-elrond ~]# ping google.com -c1
PING google.com (74.125.224.134) 56(84) bytes of data.
64 bytes from nuq04s09-in-f6.1e100.net (74.125.224.134): icmp_seq=1 ttl=55 time=
6.22 ms

--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 7ms
rtt min/avg/max/mdev = 6.226/6.226/6.226/0.000 ms
```

We have connectivity!

Class Activity

Join Elrond permanently to the CIS Lab network using DHCP



Elrond

Cable Elrond's eth0 interface to the CIS Lab network

```
/etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
TYPE="Ethernet"
NM_CONTROLLED="no"
ONBOOT="yes"
BOOTPROTO="dhcp"
```

service network restart
ping google.com

init 6
Login again

ping google.com

Joining a network (permanently via static IP)

Joining a Network
(Permanent - Red Hat Family)

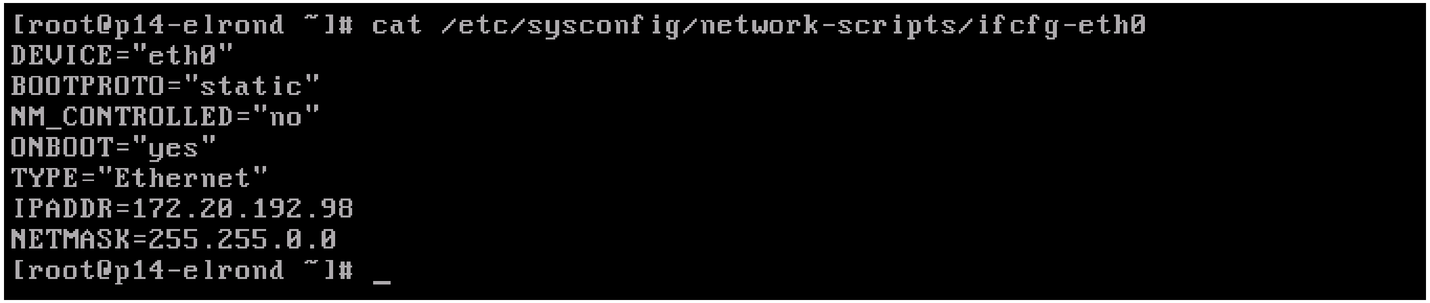
Settings kept in configuration files and used during the startup process

Area	Static (permanent)
IP and subnet mask	**/etc/sysconfig/network-scripts/ifcfg-eth***n* **DEVICE="eth***n***"** **NM_CONTROLLED="no"** **ONBOOT="yes"** **BOOTPROTO="static"** **IPADDR=***xxx.xxx.xxx.xxx* **NETMASK=***xxx.xxx.xxx.xxx*
Default gateway	**/etc/sysconfig/network** **NETWORKING=yes** **HOSTNAME=***name.domain* **GATEWAY=***xxx.xxx.xxx.xxx*
DNS	**/etc/resolv.conf** **nameserver** *xxx.xxx.xxx.xxx* **nameserver** *xxx.xxx.xxx.xxx*

Use **service network restart** for changes to take effect

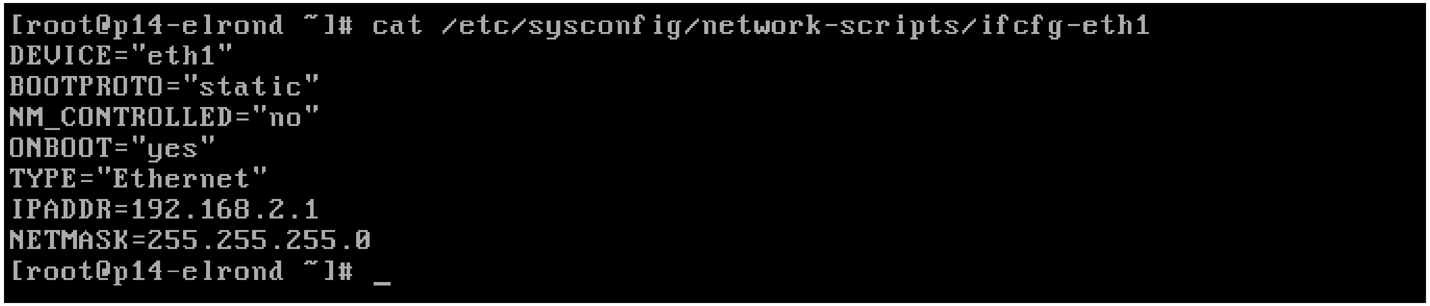
Joining a network permanently via static IP example

Example: Permanently configure both interfaces on Elrond for Lab 02 (VLab Pod 14)



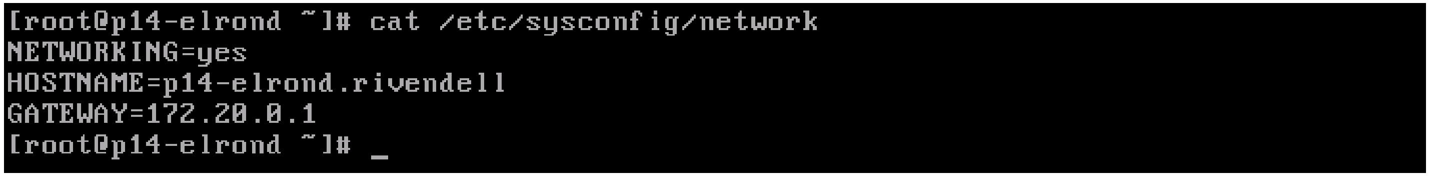
```
[root@p14-elrond ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
BOOTPROTO="static"
NM_CONTROLLED="no"
ONBOOT="yes"
TYPE="Ethernet"
IPADDR=172.20.192.98
NETMASK=255.255.0.0
[root@p14-elrond ~]# _
```

Configure eth0 with static IP on eth0



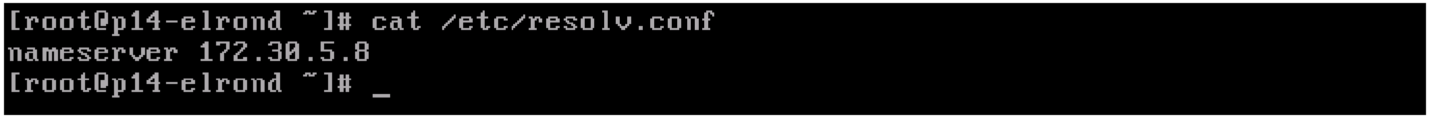
```
[root@p14-elrond ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE="eth1"
BOOTPROTO="static"
NM_CONTROLLED="no"
ONBOOT="yes"
TYPE="Ethernet"
IPADDR=192.168.2.1
NETMASK=255.255.255.0
[root@p14-elrond ~]# _
```

Configure eth0 with static IP on eth1



```
[root@p14-elrond ~]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=p14-elrond.rivendell
GATEWAY=172.20.0.1
[root@p14-elrond ~]# _
```

Configure default gateway

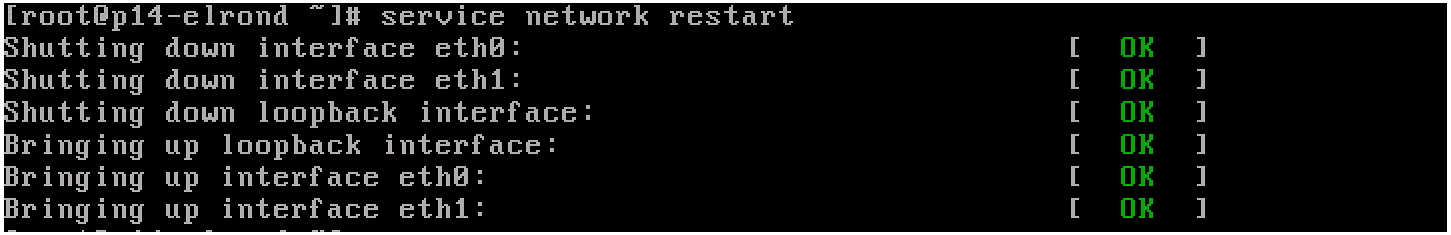


```
[root@p14-elrond ~]# cat /etc/resolv.conf
nameserver 172.30.5.8
[root@p14-elrond ~]# _
```

Configure DNS name server

Joining a network permanently via static IP example

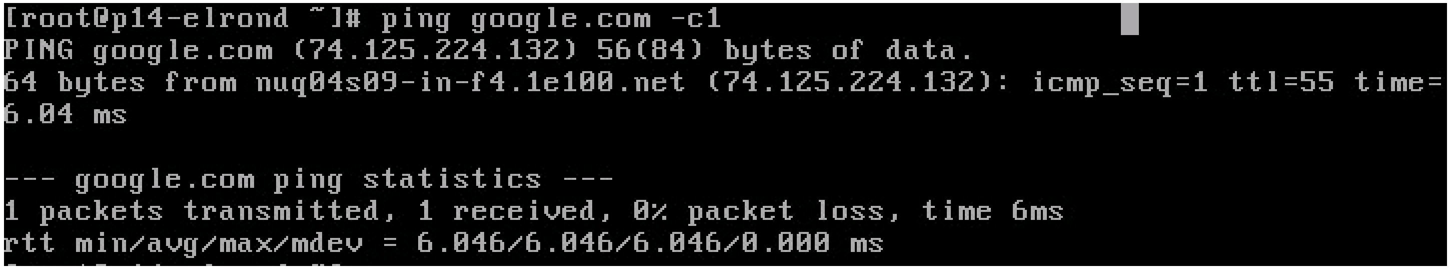
service network restart



```
[root@p14-elrond ~]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down interface eth1:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:                                [  OK  ]
Bringing up interface eth1:                                [  OK  ]
```

Restart the network service which re-reads the network configuration files

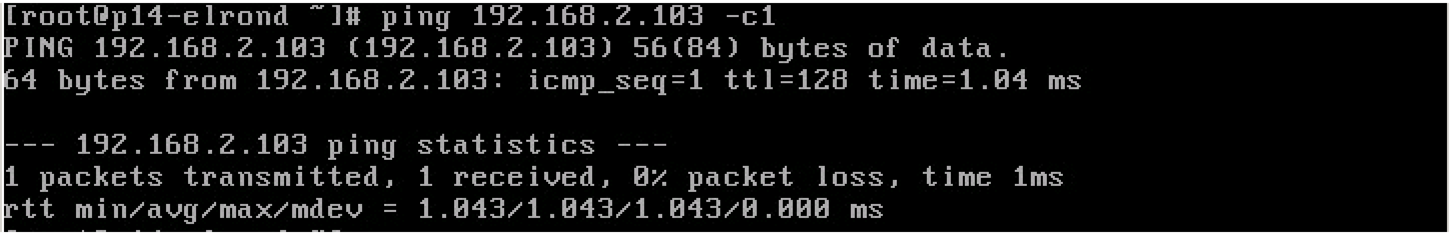
ping google.com -c1



```
[root@p14-elrond ~]# ping google.com -c1
PING google.com (74.125.224.132) 56(84) bytes of data.
64 bytes from nuq04s09-in-f4.1e100.net (74.125.224.132): icmp_seq=1 ttl=55 time=
6.04 ms

--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 6ms
rtt min/avg/max/mdev = 6.046/6.046/6.046/0.000 ms
```

Ping Internet host by name

ping 192.168.2.103 -c1

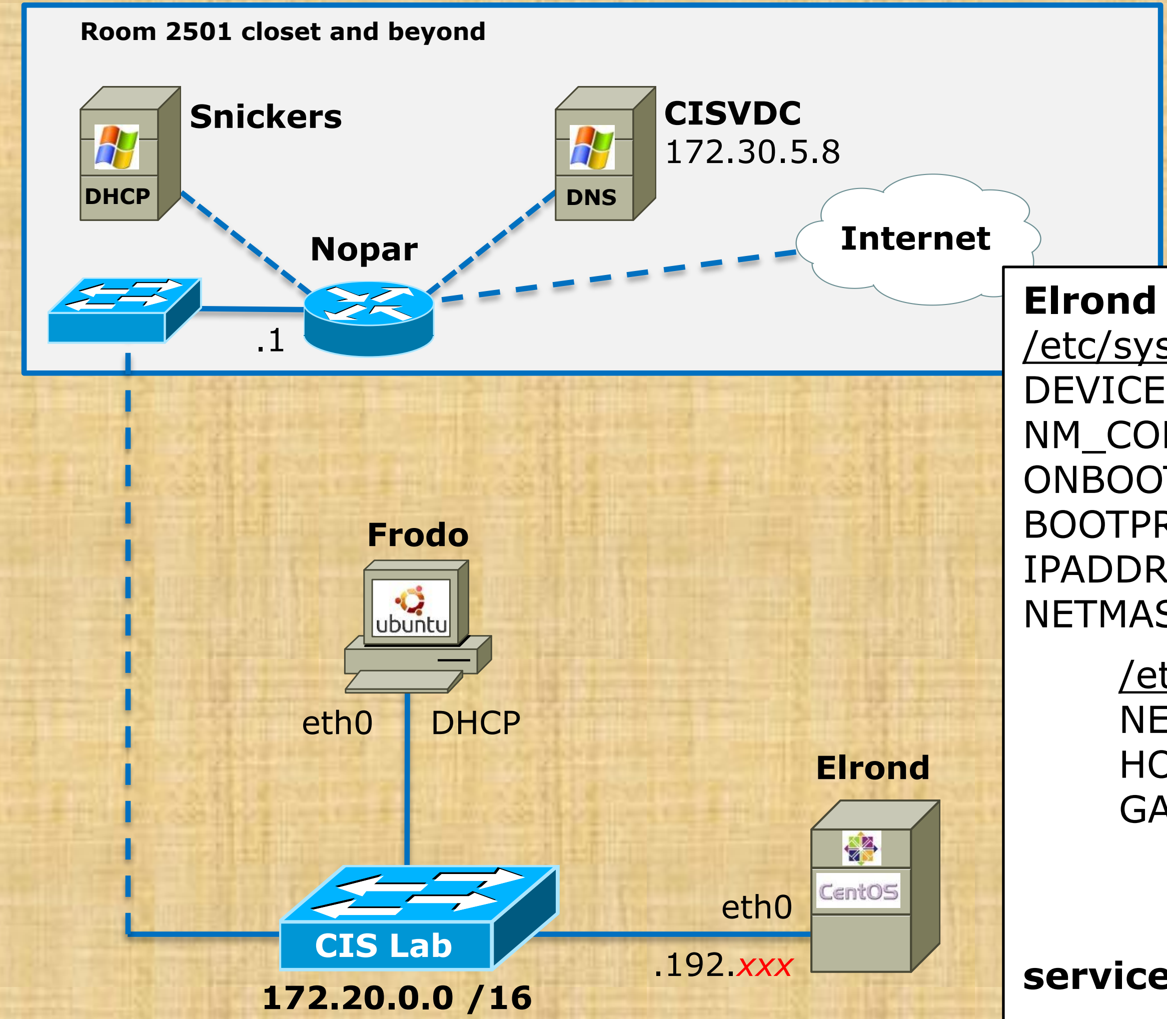


```
[root@p14-elrond ~]# ping 192.168.2.103 -c1
PING 192.168.2.103 (192.168.2.103) 56(84) bytes of data.
64 bytes from 192.168.2.103: icmp_seq=1 ttl=128 time=1.04 ms

--- 192.168.2.103 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 1ms
rtt min/avg/max/mdev = 1.043/1.043/1.043/0.000 ms
```

Ping local host on private network

Class Activity

Join Elrond permanently to the CIS Lab network using static IP



VLab RDP file

CIS 90 VLab VM Assignements

CIS 192 VLab Pod Assignements

No DUPS Please!

Elrond

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

```
DEVICE="eth0"
NM_CONTROLLED="no"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=172.20.192.xxx
NETMASK=255.255.0.0
```

/etc/sysconfig/network

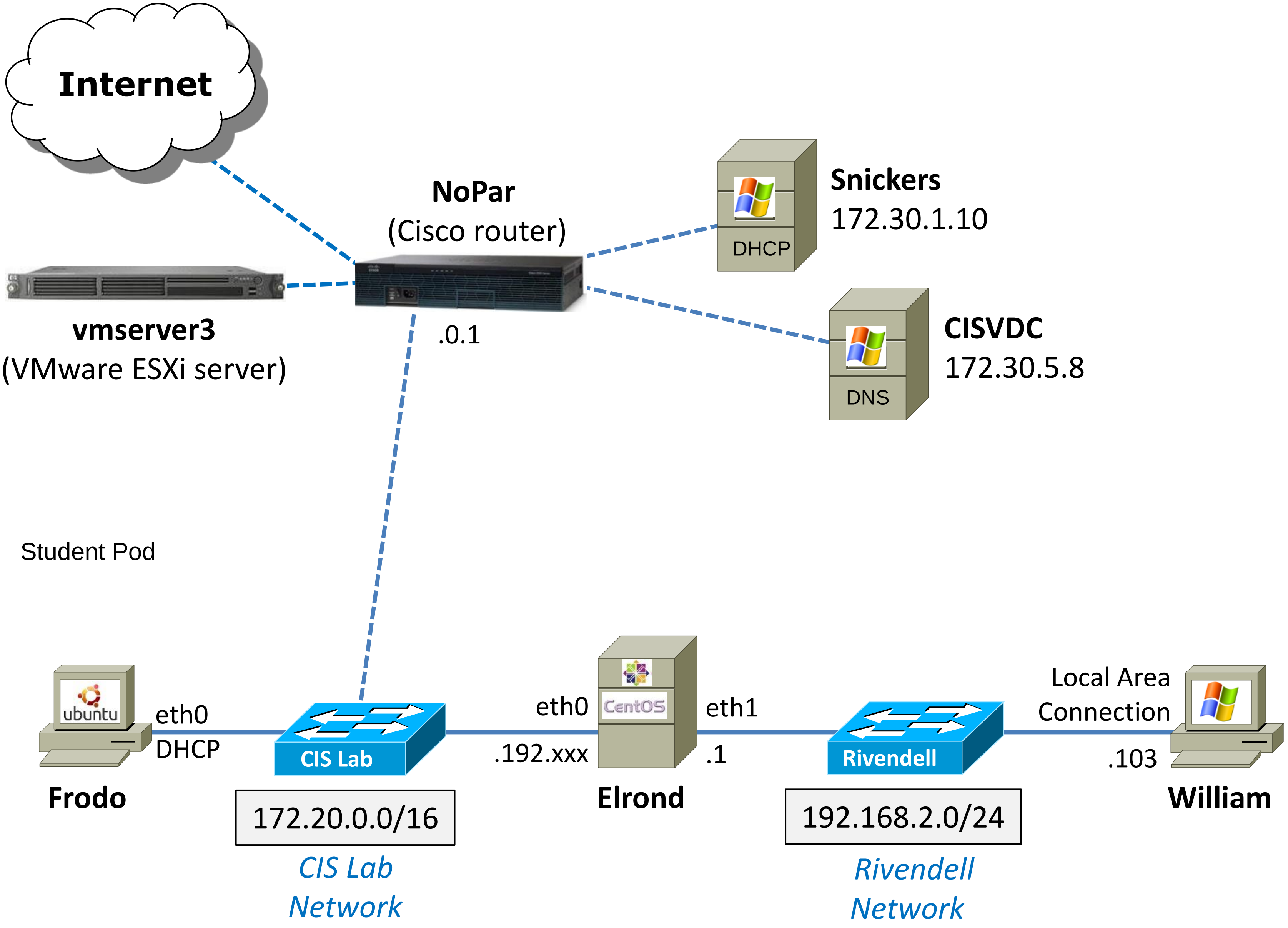
```
NETWORKING=yes
HOSTNAME=pxx-elrond.rivendell
GATEWAY=172.20.0.1
```

/etc/resolv.conf

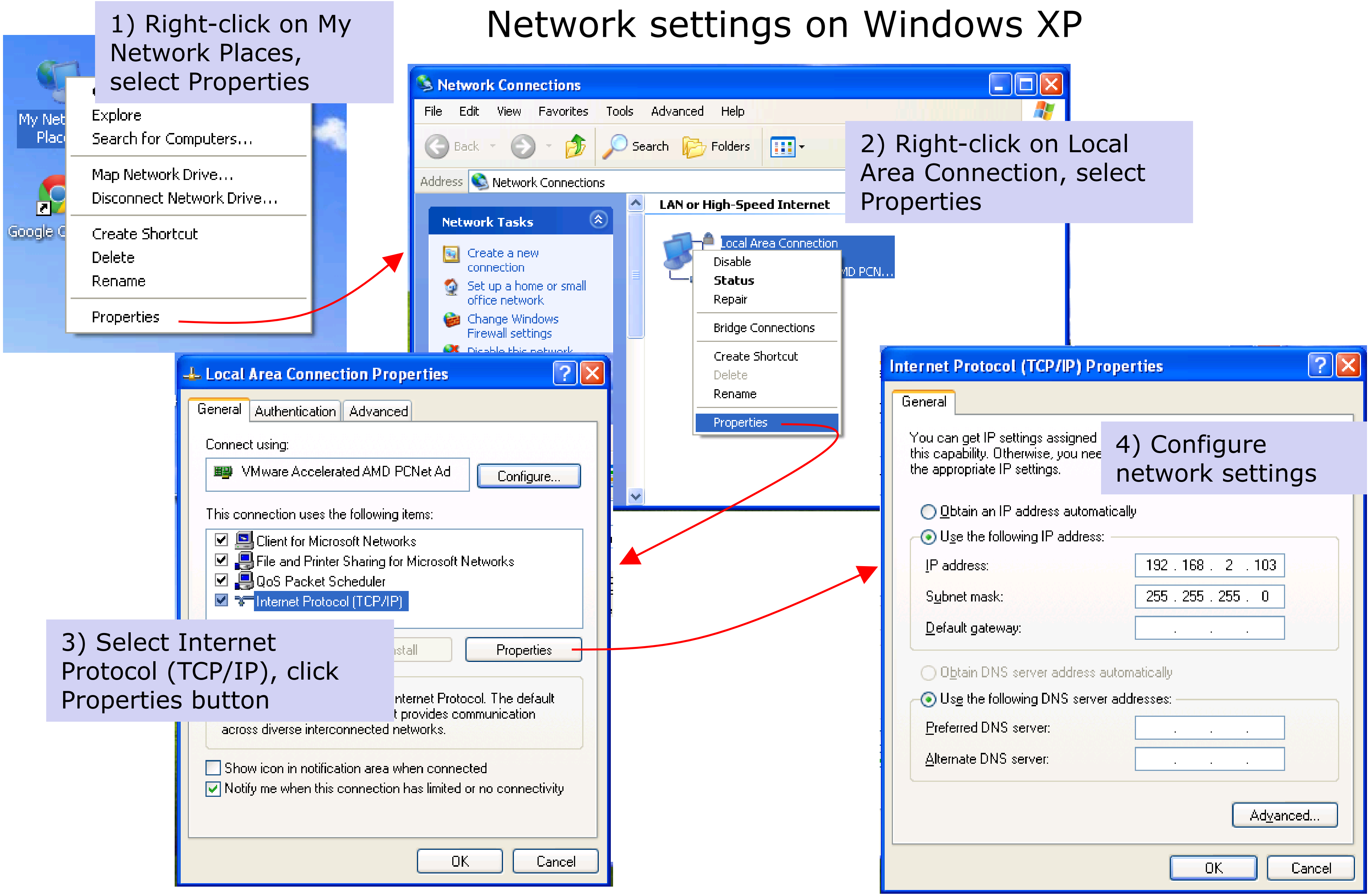
```
nameserver 172.30.5.8
```

service network restart

Windows XP network configuration


Internet
NoPar
(Cisco router)
vmserver3
(VMware ESXi server)
.0.1
Snickers
172.30.1.10
DHCP
CISVDC
172.30.5.8
DNS
Student Pod
ubuntu
eth0
DHCP
Frodo
CIS Lab
172.20.0.0/16
CIS Lab
Network
eth0
.192.xxx
CentOS
eth1
.1
Elrond
Rivendell
192.168.2.0/24
Rivendell
Network
Local Area
Connection
.103
William

Network settings on Windows XP



1) Right-click on My Network Places, select Properties

2) Right-click on Local Area Connection, select Properties

3) Select Internet Protocol (TCP/IP), click Properties button

4) Configure network settings

Local Area Connection Properties

Connect using:

VMware Accelerated AMD PCNet Ad

This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ Internet Protocol (TCP/IP)

nternet Protocol. The default t provides communication across diverse interconnected networks.

□ Show icon in notification area when connected

☑ Notify me when this connection has limited or no connectivity

Internet Protocol (TCP/IP) Properties

You can get IP settings assigned this capability. Otherwise, you nee the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

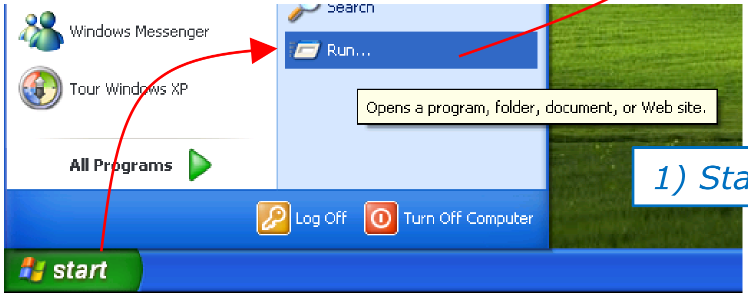
IP address:	192 . 168 . 2 . 103
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	. . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server:	. . .
Alternate DNS server:	. . .

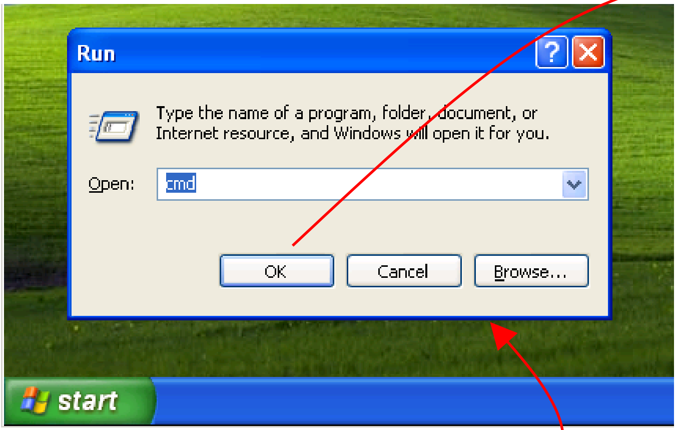
Getting a command line on Windows XP



Opens a program, folder, document, or Web site.

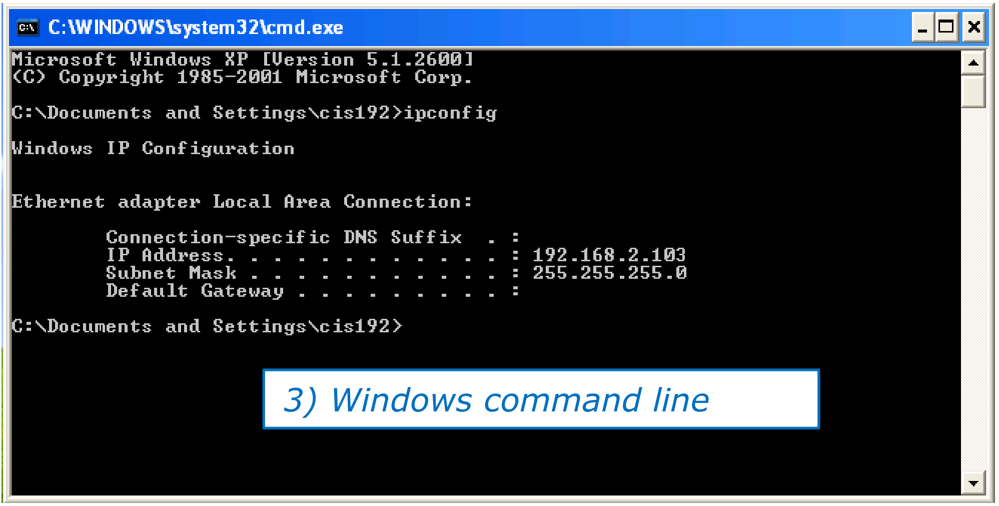
1) Start > Run ...

2) Run cmd



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd



C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\cis192>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.2.103
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :

C:\Documents and Settings\cis192>
```

3) Windows command line

Class Exercise
Windows XP network settings and command line

Try it!

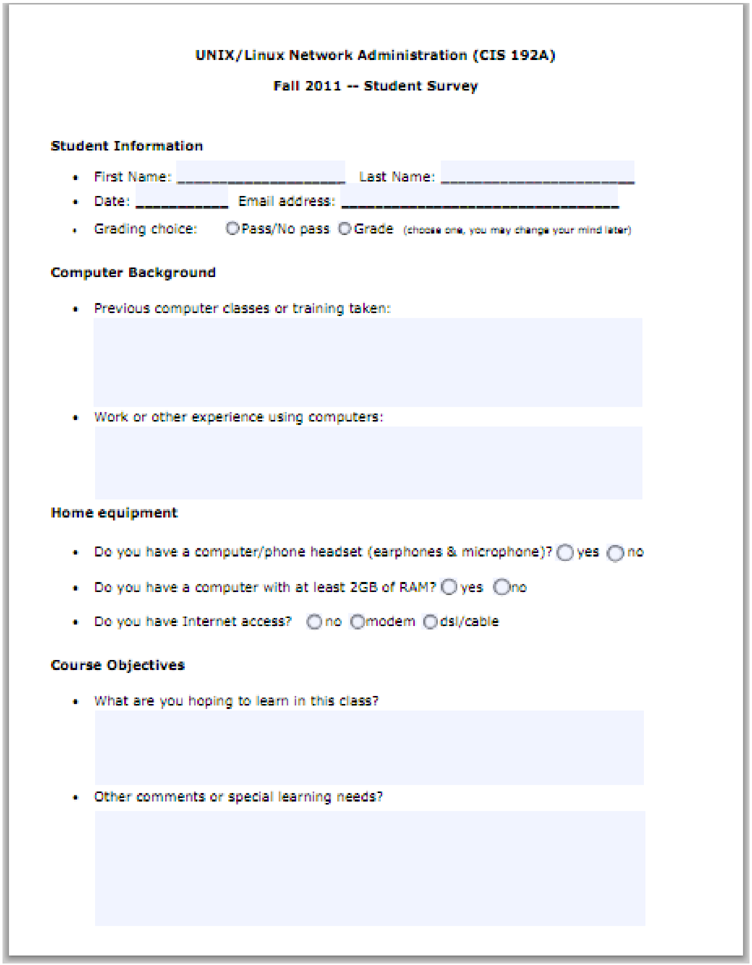
- *Cable William to Rivendell*
- *IP = 192.168.2.111*
- *Subnet mask = 255.255.255.0*
- *Run ipconfig to verify*

Housekeeping

- Lab 1 is due by 11:59PM tonight (Opus time)
- Quick check on /home/rsimms/turnin/cis192/ on Opus
- Adds - Last day to add is 2/23!

Student Survey

http://simms-teach.com/docs/cis192/cis192survey.pdf



UNIX/Linux Network Administration (CIS 192A)

Fall 2011 -- Student Survey

Student Information

- First Name: ____________________ Last Name: _______________________
- Date: ___________ Email address: _________________________________
- Grading choice: ○Pass/No pass ○Grade (choose one, you may change your mind later)

Computer Background

- Previous computer classes or training taken:
- Work or other experience using computers:

Home equipment

- Do you have a computer/phone headset (earphones & microphone)? ○ yes ○ no
- Do you have a computer with at least 2GB of RAM? ○ yes ○no
- Do you have Internet access? ○ no ○modem ○dsl/cable

Course Objectives

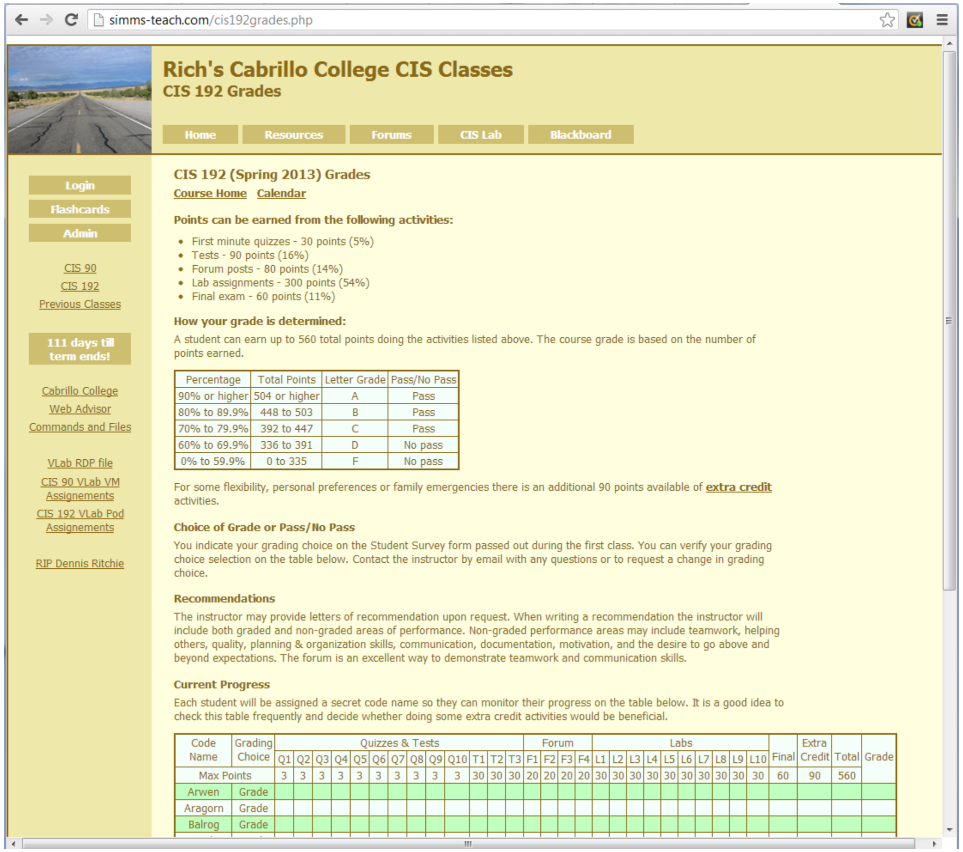
- What are you hoping to learn in this class?
- Other comments or special learning needs?

Email surveys to me at:
risimms@cabrillo.edu

CIS 192 - Code Names

Lord of the Rings Characters

http://simms-teach.com/cis192home.php



Rich's Cabrillo College CIS Classes

CIS 192 Grades

CIS 192 (Spring 2013) Grades

Course Home Calendar

Points can be earned from the following activities:

- First minute quizzes - 30 points (5%)
- Tests - 90 points (16%)
- Forum posts - 80 points (14%)
- Lab assignments - 300 points (54%)
- Final exam - 60 points (11%)

How your grade is determined:

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

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

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

Choice of Grade or Pass/No Pass

You indicate your grading choice on the Student Survey form passed out during the first class. You can verify your grading choice selection on the table below. Contact the instructor by email with any questions or to request a change in grading choice.

Recommendations

The instructor may provide letters of recommendation upon request. When writing a recommendation the instructor will include both graded and non-graded areas of performance. Non-graded performance areas may include teamwork, helping others, quality, planning & organization skills, communication, documentation, motivation, and the desire to go above and beyond expectations. The forum is an excellent way to demonstrate teamwork and communication skills.

Current Progress

Each student will be assigned a secret code name so they can monitor their progress on the table below. It is a good idea to check this table frequently and decide whether doing some extra credit activities would be beneficial.

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																															
Aragorn	Grade																															
Balrog	Grade																															

I'll be sending code names to everyone that sends me their survey

Roll Call

Turn off RECORDING

Switch to WB

CIS 192 - Lesson 2

Instructor: **Rich Simms**

Dial-in: **888-450-4821**

Passcode: **761867**

Solomon Sean C. Christopher Corey Bryan Sean F. Tony David Donna Stephanie

Dave Evan Gabriel Elia Tajvia Carlos Adam Ben Laura

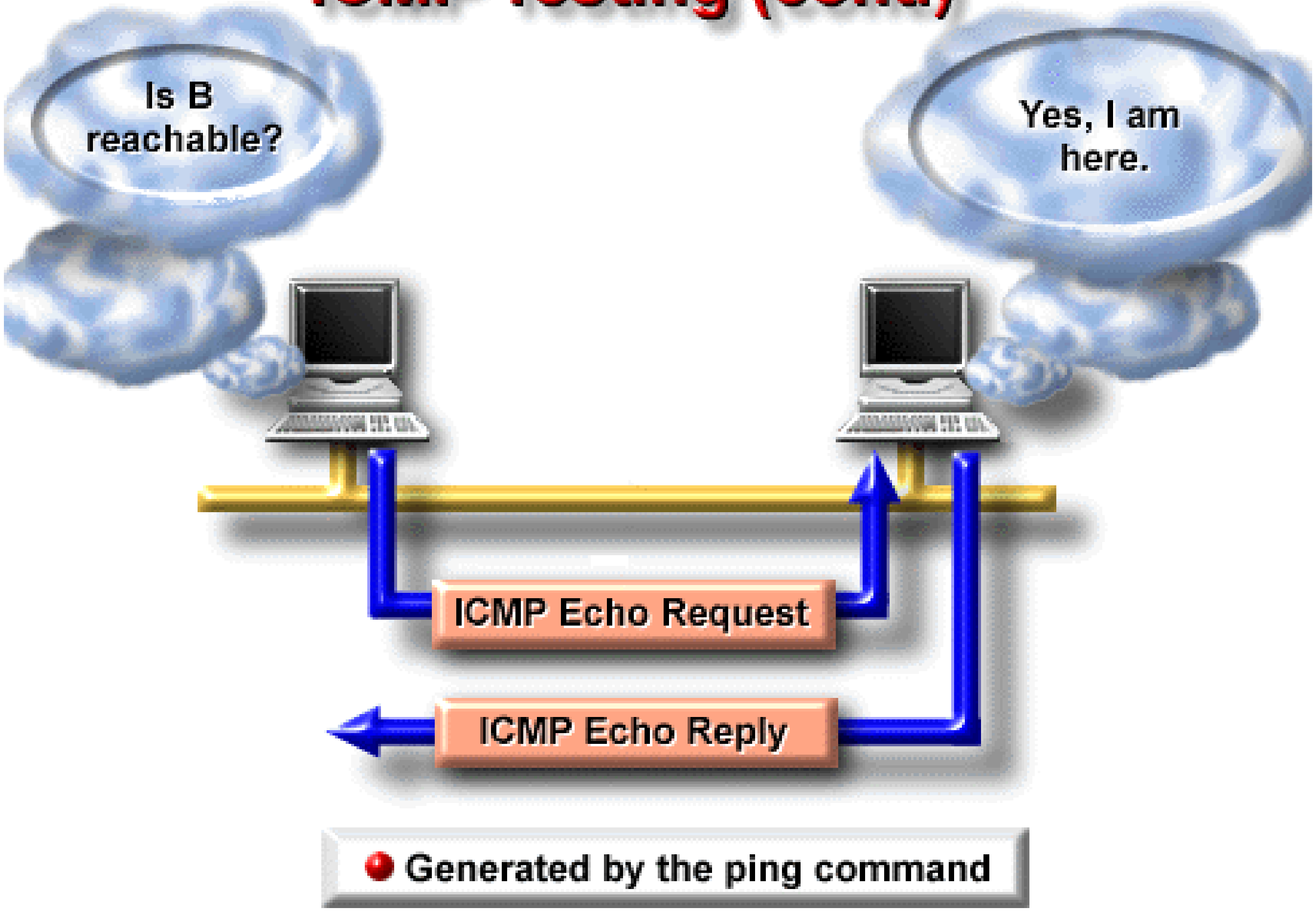
Email me (risimms@cabrillo.edu) a relatively current photo of your face for 3 points extra credit

Turn on RECORDING

Turn on RECORDING

Trouble shooting

ICMP Testing (cont.)



© Cisco Systems, Inc. 1999

Troubleshoot Network Connection

Follow these steps if your connection is not working:

1. Check **cabling**, **IP** and **subnet mask** settings by pinging another node on the same local network (which could be the router) using an IP address.

2. Check **default gateway** by pinging a node outside the local network using an IP address.

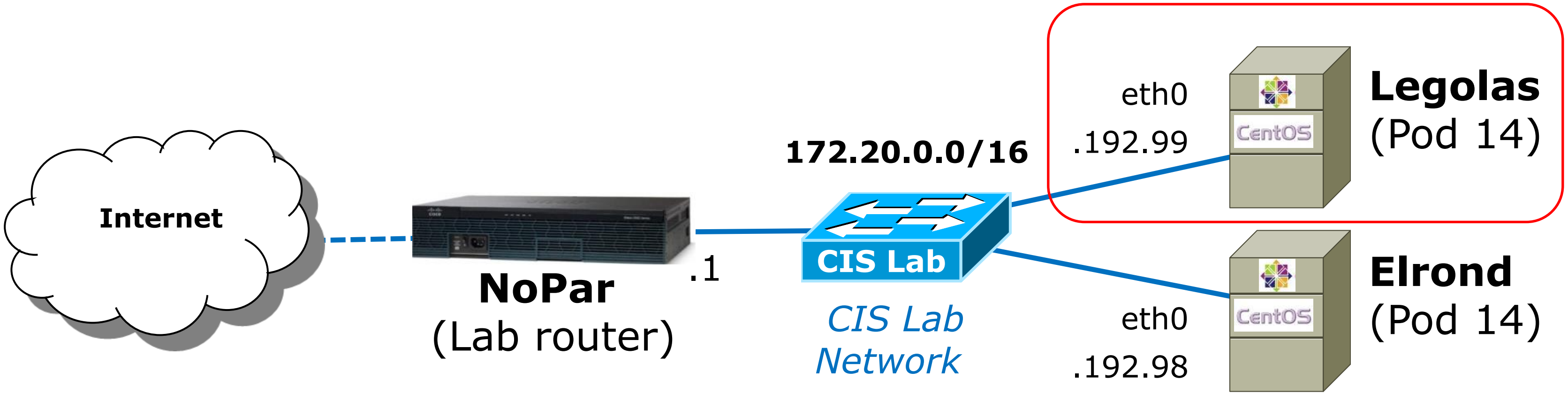
3. Check name resolution (**DNS namer server settings**) by pinging a node on the Internet by name.

Always work your way up the stack one layer at a time

Step 1 - local network

```
[root@legolas ~] ping -c4 172.20.0.1
PING 172.20.0.1 (172.20.0.1) 56(84) bytes of data.
64 bytes from 172.20.0.1: icmp_seq=1 ttl=255 time=3.90 ms
64 bytes from 172.20.0.1: icmp_seq=2 ttl=255 time=0.593 ms
64 bytes from 172.20.0.1: icmp_seq=3 ttl=255 time=0.596 ms
64 bytes from 172.20.0.1: icmp_seq=4 ttl=255 time=0.586 ms

--- 172.20.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.586/1.420/3.907/1.436 ms
```

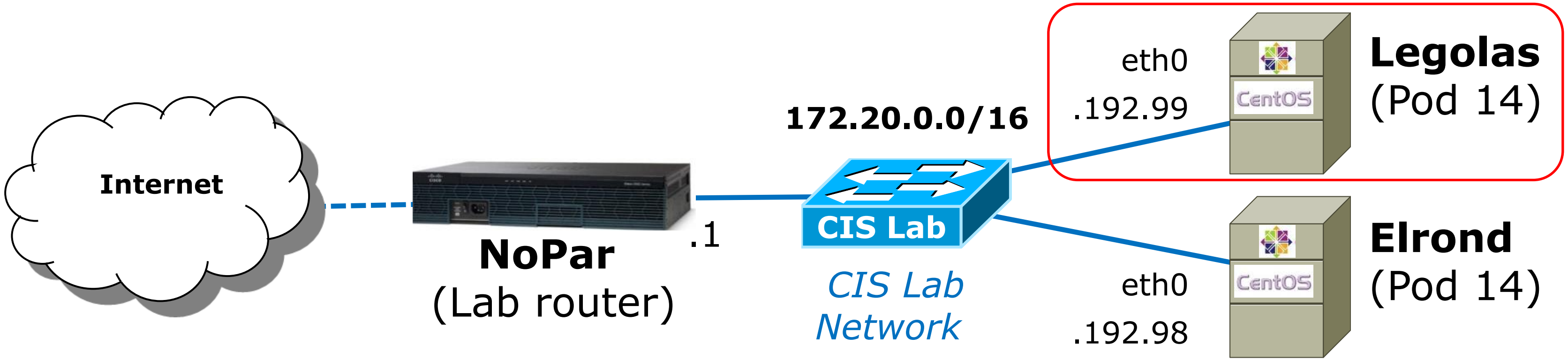
1) Successful ping of router's interface!



Step 1 - local network trouble

1) Check cabling, IP and subnet mask settings by pinging another node on the same local network (which could be the router) using an IP address.

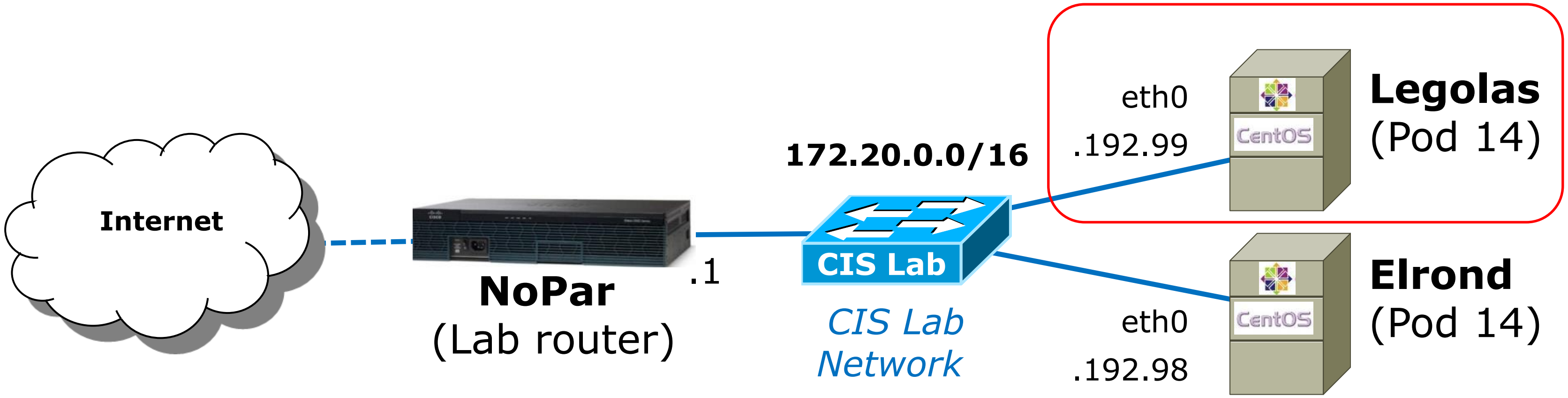
Ping test	Cabling	IP	Subnet mask	Default Gateway	DNS name servers	Ping results
ping 172.20.0.1	correct	correct	correct	correct	correct	Success
ping 172.20.0.1	Mordor (wrong network)	correct	correct	correct	correct	**Destination Host Unreachable**, 100% packet loss
ping 172.20.0.1	correct	172.20. 192.98 (DUP)	correct	correct	correct	Variable amount of **packet loss**. More loss when other node, Elrond, is active.
ping 172.20.0.1	correct	182.20. 192.99	correct	correct	correct	**Connect: Network is unreachable**



Step 2 - remote network host by IP

```
[root@legolas ~] ping -c4 10.240.1.2
PING 10.240.1.2 (10.240.1.2) 56(84) bytes of data.
64 bytes from 10.240.1.2: icmp_seq=1 ttl=62 time=1.65 ms
64 bytes from 10.240.1.2: icmp_seq=2 ttl=62 time=1.67 ms
64 bytes from 10.240.1.2: icmp_seq=3 ttl=62 time=1.11 ms
64 bytes from 10.240.1.2: icmp_seq=4 ttl=62 time=1.15 ms

--- 10.240.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.118/1.401/1.678/0.270 ms
```

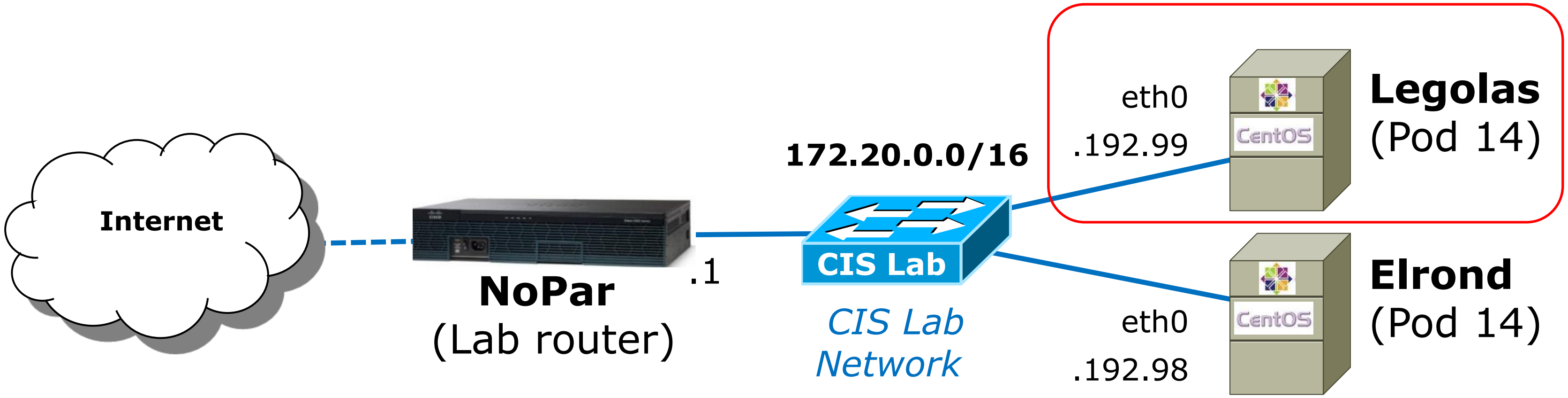
2) Successful ping of host on another network!



Step 2 - remote network host by IP trouble

2) Check default gateway by pinging a node outside the local network using an IP address.

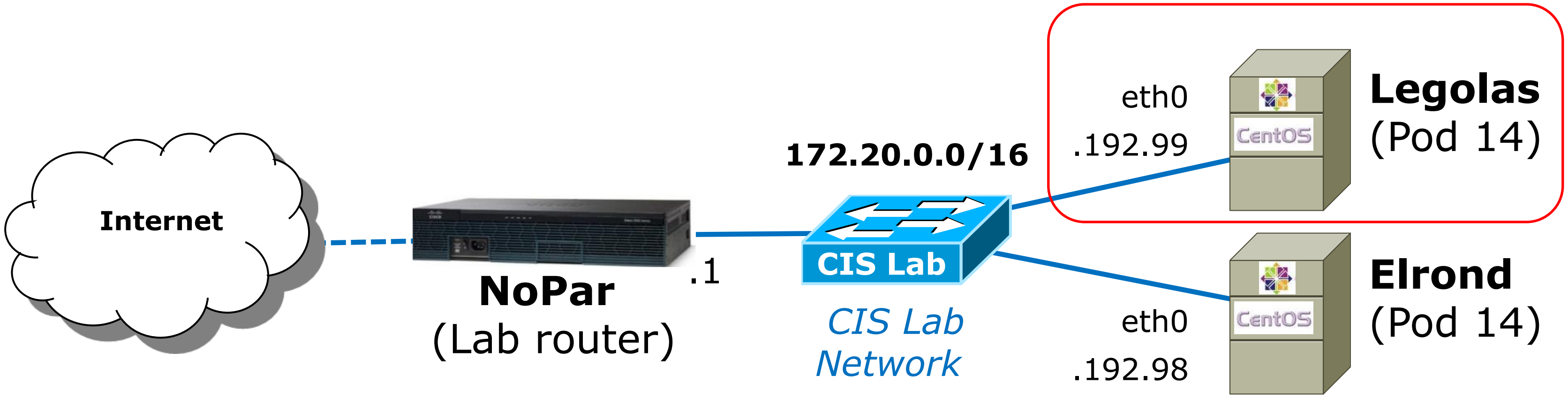
Ping test	Cabling	IP	Subnet mask	Default Gateway	DNS name servers	Ping results
ping 10.240.1.2	correct	correct	correct	correct	correct	Success
ping 10.240.1.2	correct	correct	correct	not added	correct	**connect: Network is unreachable**
ping 10.240.1.2	correct	correct	correct	non router specified	correct	no error message but **100% packet loss**



Step 3 - Internet host by name

```
[root@legolas ~] ping -c4 gogle.com
PING google.com (74.125.224.145) 56(84) bytes of data.
64 bytes from nuq04s09-in-f17.1e100.net (74.125.224.145): icmp_seq=1 ttl=54 time=6.87 ms
64 bytes from nuq04s09-in-f17.1e100.net (74.125.224.145): icmp_seq=2 ttl=54 time=6.62 ms
64 bytes from nuq04s09-in-f17.1e100.net (74.125.224.145): icmp_seq=3 ttl=54 time=6.64 ms
64 bytes from nuq04s09-in-f17.1e100.net (74.125.224.145): icmp_seq=4 ttl=54 time=6.59 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3012ms
rtt min/avg/max/mdev = 6.593/6.684/6.871/0.136 ms
```

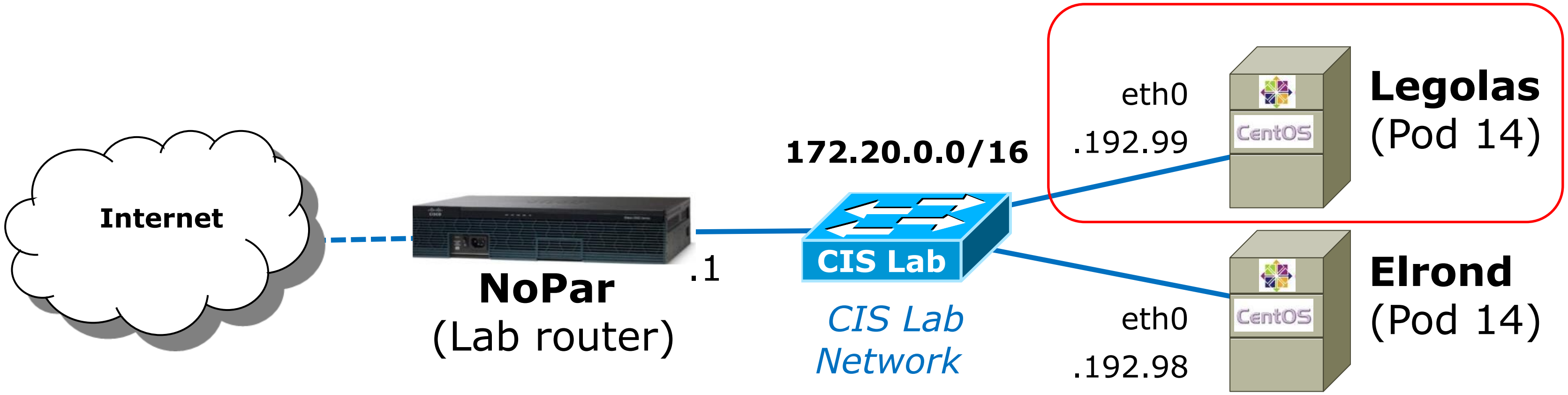
Success ping of Internet host by name!



Step 3 - Internet host by name trouble

Check name resolution (DNS settings) by ping a node on the Internet by name.

Ping test	Cabling	IP	Subnet mask	Default Gateway	DNS name servers	Ping results
ping google.com	correct	correct	correct	correct	correct	Success
ping google.com	correct	correct	correct	correct	none specified	**ping: unknown host google.com**



Commands for your toolbox

ping *xxx.xxx.xxx.xxx*	*Continuous pings (Ctrl-C to stop)*
ping *hostname*	*Continuous pings (Ctrl-C to stop)*
ping -c *n xxx.xxx.xxx.xxx*	*To ping n times only*
ping -R *xxx.xxx.xxx.xxx*	*To see route information*
ping -I eth*n xxx.xxx.xxx.xxx*	*To specify source interface ethn*
ping -b *xxx.xxx.xxx*.255	*For broadcast pings*

echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts (on other nodes being pinged)

traceroute *xxx.xxx.xxx.xxx*	*To see route information*
mtr *xxx.xxx.xxx.xxx*	To see route information

Using ping command with R and c options

```
root@frodo:~# ping -R -c 1 opus.cabrillo.edu
PING opus.cabrillo.edu (207.62.186.9) 56(124) bytes of data.
64 bytes from opus.cabrillo.edu (207.62.186.9): icmp_seq=1 ttl=63 time=2.73 ms
RR:     frodo.local (172.30.4.150)
        207.62.186.30
        opus.cabrillo.edu (207.62.186.9)
        opus.cabrillo.edu (207.62.186.9)
        172.30.4.1
        frodo.local (172.30.4.150)


--- opus.cabrillo.edu ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.732/2.732/2.732/0.000 ms
root@frodo:~#
```

Similar to traceroute

-R records the route used for the ping, -c sets the count of how many pings to send

traceroute command

```
[root@elrond ~]# traceroute google.com
traceroute to google.com (209.85.171.100), 30 hops max, 40 byte packets
 1  172.30.4.1 (172.30.4.1)  5.649 ms  6.507 ms  7.695 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
```

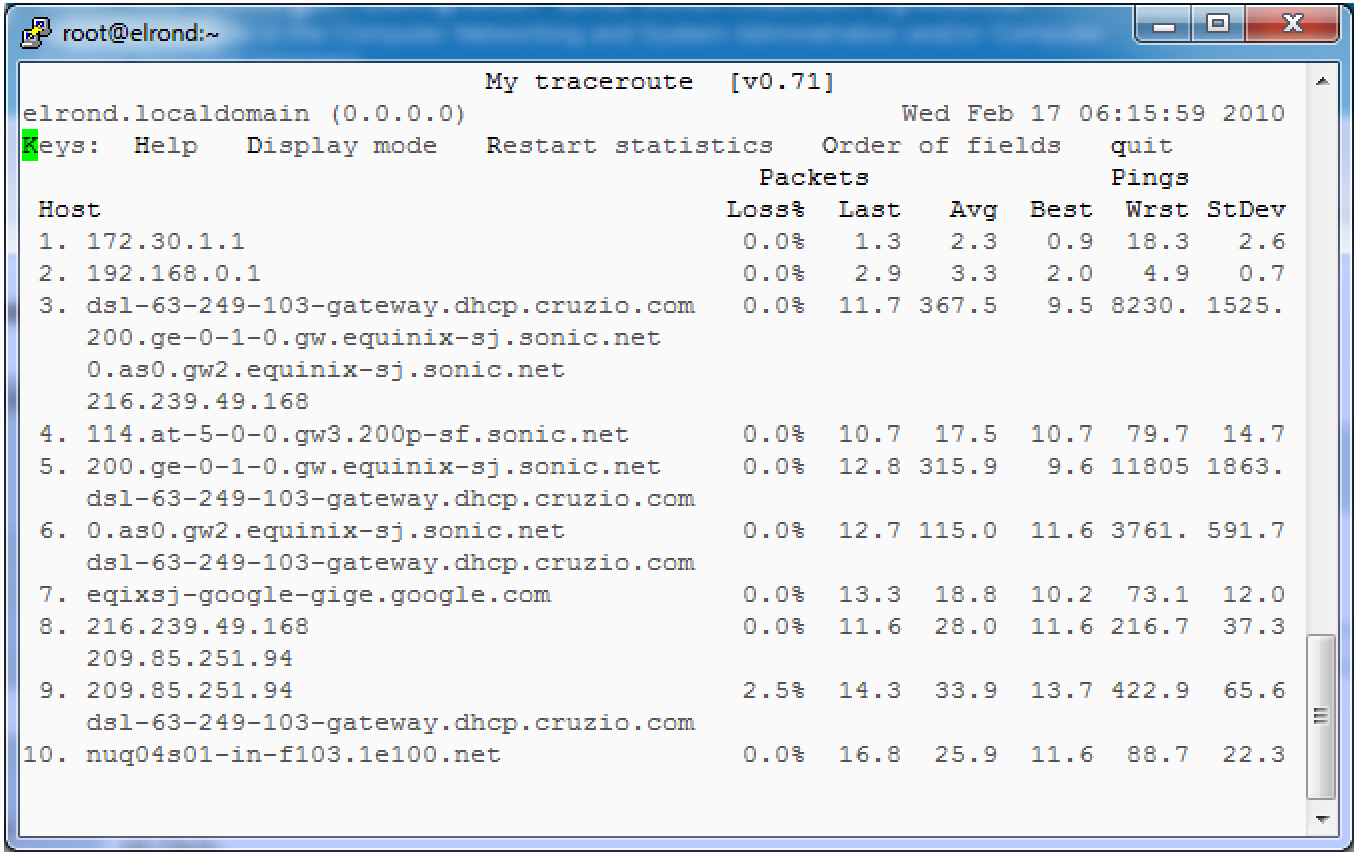
Ctrl-C to stop

Using -I option to use ICMP instead of UDP

```
[root@elrond ~]# traceroute -I google.com
traceroute to google.com (209.85.171.100), 30 hops max, 40 byte packets
 1  172.30.4.1 (172.30.4.1)  4.756 ms  6.571 ms  7.829 ms
 2  207.62.184.4 (207.62.184.4)  14.907 ms  15.631 ms  15.996 ms
 3  dc-oak-dc1--cab-cc-egm.cenic.net (137.164.34.120)  16.785 ms  17.534 ms  17.862 ms
 4  dc-oak-core1--oak-agg1-ge.cenic.net (137.164.46.55)  18.490 ms  19.003 ms  19.769 ms
 5  dc-svl-core1--oak-core1-ge-1.cenic.net (137.164.46.212)  20.769 ms  23.570 ms  26.460 ms
 6  dc-svl-peer1--svl-core1-10ge.cenic.net (137.164.46.205)  27.112 ms  10.025 ms  10.635 ms
 7  te4-4--482.tr01-plalca01.transitrail.net (137.164.131.237)  10.969 ms  9.992 ms  10.718 ms
 8   (137.164.130.94)  10.735 ms  10.675 ms  11.063 ms
 9  209.85.240.114 (209.85.240.114)  11.610 ms  10.864 ms  11.106 ms
10  216.239.49.198 (216.239.49.198)  24.040 ms  21.596 ms  21.487 ms
11  216.239.48.34 (216.239.48.34)  23.582 ms  25.061 ms  25.734 ms
12  64.233.174.101 (64.233.174.101)  20.129 ms 64.233.174.125 (64.233.174.125)  19.820 ms  19.706 ms
13  209.85.251.137 (209.85.251.137)  22.856 ms 209.85.251.129 (209.85.251.129)  33.682 ms 209.85.251.149
    (209.85.251.149)  29.731 ms
14  74.125.31.6 (74.125.31.6)  23.278 ms 74.125.31.134 (74.125.31.134)  20.824 ms 74.125.31.6 (74.125.31.6)
    21.776 ms
15  cg-in-f100.google.com (209.85.171.100)  20.158 ms  19.939 ms  19.710 ms
[root@elrond ~]#
```

mtr command

[root@elrond ~]# **mtr google.com**



```
                             My traceroute  [v0.71]
elrond.localdomain (0.0.0.0)                           Wed Feb 17 06:15:59 2010
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                              Packets               Pings
 Host                                       Loss%  Last   Avg  Best  Wrst StDev
 1. 172.30.1.1                               0.0%   1.3   2.3   0.9  18.3   2.6
 2. 192.168.0.1                              0.0%   2.9   3.3   2.0   4.9   0.7
 3. dsl-63-249-103-gateway.dhcp.cruzio.com   0.0%  11.7 367.5   9.5 8230. 1525.
    200.ge-0-1-0.gw.equinix-sj.sonic.net
    0.as0.gw2.equinix-sj.sonic.net
    216.239.49.168
 4. 114.at-5-0-0.gw3.200p-sf.sonic.net       0.0%  10.7  17.5  10.7  79.7  14.7
 5. 200.ge-0-1-0.gw.equinix-sj.sonic.net     0.0%  12.8 315.9   9.6 11805 1863.
    dsl-63-249-103-gateway.dhcp.cruzio.com
 6. 0.as0.gw2.equinix-sj.sonic.net           0.0%  12.7 115.0  11.6 3761. 591.7
    dsl-63-249-103-gateway.dhcp.cruzio.com
 7. eqixsj-google-gige.google.com            0.0%  13.3  18.8  10.2  73.1  12.0
 8. 216.239.49.168                           0.0%  11.6  28.0  11.6 216.7  37.3
    209.85.251.94
 9. 209.85.251.94                            2.5%  14.3  33.9  13.7 422.9  65.6
    dsl-63-249-103-gateway.dhcp.cruzio.com
10. nuq04s01-in-f103.1e100.net               0.0%  16.8  25.9  11.6  88.7  22.3
```

A very nice alternative to traceroute

Class Activity

Group Troubleshooting Session

Did anyone have trouble getting their Elrond to connect with a permanent static IP address?

Q2

ipv6
link-local

Link-local IPv6 Addresses

- IPv6 is a layer 3 protocol designed to replace IPv4
- IPv6 uses 128 bits to form an IP address as opposed to 32 bits in IPv4
- Link-local IPv6 addresses are automatically assigned but only work on the local LAN
- The VMs for this course support IPv6
- Cabrillo College IPv6 Internet access coming "soon"

New commands for your toolbox

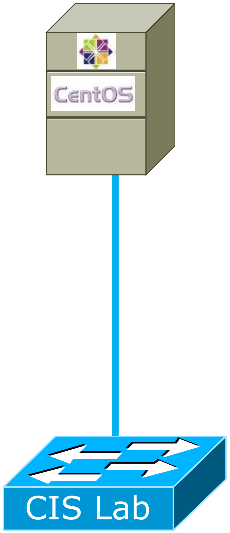
ifconfig	*Shows current IPv4 and IPV6 addresses*
ping 127.0.0.1	*Pings "yourself" using IPv4 loopback address*
ping6 ::1	*Pings "yourself" using IPv6 loopback address*
ping6 -I eth0 ff02::1	*Multicast ping (local link only)*

ssh *username*@**fe80::***xxx*:*xxxx*:*xxxx*:*xxxx***%eth0**

SSH using link-local IPv6 address

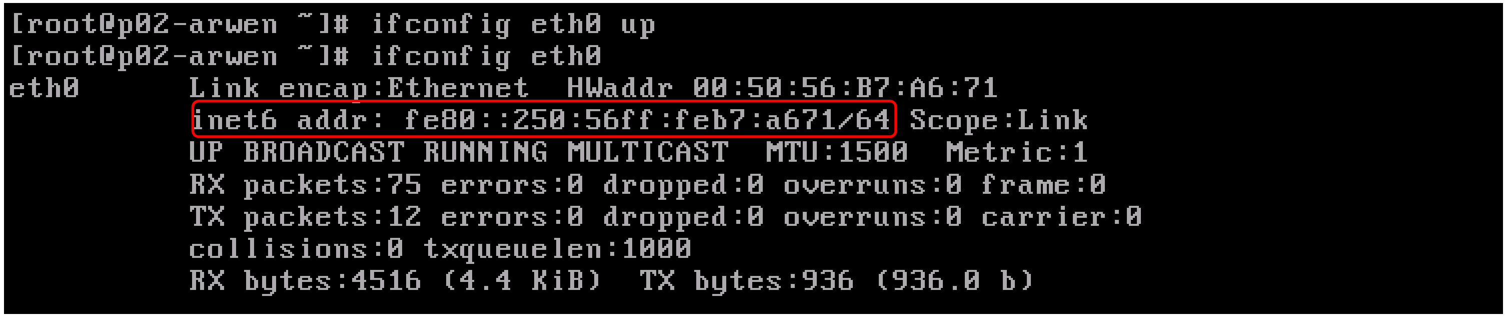
Showing IPv6 addresses

p02-Arwen



ifconfig eth0 up
ifconfig eth0

Arwen is cabled to the CIS Lab network. The eth0 interface is brought up with no IPv4 address.



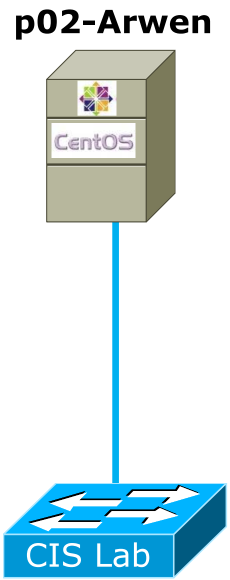
```
[root@p02-arwen ~]# ifconfig eth0 up
[root@p02-arwen ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:B7:A6:71
          inet6 addr: fe80::250:56ff:feb7:a671/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:75 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4516 (4.4 KiB)  TX bytes:936 (936.0 b)
```

The link-local IPv6 address is set to fe80::250:56ff:feb7:a671

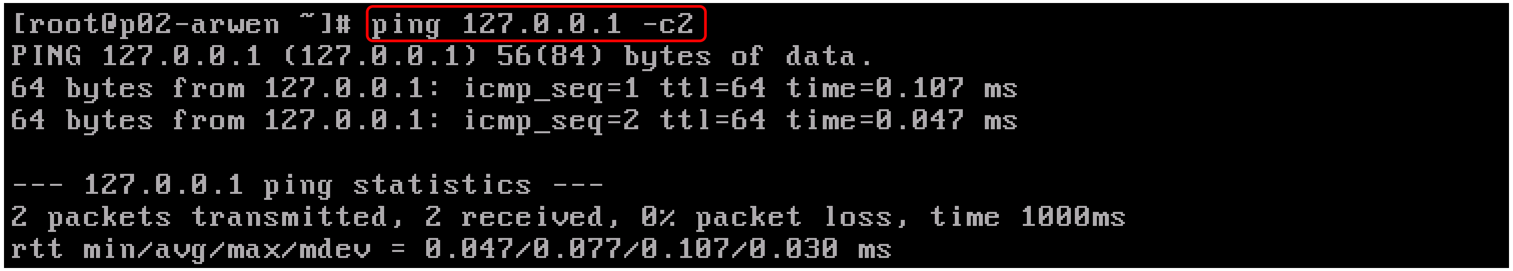
HWaddr 00:50:56:B7:A6:71

inet6 addr: fe80::250:56ff:feb7:a671/64

Ping IPv4 and IPv6 loopback addresses

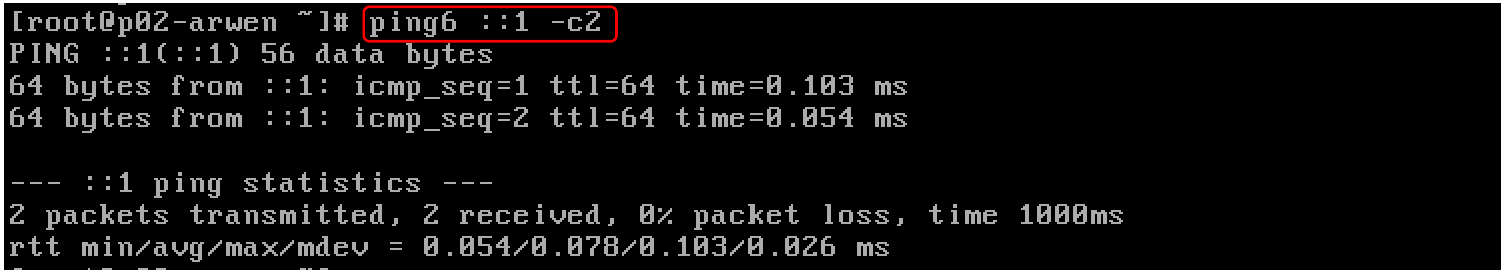


ping 127.0.0.1 *IPv4 loopback address*



```
[root@p02-arwen ~]# ping 127.0.0.1 -c2
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.107 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.047 ms

--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.047/0.077/0.107/0.030 ms
```

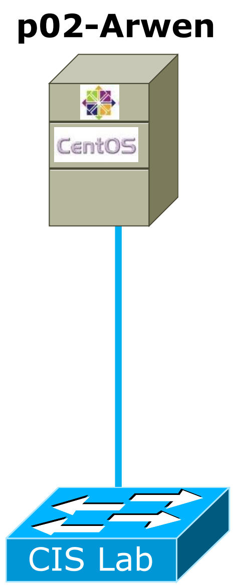
ping6 ::1 *IPv6 loopback address*



```
[root@p02-arwen ~]# ping6 ::1 -c2
PING ::1(::1) 56 data bytes
64 bytes from ::1: icmp_seq=1 ttl=64 time=0.103 ms
64 bytes from ::1: icmp_seq=2 ttl=64 time=0.054 ms

--- ::1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.054/0.078/0.103/0.026 ms
```

Arwen can ping its own IPv4 or IPv6 loopback address

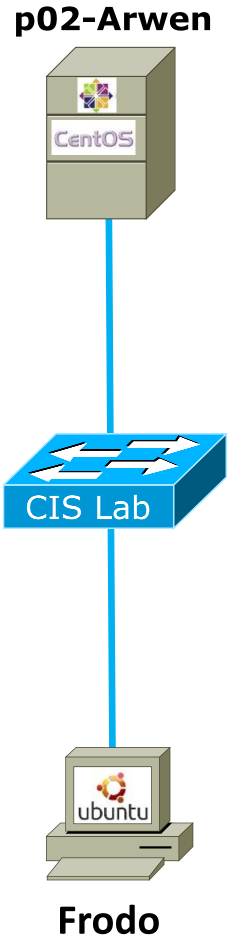
Multicast IPv6 ping



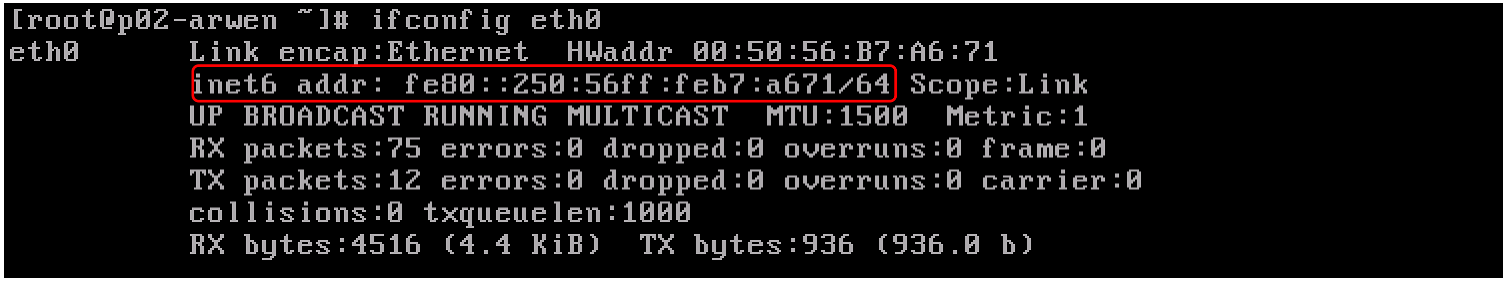
```
[cis192@p02-arwen ~]$ ping6 -I eth0 ff02::1 -c2
PING ff02::1(ff02::1) from fe80::250:56ff:feb7:a671 eth0: 56 data bytes
64 bytes from fe80::250:56ff:feb7:a671: icmp_seq=1 ttl=64 time=0.078 ms
64 bytes from fe80::250:56ff:feb7:7f83: icmp_seq=1 ttl=64 time=0.303 ms (DUP!)
64 bytes from fe80::250:56ff:feb7:7b27: icmp_seq=1 ttl=64 time=0.327 ms (DUP!)
64 bytes from fe80::250:56ff:febd:81fe: icmp_seq=1 ttl=255 time=0.334 ms (DUP!)
64 bytes from fe80::250:56ff:febd:994e: icmp_seq=1 ttl=255 time=0.340 ms (DUP!)
64 bytes from fe80::250:56ff:febd:783c: icmp_seq=1 ttl=255 time=0.390 ms (DUP!)
64 bytes from fe80::250:56ff:febd:7c05: icmp_seq=1 ttl=255 time=0.402 ms (DUP!)
64 bytes from fe80::250:56ff:febd:227: icmp_seq=1 ttl=255 time=0.409 ms (DUP!)
64 bytes from fe80::250:56ff:febd:eb33: icmp_seq=1 ttl=255 time=0.434 ms (DUP!)
64 bytes from fe80::250:56ff:febd:cb20: icmp_seq=1 ttl=255 time=0.442 ms (DUP!)
64 bytes from fe80::250:56ff:feb7:c7df: icmp_seq=1 ttl=255 time=0.462 ms (DUP!)
64 bytes from fe80::250:56ff:feb7:eaf9: icmp_seq=1 ttl=64 time=0.487 ms (DUP!)
64 bytes from fe80::211:43ff:fecd:21d6: icmp_seq=1 ttl=64 time=0.532 ms (DUP!)
64 bytes from fe80::250:56ff:febd:386a: icmp_seq=1 ttl=255 time=0.543 ms (DUP!)
64 bytes from fe80::250:56ff:feb7:1b23: icmp_seq=1 ttl=64 time=0.550 ms (DUP!)
64 bytes from fe80::250:56ff:feb7:ac2a: icmp_seq=1 ttl=64 time=0.555 ms (DUP!)
64 bytes from fe80::250:56ff:feb7:2e29: icmp_seq=1 ttl=64 time=0.585 ms (DUP!)
64 bytes from fe80::250:56ff:feb7:2ecd: icmp_seq=1 ttl=64 time=0.594 ms (DUP!)
64 bytes from fe80::250:56ff:feb7:bbcc: icmp_seq=1 ttl=64 time=0.615 ms (DUP!)
64 bytes from fe80::250:56ff:feb7:c951: icmp_seq=1 ttl=255 time=0.623 ms (DUP!)
64 bytes from fe80::250:56ff:feb7:355a: icmp_seq=1 ttl=255 time=0.786 ms (DUP!)
64 bytes from fe80::250:56ff:feb7:6a96: icmp_seq=1 ttl=64 time=0.825 ms (DUP!)
64 bytes from fe80::250:56ff:feb7:9bfb: icmp_seq=1 ttl=255 time=0.832 ms (DUP!)
64 bytes from fe80::250:56ff:febd:b9bd: icmp_seq=1 ttl=255 time=0.858 ms (DUP!)
64 bytes from fe80::250:56ff:feb7:d329: icmp_seq=1 ttl=64 time=0.881 ms (DUP!)
64 bytes from fe80::250:56ff:feb7:19d5: icmp_seq=1 ttl=64 time=0.950 ms (DUP!)
64 bytes from fe80::250:56ff:febd:69bd: icmp_seq=1 ttl=255 time=1.51 ms (DUP!)
64 bytes from fe80::250:56ff:feb7:a671: icmp_seq=2 ttl=64 time=0.085 ms

--- ff02::1 ping statistics ---
2 packets transmitted, 2 received, +26 duplicates, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.078/0.562/1.516/0.284 ms
[cis192@p02-arwen ~]$
```

Quick way to discover other IPv6 interfaces on the local netwrok

Ping link-local IPv6 addresses



ifconfig eth0 *show link-local IPv6 address*

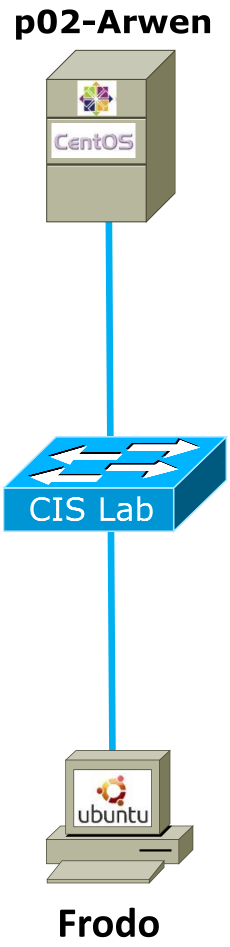


```
[root@p02-arwen ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:B7:A6:71
          inet6 addr: fe80::250:56ff:feb7:a671/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:75 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4516 (4.4 KiB)  TX bytes:936 (936.0 b)
```

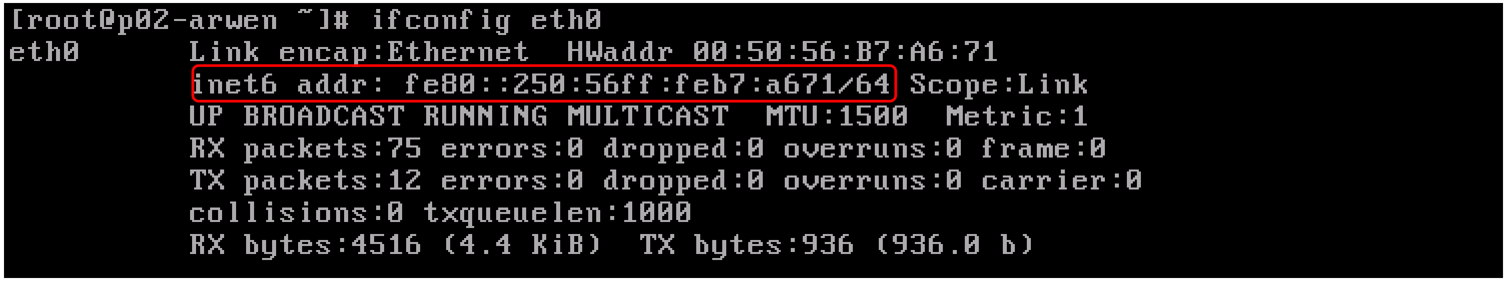
Frodo pings Awen's link-local IPv6 address

```
cis192@p02-frodo:~$ ping6 -I eth0 fe80::250:56ff:feb7:a671 -c2
PING fe80::250:56ff:feb7:a671(fe80::250:56ff:feb7:a671) from
fe80::250:56ff:feb7:1b23 eth0: 56 data bytes
64 bytes from fe80::250:56ff:feb7:a671: icmp_seq=1 ttl=64 time=0.464 ms
64 bytes from fe80::250:56ff:feb7:a671: icmp_seq=2 ttl=64 time=0.297 ms

--- fe80::250:56ff:feb7:a671 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.297/0.380/0.464/0.085 ms
cis192@p02-frodo:~$
```

SSH using link-local IPv6 addresses



ifconfig eth0 *show link-local IPv6 address*



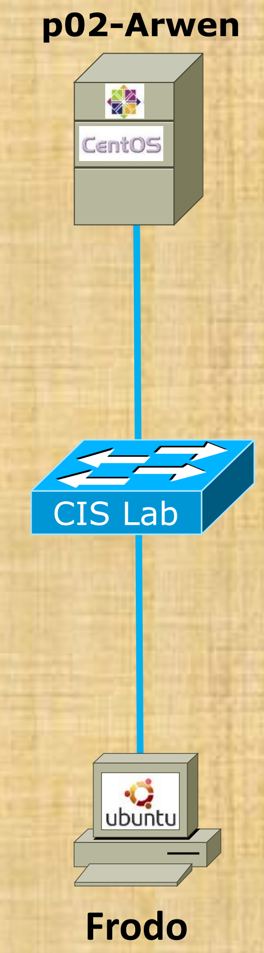
```
[root@p02-arwen ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:B7:A6:71
          inet6 addr: fe80::250:56ff:feb7:a671/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:75 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4516 (4.4 KiB)  TX bytes:936 (936.0 b)
```

From Frodo we ssh into Arwen using its link-local IPv6 address

```
cis192@p02-frodo:~$ ssh cis192@fe80::250:56ff:feb7:a671%eth0
The authenticity of host 'fe80::250:56ff:feb7:a671%eth0
(fe80::250:56ff:feb7:a671%eth0)' can't be established.
RSA key fingerprint is 81:46:a3:17:7a:4b:91:c9:24:96:f3:ac:05:5a:c4:29.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'fe80::250:56ff:feb7:a671%eth0' (RSA) to the list of
known hosts.
cis192@fe80::250:56ff:feb7:a671%eth0's password:
Last login: Tue Feb 12 20:45:14 2013
[cis192@p02-arwen ~]$
```

Class Activity

IPv6



Your turn!

Prepare your Frodo
- Make sure it is cabled to the CIS Lab network
- Power it on

ping p2_arwen using IPv6
- **ping6 -I eth0 fe80::250:56ff:feb7:a671 -c2**

ssh into p2_arwen using IPv6
- **ssh cis192@fe80::250:56ff:feb7:a671%eth0**

ifconfig and aliases

Alias IP Addresses

What is it
- It lets you assign more than one IP address to an interface

Why?
- It give you additional flexibility for customizing access to different groups of users for different services

It is possible to have more than one IP address on an interface using aliases. This is different than multi-homing which is having multiple interfaces on a computer.

New commands for your toolbox

Set

- To set an alias IP address and subnet mask:

```
    ifconfig ethn:m xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx
```

- To set an alias IP address using the prefix instead:

```
    ifconfig ethn:m xxx.xxx.xxx.xxx/pp
```

Verify

- To show all interfaces (and to show your IP address):

```
    ifconfig
```

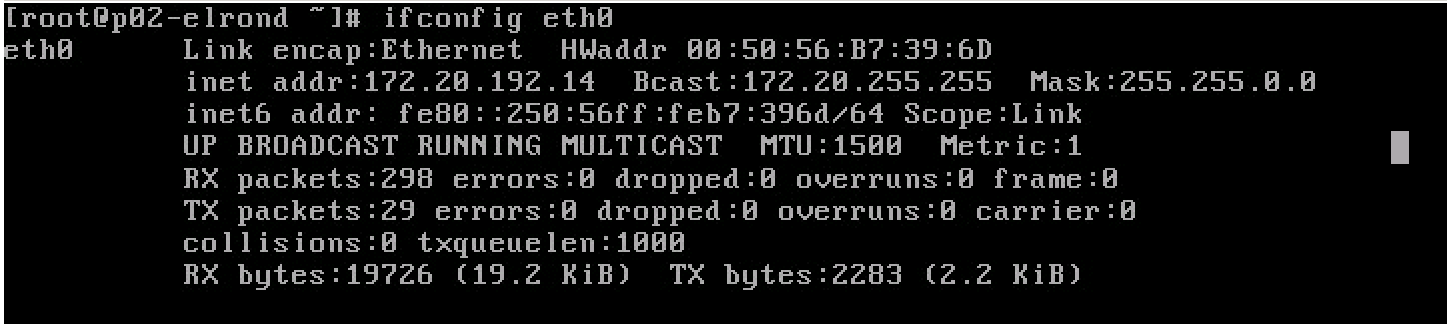
- To show a single alias interface:

```
    ifconfig ethn:m
```

n=interface number, *m*=arbitrary number to distinguish the alias

Create an Alias IP Address

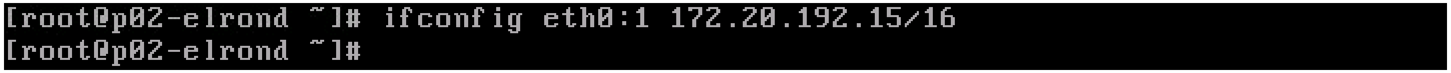
ifconfig eth0



```
[root@p02-elrond ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:B7:39:6D
          inet addr:172.20.192.14  Bcast:172.20.255.255  Mask:255.255.0.0
          inet6 addr: fe80::250:56ff:feb7:396d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:298 errors:0 dropped:0 overruns:0 frame:0
          TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:19726 (19.2 KiB)  TX bytes:2283 (2.2 KiB)
```

Elrond still has the IP address we set earlier on eth0

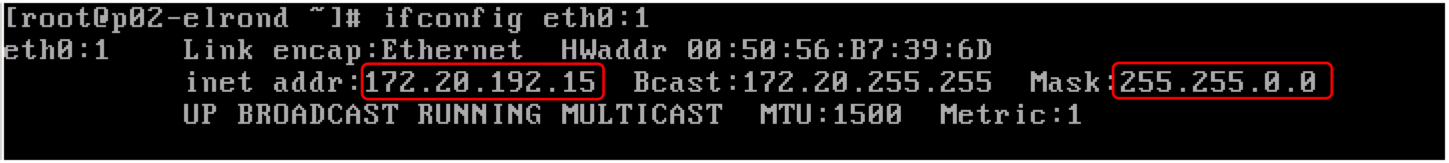
ifconfig eth0:1 172.20.192.15/16



```
[root@p02-elrond ~]# ifconfig eth0:1 172.20.192.15/16
[root@p02-elrond ~]#
```

Lets add a second IP address to the same interface

ifconfig eth0:1

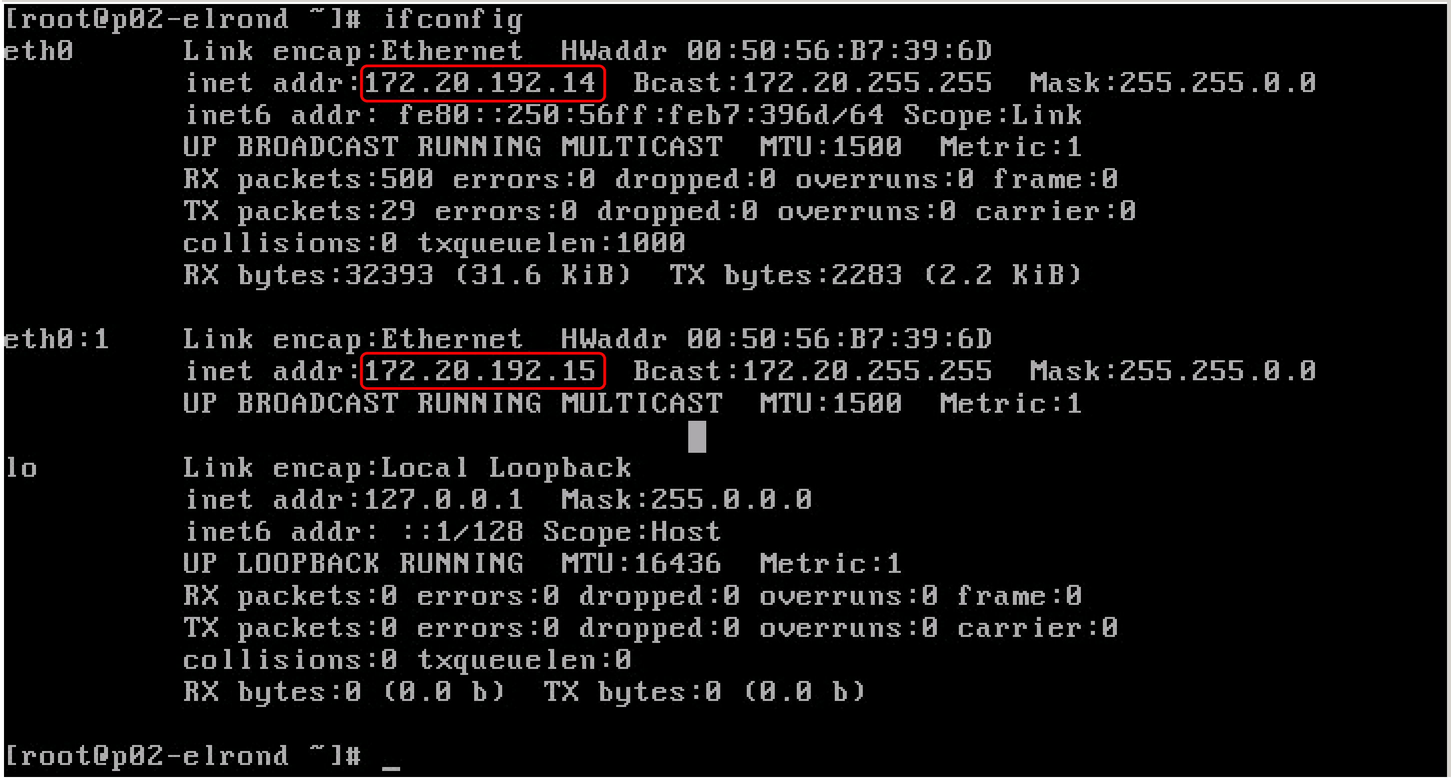


```
[root@p02-elrond ~]# ifconfig eth0:1
eth0:1    Link encap:Ethernet  HWaddr 00:50:56:B7:39:6D
          inet addr:172.20.192.15  Bcast:172.20.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

Verify the second IP address

Verify an Alias IP Address

ifconfig



```
[root@p02-elrond ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:B7:39:6D
          inet addr:172.20.192.14  Bcast:172.20.255.255  Mask:255.255.0.0
          inet6 addr: fe80::250:56ff:feb7:396d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:500 errors:0 dropped:0 overruns:0 frame:0
          TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:32393 (31.6 KiB)  TX bytes:2283 (2.2 KiB)

eth0:1    Link encap:Ethernet  HWaddr 00:50:56:B7:39:6D
          inet addr:172.20.192.15  Bcast:172.20.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

[root@p02-elrond ~]# _
```

Verify all addresses on all interfaces

Test an Alias IP Address

```
cis192@p02-frodo:~$ ping 172.20.192.14 -c2
PING 172.20.192.14 (172.20.192.14) 56(84) bytes of data.
64 bytes from 172.20.192.14: icmp_req=1 ttl=64 time=0.773 ms
64 bytes from 172.20.192.14: icmp_req=2 ttl=64 time=0.296 ms

--- 172.20.192.14 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.296/0.534/0.773/0.239 ms
```

Frodo can ping Elrond's first IP address

```
cis192@p02-frodo:~$ ping 172.20.192.15 -c2
PING 172.20.192.15 (172.20.192.15) 56(84) bytes of data.
64 bytes from 172.20.192.15: icmp_req=1 ttl=64 time=0.967 ms
64 bytes from 172.20.192.15: icmp_req=2 ttl=64 time=0.245 ms

--- 172.20.192.15 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.245/0.606/0.967/0.361 ms
cis192@p02-frodo:~$
```

Frodo can ping Elrond's second IP address

Verify you can ping both addresses from another host

Create a permanent alias IP Address

Create this file on Elrond
(make a copy of ifcfg-eth0 and modify it)

/etc/sysconfig/network-scripts/ifcfg-eth0:1

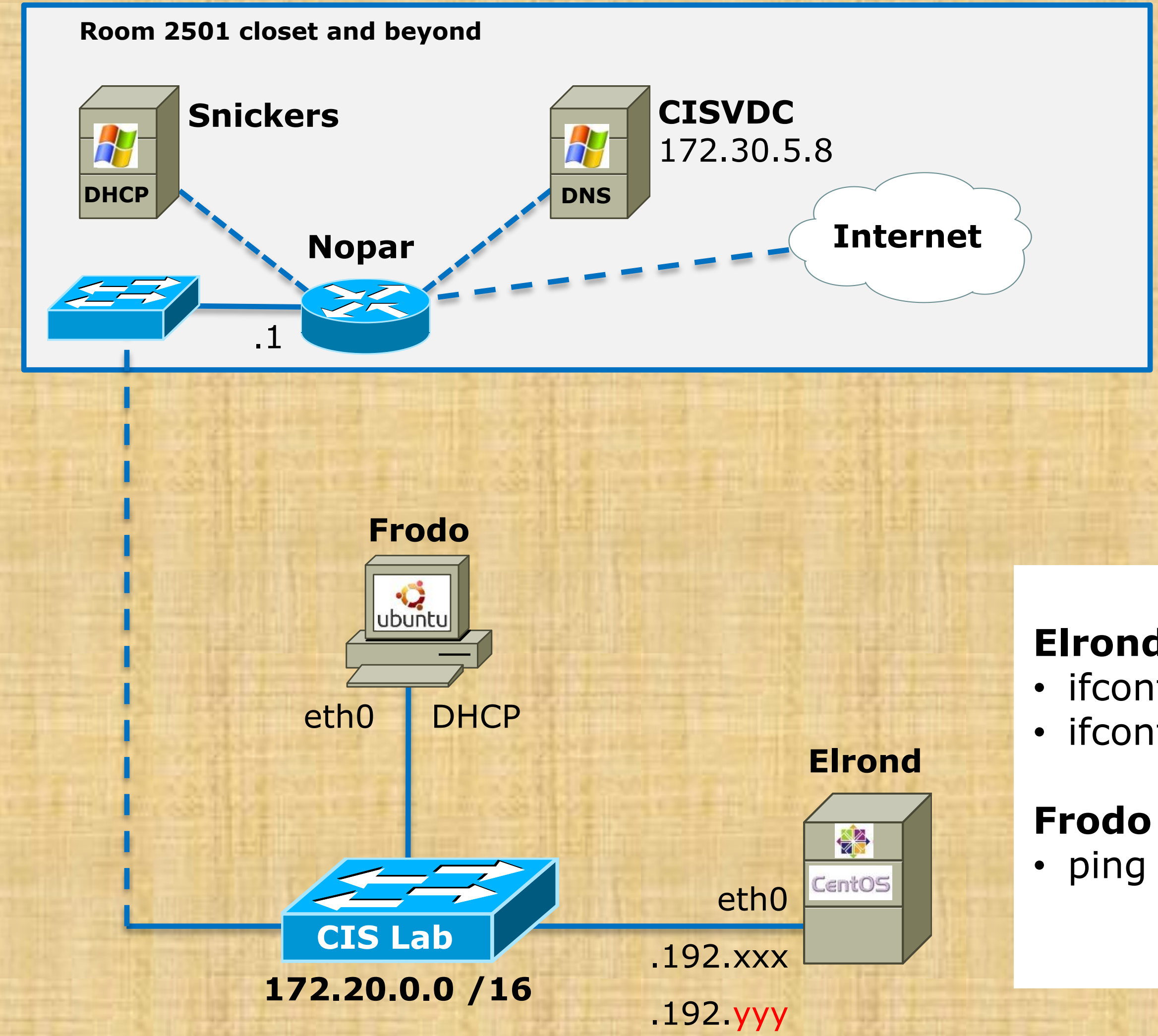
```
DEVICE="eth0:1"
NM_CONTROLLED="no"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=172.20.192.15
NETMASK=255.255.0.0
```

then run **service network restart**

This would create the alias IP address permanently on Elrond

Class Activity

Give Elrond a temporary alias IP address



VLab RDP file

CIS 90 VLab VM Assignements

CIS 192 VLab Pod Assignements

No DUPS Please!

Elrond
- ifconfig eth0:1 172.20.192.yyyy/16
- ifconfig

Frodo
- ping both of Elrond’s IP addresses

ARP

ARP - Address Resolution Protocol

Overview

The purpose of ARP is to provide the correct destination physical address given the destination IP address.

- RFC 826 (http://tools.ietf.org/html/rfc826)
- Part of IPv4 (IPv6 uses NDP, neighbor discovery protocol)
- The ARP request:
  "Who has this IP address?"
  (broadcast to all)
- The ARP reply:
  "I do and my MAC address is xx:xx:xx:xx:xx:xx"
  (unicast back to requester)

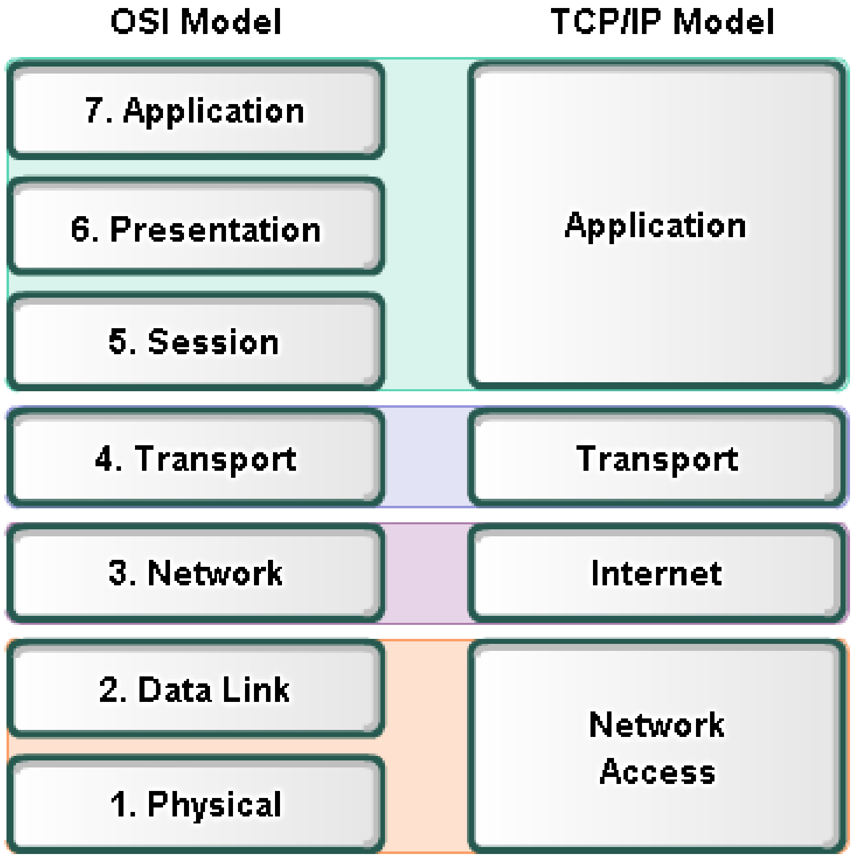
TCP/IP and ARP

The TCP/IP Suite of Protocols	
Application	File Transfer: FTP, TFTP, NFS, HTTP Email: SMTP Remote Login: Telnet, rlogin Network Management: SNMP, BootP Name Management: DNS, DHCP
Transport	TCP, UDP
Internet/Network	IP, ICMP, IGMP, **ARP, RARP**
Network Interface (Link Layer)	Not Specified: Ethernet, 802.3, Token Ring, 802.5, FDDI, ATM,

ARP is a layer 3 protocol, one of many protocols within the TCP/IP suite of protocols.

Protocol and Reference Models

ARP is a layer 3 protocol

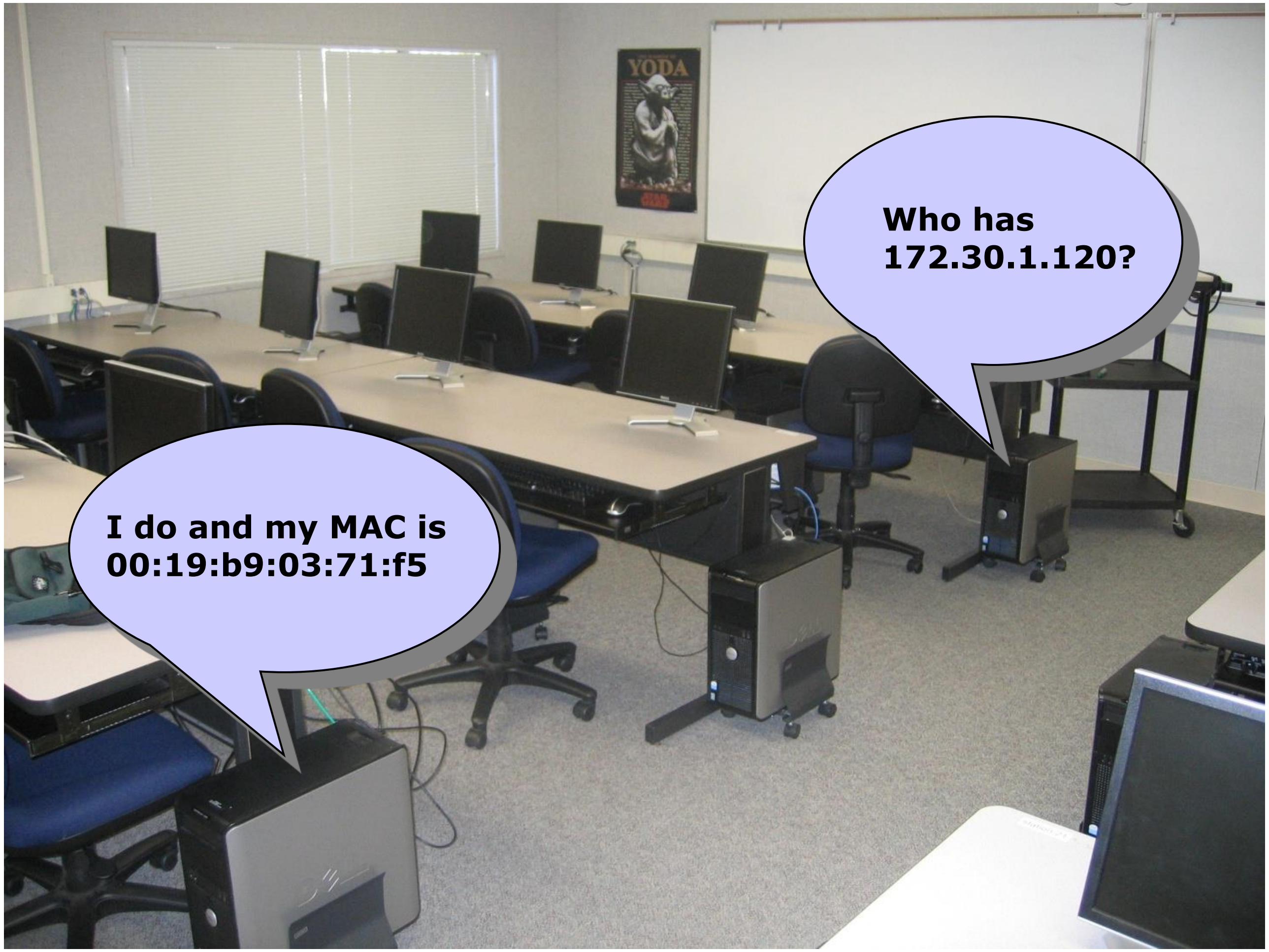


- The **Open Systems Interconnection (OSI)** model is the *most widely known internetwork reference model*.

ARP - Address Resolution Protocol

Overview Example

Station04 wants to ping Station20



ARP - Address Resolution Protocol

Overview

Devices will remember pairings of IP addresses and MAC addresses which are kept in an ARP cache table

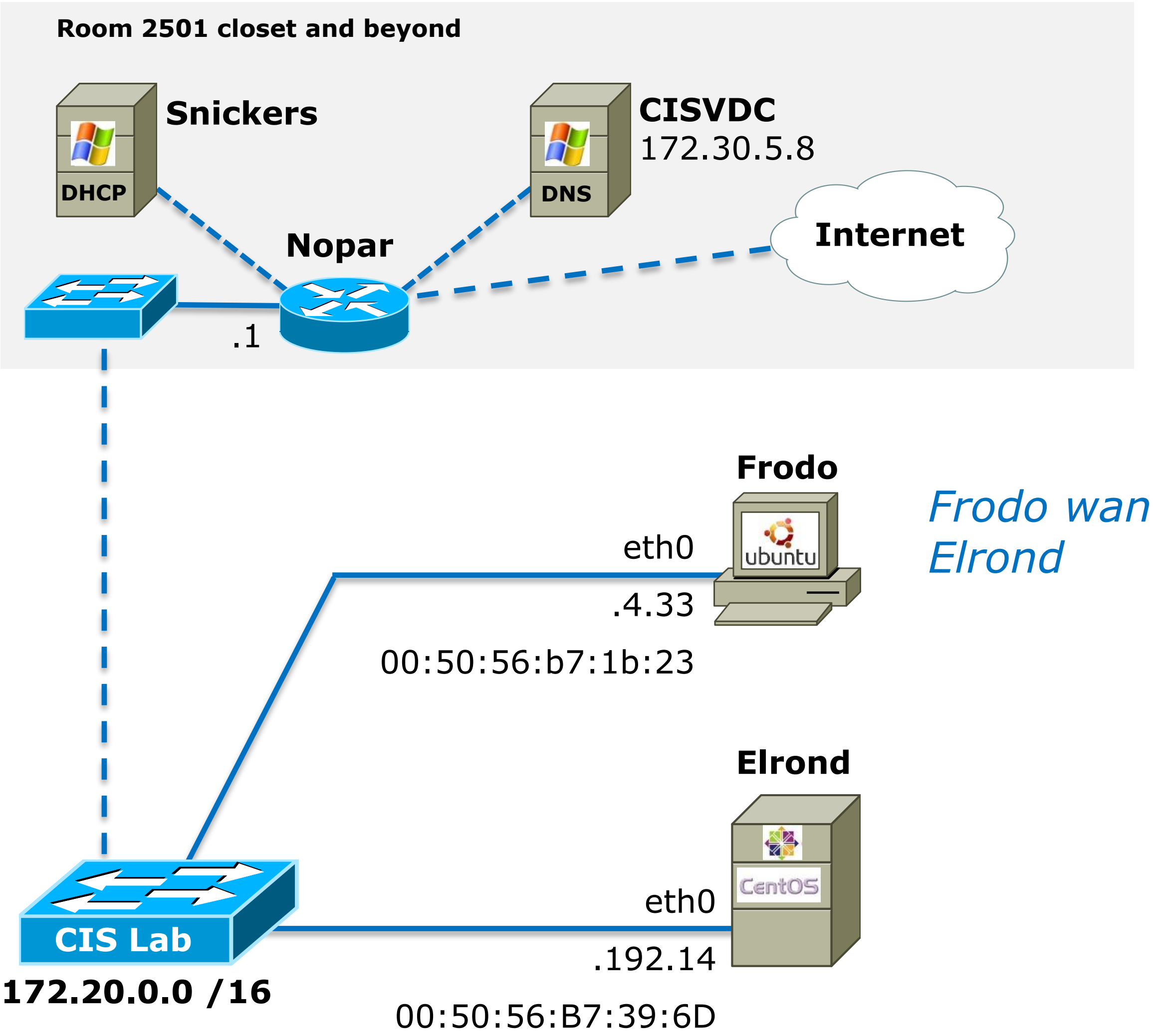
- In Linux, the **arp** command is used to show the ARP cache
- ARP cache entries will eventually timeout and be removed

ARP Poisoning

A NIC is gullible and will accept ARP replies even when not requested

- An attacker can send arp replies (even as a broadcast) to populate arp caches with bogus MAC/IP pairs
  - Denial of service: pair a non-existing MAC address with the router's IP address. External destination packets can never leave the subnet.
  - Man-in-the-middle: pair an existing hosts IP address with attackers MAC address so attacker can snoop all packets for that host.
  - MAC flooding: overload a switch so it behaves like a hub allowing a sniffer to see all traffic.

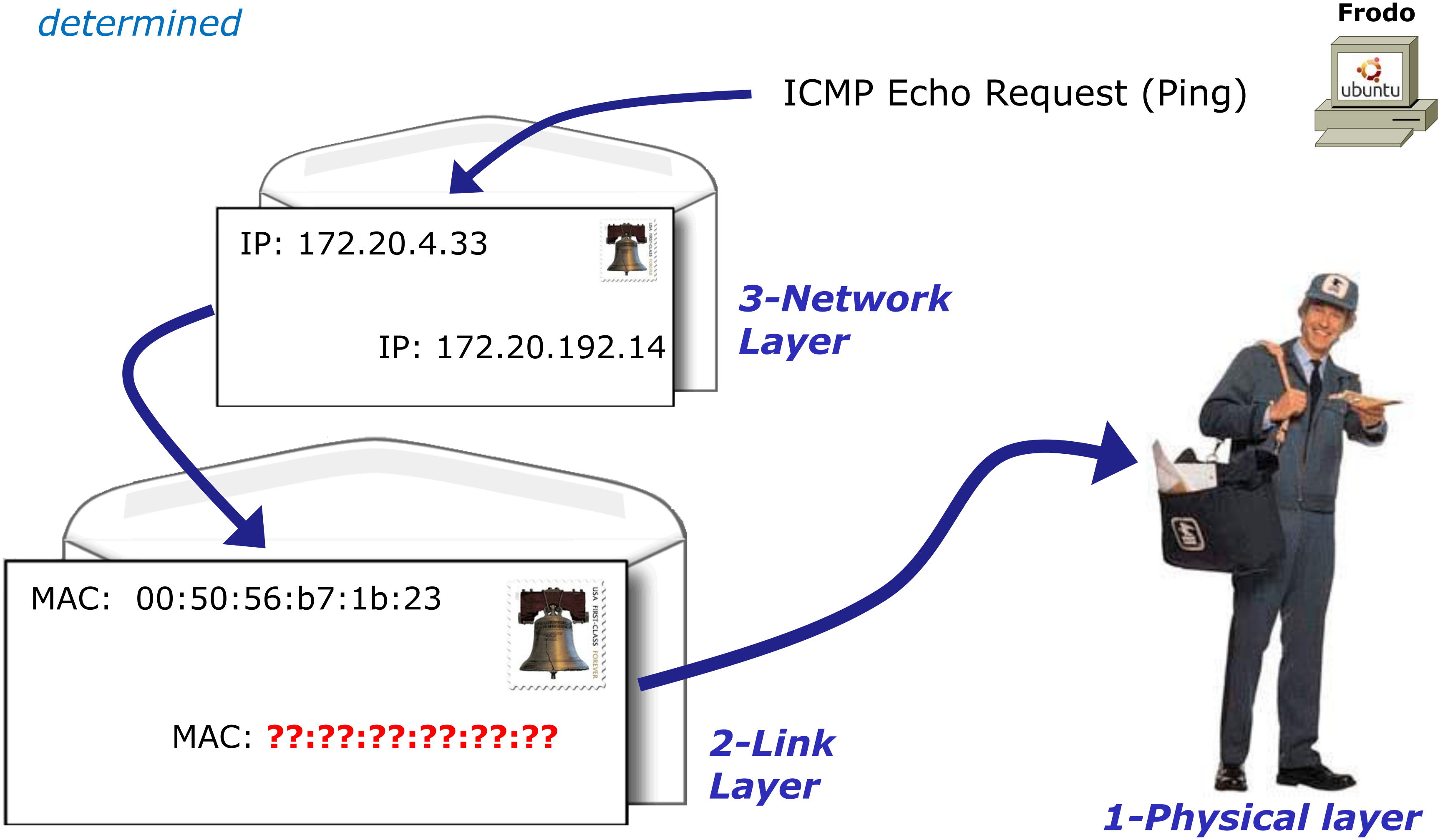
ARP Example - Frodo wants to ping Elrond



Frodo wants to ping Elrond

ARP Example - Frodo needs Elrond's MAC address

However, using encapsulation, the ping packet cannot be placed on the network until a destination MAC address for Station 09 can be determined



Frodo

ICMP Echo Request (Ping)

IP: 172.20.4.33

IP: 172.20.192.14

3-Network Layer

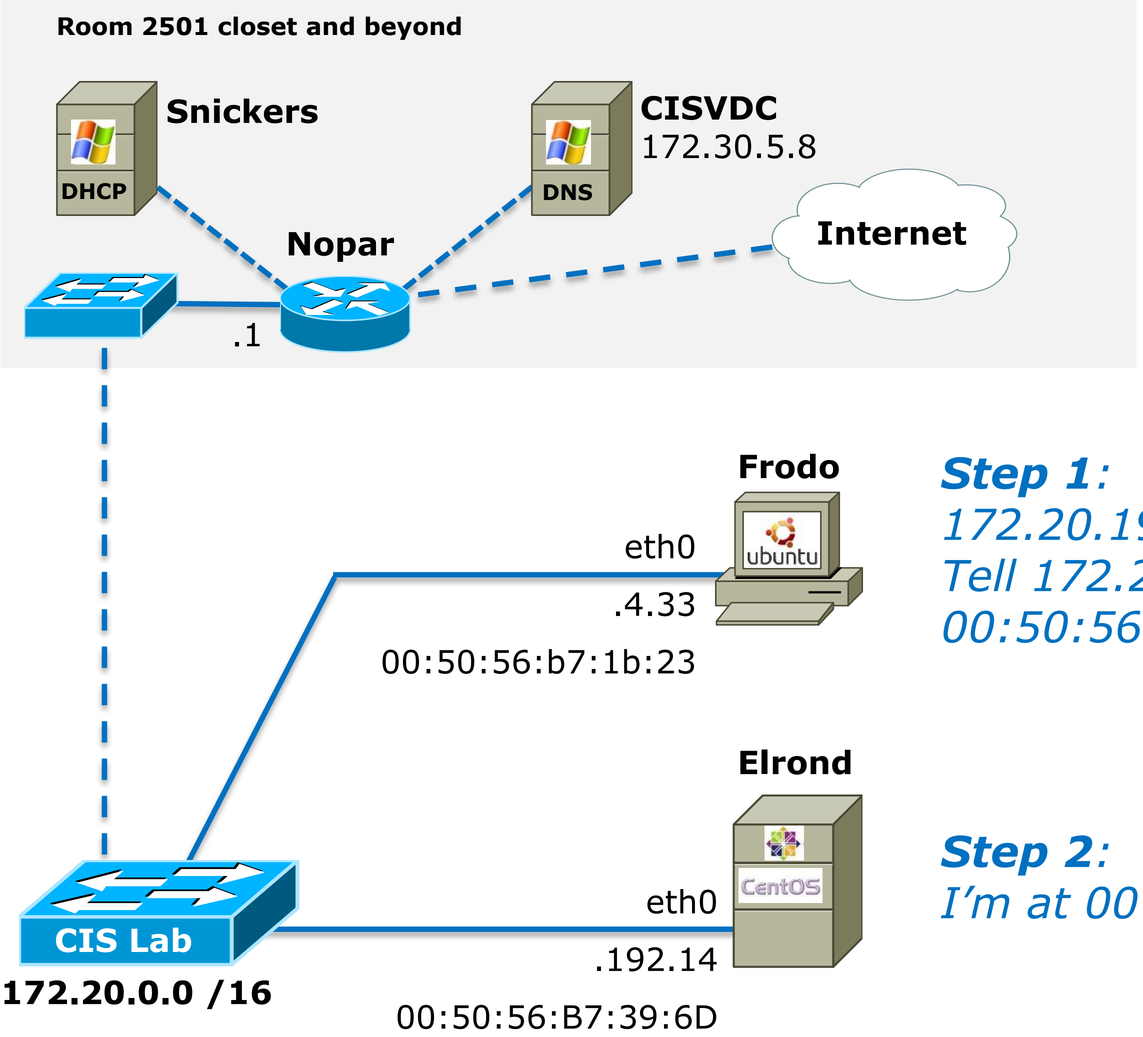
MAC: 00:50:56:b7:1b:23

MAC: **??:??:??:??:??:??**

2-Link Layer

1-Physical layer

ARP Example - Frodo requests MAC address

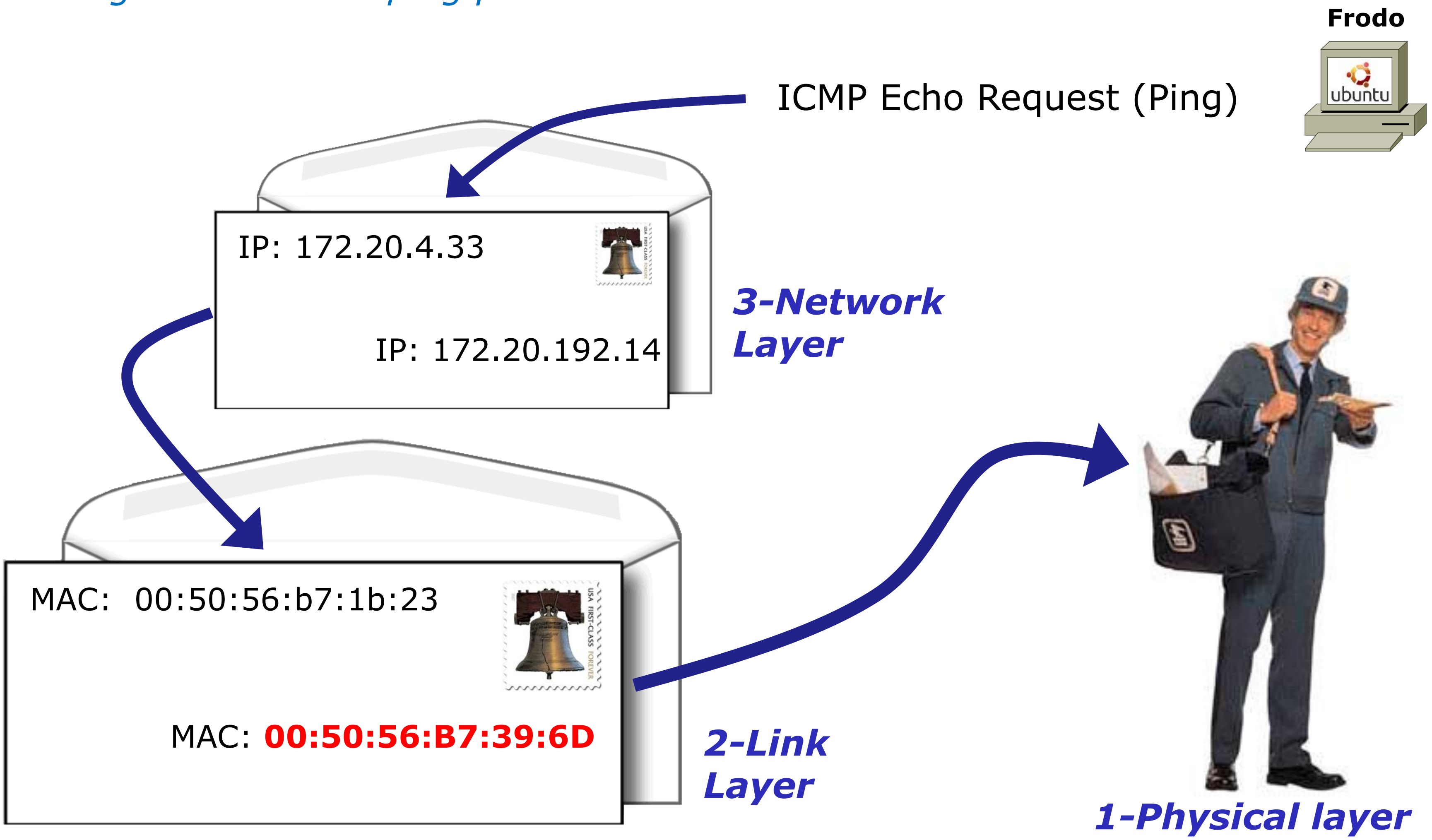


***Step 1**: Who has IP Address 172.20.192.14? (broadcast to all) Tell 172.20.4.33 at 00:50:56:b7:1b:23*

***Step 2**: I do (unicast to 172.20.4.33) I'm at 00:50:56:B7:39:6D*

ARP Example - Frodo pings Elrond

Once the destination MAC address for Elrond has been determined using ARP then the ping packet can be sent out.



Frodo

ICMP Echo Request (Ping)

IP: 172.20.4.33

IP: 172.20.192.14

3-Network Layer

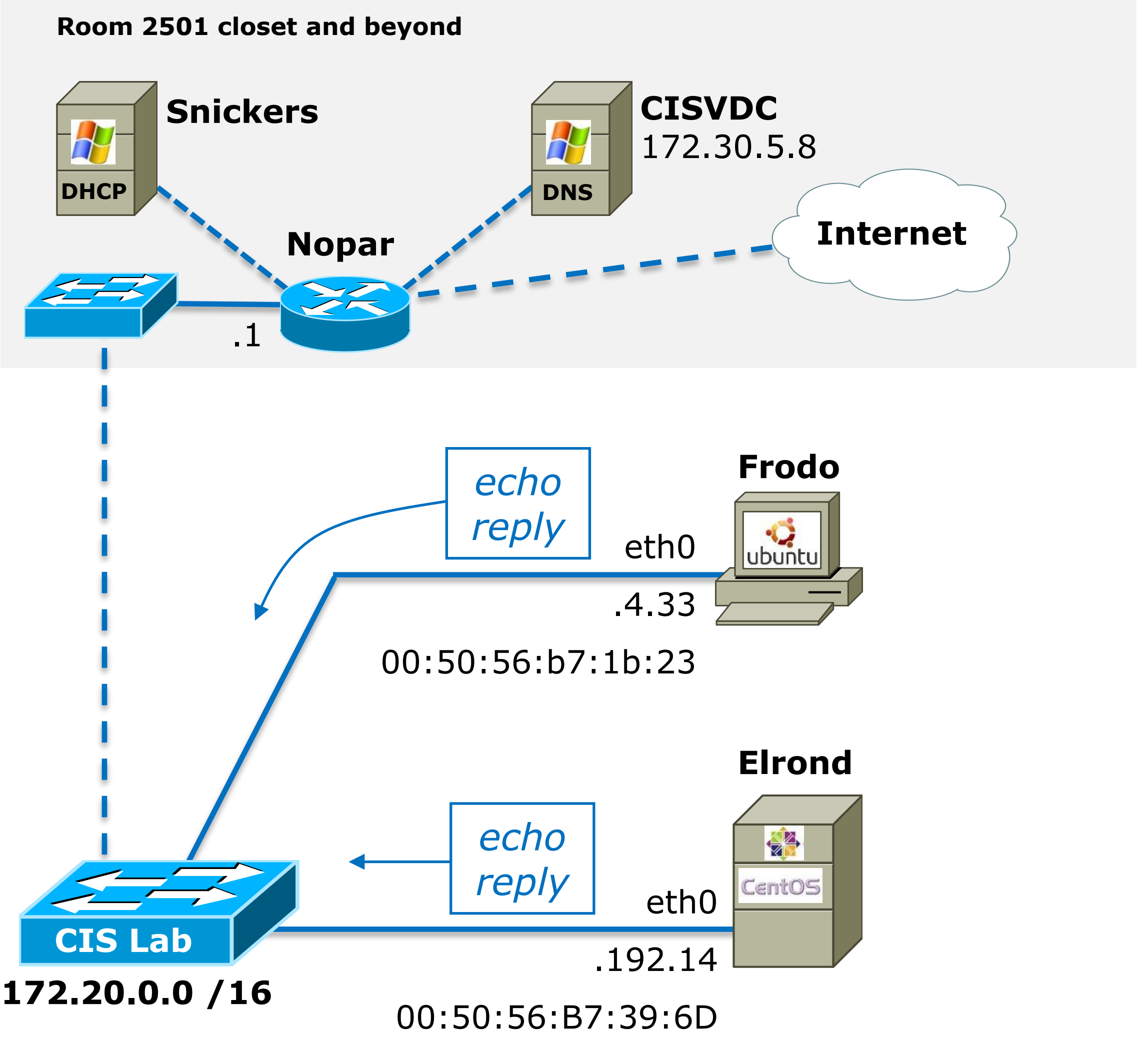
MAC: 00:50:56:b7:1b:23

MAC: **00:50:56:B7:39:6D**

2-Link Layer

1-Physical layer

ARP Example - Frodo requests MAC address



Once the destination MAC address for Elrond has been determined using ARP then the ping packet can be sent out and the reply is sent back.

ARP Example - Frodo pings Elrond

```
cis192@p02-frodo:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:b7:1b:23
          inet addr:172.20.4.33  Bcast:172.20.255.255  Mask:255.255.0.0
< snipped >
```

Frodo’s IP address is 172.20.4.33

```
cis192@p02-frodo:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.20.0.1               ether   c8:9c:1d:4f:77:01   C                     eth0
```

Frodo’s ARP cache currently only has one entry for the router

```
cis192@p02-frodo:~$ ping 172.20.192.14 -c1
PING 172.20.192.14 (172.20.192.14) 56(84) bytes of data.
64 bytes from 172.20.192.14: icmp_req=1 ttl=64 time=0.801 ms
< snipped >
```

Pinging Elrond to populate the ARP cache with Elrond's MAC address

```
cis192@p02-frodo:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.20.192.14            ether   00:50:56:b7:39:6d   C                     eth0
172.20.0.1               ether   c8:9c:1d:4f:77:01   C                     eth0
```

Elrond's MAC address is now in Frodo's ARP cache (temporarily)

ARP Example - Frodo pings Elrond

Frodo

eth0

.4.33

00:50:56:b7:1b:23

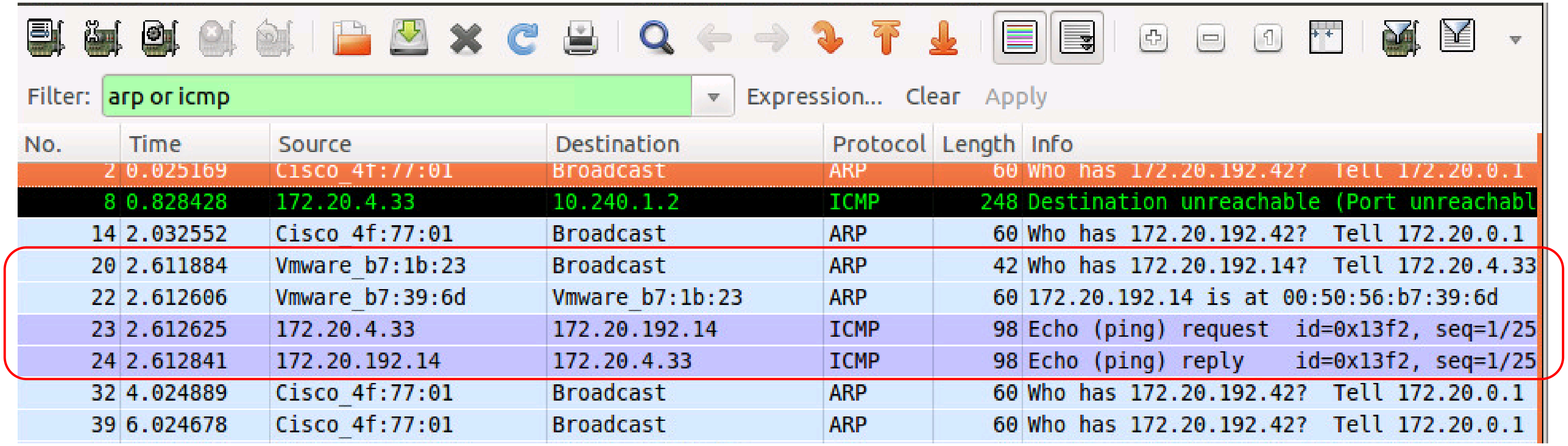
Elrond

eth0

.192.14

00:50:56:B7:39:6D

```
root@p02-frodo:~# ping 172.20.192.14 -c1
PING 172.20.192.14 (172.20.192.14) 56(84) bytes of data.
64 bytes from 172.20.192.14: icmp_req=1 ttl=64 time=0.986 ms

--- 172.20.192.14 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.986/0.986/0.986/0.000 ms
root@p02-frodo:~#
```



Filter: arp or icmp

No.	Time	Source	Destination	Protocol	Length	Info
2	0.025169	Cisco_4f:77:01	Broadcast	ARP	60	Who has 172.20.192.42? Tell 172.20.0.1
8	0.828428	172.20.4.33	10.240.1.2	ICMP	248	Destination unreachable (Port unreachabl
14	2.032552	Cisco_4f:77:01	Broadcast	ARP	60	Who has 172.20.192.42? Tell 172.20.0.1
20	2.611884	Vmware_b7:1b:23	Broadcast	ARP	42	Who has 172.20.192.14? Tell 172.20.4.33
22	2.612606	Vmware_b7:39:6d	Vmware_b7:1b:23	ARP	60	172.20.192.14 is at 00:50:56:b7:39:6d
23	2.612625	172.20.4.33	172.20.192.14	ICMP	98	Echo (ping) request id=0x13f2, seq=1/25
24	2.612841	172.20.192.14	172.20.4.33	ICMP	98	Echo (ping) reply id=0x13f2, seq=1/25
32	4.024889	Cisco_4f:77:01	Broadcast	ARP	60	Who has 172.20.192.42? Tell 172.20.0.1
39	6.024678	Cisco_4f:77:01	Broadcast	ARP	60	Who has 172.20.192.42? Tell 172.20.0.1

Running wireshark on Frodo to see ARP and ICMP packets

ARP Request Details

Frodo

eth0

.4.33

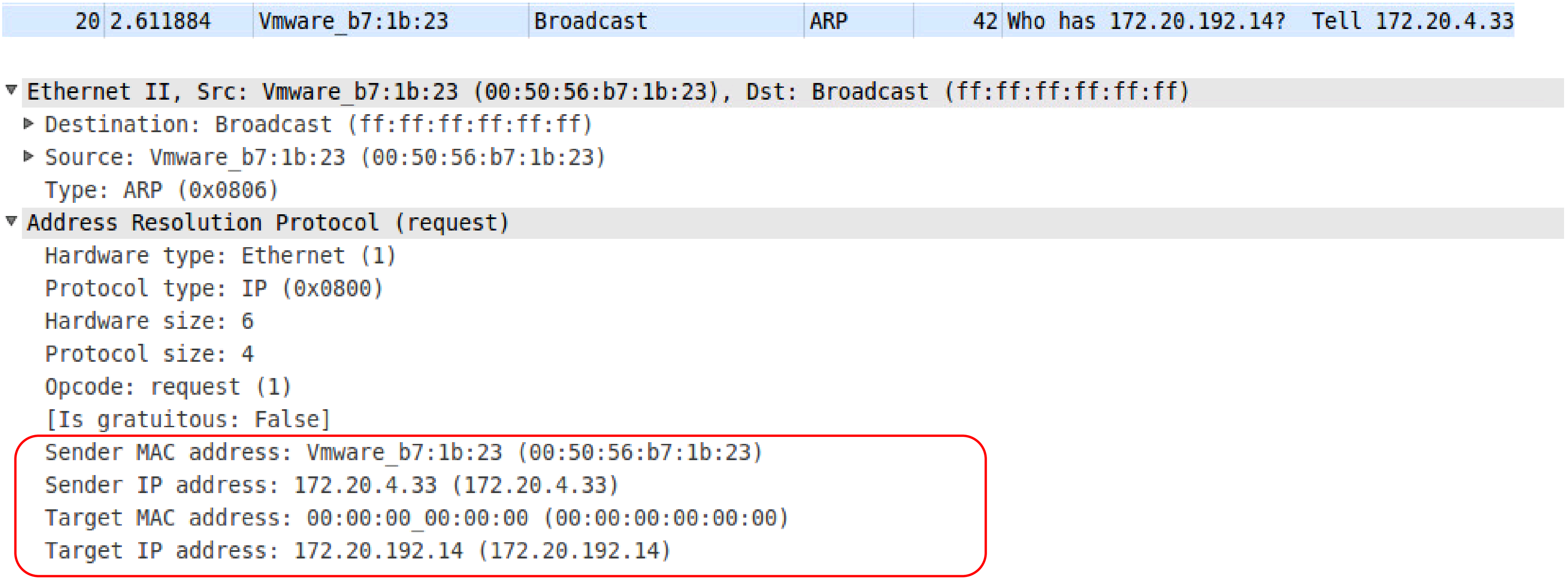
00:50:56:b7:1b:23

Elrond

eth0

.192.14

00:50:56:B7:39:6D



```
    20 2.611884    Vmware_b7:1b:23    Broadcast    ARP    42 Who has 172.20.192.14?  Tell 172.20.4.33

▼ Ethernet II, Src: Vmware_b7:1b:23 (00:50:56:b7:1b:23), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Source: Vmware_b7:1b:23 (00:50:56:b7:1b:23)
    Type: ARP (0x0806)
▼ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    [Is gratuitous: False]
    Sender MAC address: Vmware_b7:1b:23 (00:50:56:b7:1b:23)
    Sender IP address: 172.20.4.33 (172.20.4.33)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 172.20.192.14 (172.20.192.14)
```

Drill down into the ARP request packet

ARP Reply Details

Frodo

eth0

.4.33

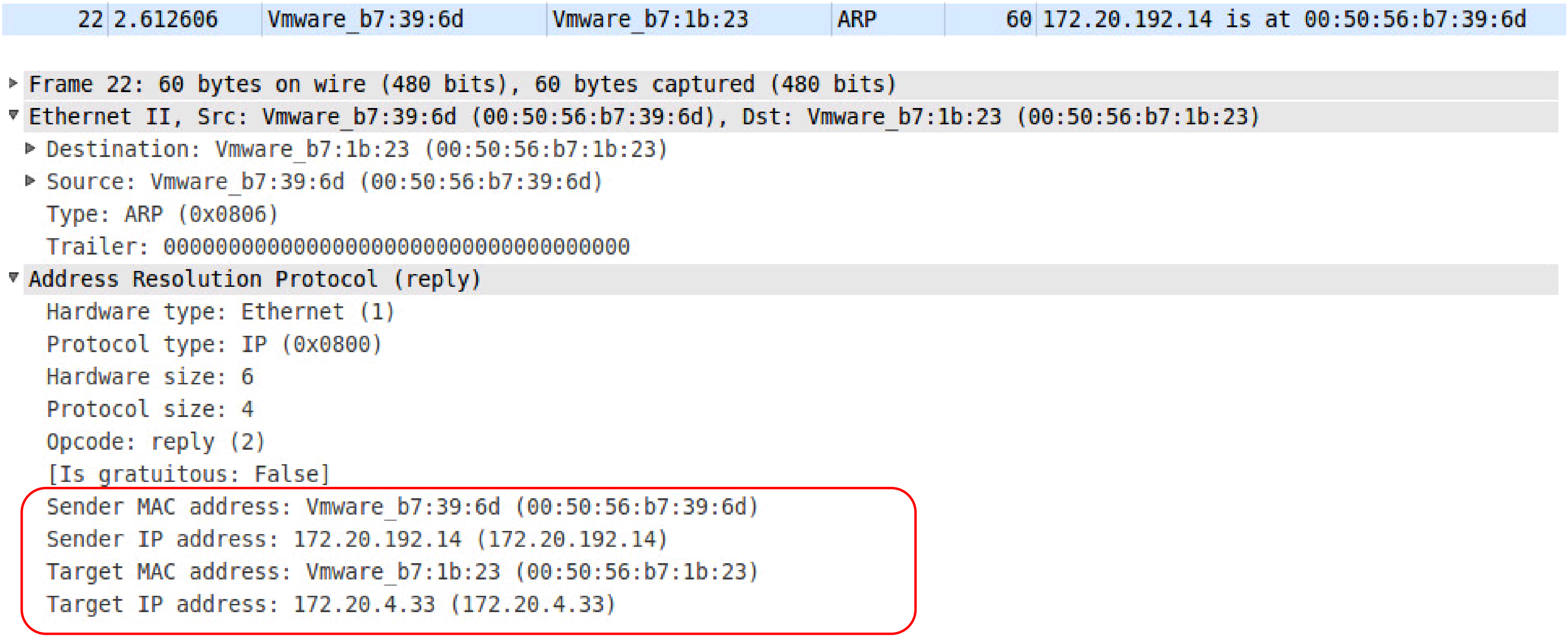
00:50:56:b7:1b:23

Elrond

eth0

.192.14

00:50:56:B7:39:6D



```
   22 2.612606    Vmware_b7:39:6d      Vmware_b7:1b:23      ARP        60 172.20.192.14 is at 00:50:56:b7:39:6d

▸ Frame 22: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
▾ Ethernet II, Src: Vmware_b7:39:6d (00:50:56:b7:39:6d), Dst: Vmware_b7:1b:23 (00:50:56:b7:1b:23)
  ▸ Destination: Vmware_b7:1b:23 (00:50:56:b7:1b:23)
  ▸ Source: Vmware_b7:39:6d (00:50:56:b7:39:6d)
    Type: ARP (0x0806)
    Trailer: 000000000000000000000000000000000000
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    [Is gratuitous: False]
    Sender MAC address: Vmware_b7:39:6d (00:50:56:b7:39:6d)
    Sender IP address: 172.20.192.14 (172.20.192.14)
    Target MAC address: Vmware_b7:1b:23 (00:50:56:b7:1b:23)
    Target IP address: 172.20.4.33 (172.20.4.33)
```

Drill down into the ARP reply packet

ARP Cache

New commands for your toolbox

- List ARP cache entries (IP/MAC pairs)

```
arp

arp -n        (no name resolution, faster)

arp -a        (uses BSD format for output)

ip neigh show      (shows more state information)
```

- Delete ARP cache entries (IP/MAC pairs)

```
ip neigh flush all
```

Showing the ARP cache

```
[root@elrond ~]# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
172.30.1.8                       (incomplete)                              eth0
172.30.1.196             ether   00:0C:29:BF:E4:F9   C                     eth0
172.30.1.108             ether   C8:00:0A:5C:00:00   C                     eth0
nosmo                    ether   00:0C:29:49:88:B8   C                     eth0
```

```
[root@elrond ~]# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.30.1.8                       (incomplete)                              eth0
172.30.1.196             ether   00:0C:29:BF:E4:F9   C                     eth0
172.30.1.108             ether   C8:00:0A:5C:00:00   C                     eth0
172.30.1.1               ether   00:0C:29:49:88:B8   C                     eth0
```

```
[root@elrond ~]# arp -a
? (172.30.1.8) at <incomplete> on eth0
? (172.30.1.196) at 00:0C:29:BF:E4:F9 [ether] on eth0
? (172.30.1.108) at C8:00:0A:5C:00:00 [ether] on eth0
nosmo (172.30.1.1) at 00:0C:29:49:88:B8 [ether] on eth0
```

***Incomplete** entries result from pings failing (device down or non-existent)*
***Complete** "C" means there is a complete MAC/IP pair*

Showing the ARP cache

```
[root@elrond ~]# ip neigh show
172.30.1.8 dev eth0  FAILED
172.30.1.196 dev eth0 lladdr 00:0c:29:bf:e4:f9 STALE
172.30.1.108 dev eth0 lladdr c8:00:0a:5c:00:00 STALE
172.30.1.1 dev eth0 lladdr 00:0c:29:49:88:b8 REACHABLE
```

ARP cache entry state	meaning	action if used
permanent	never expires; never verified	reset use counter
noarp	normal expiration; never verified	reset use counter
reachable	normal expiration	reset use counter
stale	still usable; needs verification	reset use counter; change state to delay
delay	schedule ARP request; needs verification	reset use counter
probe	sending ARP request	reset use counter
incomplete	first ARP request sent	send ARP request
failed	no response received	send ARP request

Source: http://linux-ip.net/html/ether-arp.html

Showing the ARP cache

Flags shown on ARP command output:

- Complete (C)

  Temporary ARP cache entries are aged out after several minutes.

- Permanent (M)

  Till next system restart

- Published (P)

  The system will act as a ARP server and respond to ARP requests for IP addresses that are not its own

*Note, there may be **incomplete** entries for failed ARP requests (pinging a non-existent or powered-off device) or entries that were manually deleted*

ARP commands on the different planets

[root@elrond ~]# arp -n

```
Address                  HWtype  HWaddress           Flags Mask            Iface
172.30.1.108             ether   C8:00:0A:5C:00:00   C                     eth0
172.30.1.1               ether   00:0C:29:49:88:B8   C                     eth0
```

R1#show arp

```
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.2.10            -   c800.0a5c.0001  ARPA   FastEthernet0/1
Internet  172.30.1.1              0   000c.2949.88b8  ARPA   FastEthernet0/0
Internet  172.30.1.107            8   000c.2968.3687  ARPA   FastEthernet0/0
Internet  172.30.1.108            -   c800.0a5c.0000  ARPA   FastEthernet0/0
```

C:\Users\Administrator>arp -a

```
Interface: 192.168.0.21 --- 0xe
  Internet Address      Physical Address      Type
  192.168.0.1           00-a0-c5-e1-c9-a8     dynamic
  192.168.0.2           00-0c-29-49-88-ae     dynamic
  192.168.0.12          00-14-38-9c-59-5f     dynamic
  192.168.0.18          00-24-8d-85-55-85     dynamic
  192.168.0.25          00-0c-6e-51-4c-2d     dynamic
  192.168.0.27          00-0c-f1-96-8e-68     dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  224.0.0.253           01-00-5e-00-00-fd     static
  239.192.152.143       01-00-5e-40-98-8f     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

New commands for your toolbox

- List ARP cache entry for a host

```
arp -a 172.30.1.1
```

- Add permanent ARP entries (lasts until next restart)

```
arp -s 172.30.1.1 00:b0:64:53:42:01     (add one IP/MAC entry)

arp -f /etc/ethers      (ASCII file of MAC/IP entries)
```

- Delete ARP entry

```
arp -d 172.30.1.1
```

arp command

Populate the arp cache via manual entries

Before

```
root@frodo:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.30.1.109             ether   00:19:b9:03:70:d4   C                     eth0
172.30.1.1               ether   00:b0:64:53:42:01   C                     eth0
```

Add permanent entry for a node

```
root@frodo:~# arp -s 172.30.1.1 00:b0:64:53:42:01
```

After

```
root@frodo:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.30.1.109             ether   00:19:b9:03:70:d4   C                     eth0
172.30.1.1               ether   00:b0:64:53:42:01   CM                    eth0
```

CM flags = Complete and Permanent

arp command

Populate the arp cache via ping usage

Before

```
root@frodo:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.30.1.109             ether   00:19:b9:03:70:d4   C                     eth0
172.30.1.1               ether   00:b0:64:53:42:01   CM                    eth0

root@frodo:~# ping 172.30.1.110
PING 172.30.1.110 (172.30.1.110) 56(84) bytes of data.
64 bytes from 172.30.1.110: icmp_seq=1 ttl=128 time=0.741 ms
< snipped >
root@frodo:~# ping 172.30.1.111
PING 172.30.1.111 (172.30.1.111) 56(84) bytes of data.
64 bytes from 172.30.1.111: icmp_seq=1 ttl=128 time=2.01 ms
< snipped >
```

After

```
root@frodo:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.30.1.1               ether   00:b0:64:53:42:01   CM                    eth0
172.30.1.109             ether   00:19:b9:03:70:d4   C                     eth0
172.30.1.111             ether   00:18:8b:28:ac:ab   C                     eth0
172.30.1.110             ether   00:19:b9:03:71:00   C                     eth0
```

Note the new entries for 172.30.1.110 and 172.30.1.111 that were added because of the last two pings.

arp command

Populate the arp cache via manual entries

Before

```
root@frodo:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.30.1.109             ether   00:19:b9:03:70:d4   C                     eth0
172.30.1.1               ether   00:b0:64:53:42:01   C                     eth0
```

Add permanent entry for a node

```
root@frodo:~# arp -s 172.30.1.1 00:b0:64:53:42:01
```

After

```
root@frodo:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.30.1.109             ether   00:19:b9:03:70:d4   C                     eth0
172.30.1.1               ether   00:b0:64:53:42:01   CM                    eth0
```

CM flags = Complete and Permanent

arp cache

Populating the arp cache via a file option

Before

```
root@frodo:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.30.1.109             ether   00:19:b9:03:70:d4   C                     eth0


root@frodo:~# vi /etc/ethers
root@frodo:~# cat /etc/ethers
172.30.1.1         00:b0:64:53:42:01
172.30.1.10        00:90:27:76:97:ab

root@frodo:~# arp -f /etc/ethers
```

Permanent entries can also be added from a file using the -f option.

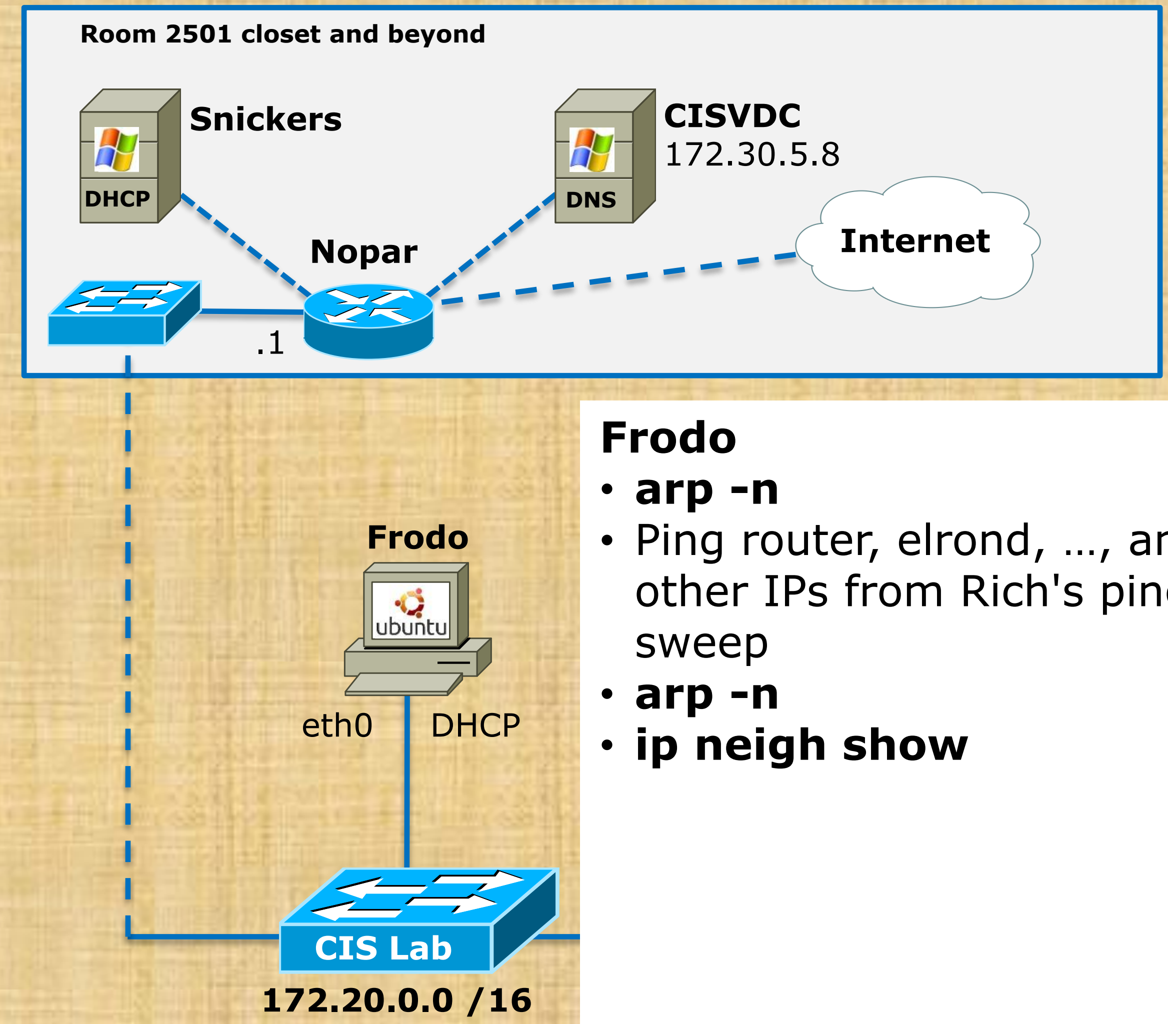
After

```
root@frodo:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.30.1.1               ether   00:b0:64:53:42:01   CM                    eth0
172.30.1.109             ether   00:19:b9:03:70:d4   C                     eth0
172.30.1.10              ether   00:90:27:76:97:ab   CM                    eth0
```

CM flags = Complete and Permanent

Class Activity

Populate Frodo's ARP cache



Frodo

- **arp -n**
- Ping router, elrond, …, and other IPs from Rich's ping sweep
- **arp -n**
- **ip neigh show**

```
root@p02-frodo:~# nmap -sP
172.20.4.1-254 | grep 172
Nmap scan report for 172.20.4.12
Nmap scan report for 172.20.4.18
Nmap scan report for 172.20.4.21
Nmap scan report for 172.20.4.22
Nmap scan report for 172.20.4.28
Nmap scan report for 172.20.4.29
Nmap scan report for 172.20.4.31
Nmap scan report for 172.20.4.32
Nmap scan report for 172.20.4.33
Nmap scan report for 172.20.4.34
Nmap scan report for 172.20.4.38
Nmap scan report for 172.20.4.42
Nmap scan report for 172.20.4.45
Nmap scan report for 172.20.4.49
Nmap scan report for 172.20.4.56
Nmap scan report for 172.20.4.65
Nmap scan report for 172.20.4.71
Nmap scan report for 172.20.4.72
Nmap scan report for 172.20.4.105
Nmap scan report for 172.20.4.110
Nmap scan report for 172.20.4.184
Nmap scan report for 172.20.4.188
```

Installing Commands

Installing software packages

Red Hat family:

rpm -qa | grep *package*
yum install *package*
yum remove *package*

Debian family:

dpkg -l | grep *package*
apt-get install *package*
apt-get remove *package*

New commands for your toolbox

Red Hat family:

rpm -qa | grep *package*	*Check if package is installed*
yum install *package*	*Install package*
yum remove *package*	*Remove package*

Debian family:

dpkg -l | grep *package*	*Check if package is installed*
apt-get install *package*	*Install package*
apt-get remove *package*	*Remove package*

1) No mtr command found

```
[root@p14-elrond ~]# mtr -rc1 snickers.cislab.net
-bash: mtr: command not found
[root@p14-elrond ~]#
```

2) Install mtr package (must be logged in as root)

```
[root@p14-elrond ~]# yum install mtr
Loaded plugins: fastestmirror, presto
Loading mirror speeds from cached hostfile
 * base: mirror.pac-12.org
 * extras: mirrors.usc.edu
 * updates: mirror.hmc.edu
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package mtr.x86_64 2:0.75-5.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolve

====================
 Package         Arch
====================
Installing:
 mtr             x86_

Transaction Summary
====================
Install       1 Pack

Total download size:
Installed size: 96 k
Is this ok [y/N]: y
Downloading Packages:
Setting up and reading Presto delta metadata
Processing delta metadata
Package(s) data still to download: 54 k
mtr-0.75-5.el6.x86_64.rpm                                |  54 kB     00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : 2:mtr-0.75-5.el6.x86_64                                      1/1
  Verifying  : 2:mtr-0.75-5.el6.x86_64                                      1/1

Installed:
  mtr.x86_64 2:0.75-5.el6

Complete!
[root@p14-elrond ~]#
```

3) Run new mtr command

```
[root@p14-elrond ~]# mtr -rc1 snickers.cislab.net
HOST: p14-elrond.rivendell           Loss%   Snt   Last   Avg  Best  Wrst StDev
  1. nopar.cislab.net                 0.0%     1    0.5   0.5   0.5   0.5   0.0
  2. snickers.cislab.net              0.0%     1    0.6   0.6   0.6   0.6   0.0
[root@p14-elrond ~]#
```

CIS 192 - Lesson 2

ubuntu

debian

Linux Mint from freedom came elegance

1) No traceroute command

```
root@p14-frodo:~# traceroute google.com
The program 'traceroute' can be found in the following packages:
 * inetutils-traceroute
 * traceroute
Try: apt-get install <selected package>
root@p14-frodo:~#
```

2) Install traceroute package (must be logged in as root)

```
root@p14-frodo:~# apt-get install traceroute
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  traceroute
0 upgraded, 1 newly installed, 0 to remove and 265 not upgraded.
Need to get 53.1 kB of archives.
After this operation, 162 kB of additional disk space will be used.
WARNING: The following packages cannot be authenticated!
  traceroute
Install thes
Get:1 http:/
Fetched 53.1
Selecting pr
(Reading dat
Unpacking tr
Processing t
Setting up t
update-alter
update-alter
update-alter
update-alter
auto mode.
root@p14-fro
```

3) Run new traceroute command

```
root@p14-frodo:~# traceroute google.com
traceroute to google.com (74.125.224.136), 30 hops max, 60 byte packets
 1  172.20.0.1 (172.20.0.1)  1.451 ms  1.827 ms  1.951 ms
 2  10.98.1.2 (10.98.1.2)  1.086 ms  1.196 ms  1.195 ms
 3  cenic-egm-gw.cabrillo.edu (207.62.184.4)  1.916 ms  1.998 ms  2.032 ms
 4  dc-oak-dc1--cab-cc-egm.cenic.net (137.164.34.120)  4.361 ms  4.304 ms  4.383 ms
 5  dc-oak-core1--oak-agg1-10ge.cenic.net (137.164.47.113)  5.778 ms  5.826 ms  5.864 ms
 6  dc-paix-px1--oak-core1-ge.cenic.net (137.164.47.18)  5.864 ms  5.365 ms  5.383 ms
 7  google--paix-px1.cenic.net (198.32.251.198)  5.562 ms  6.021 ms  5.655 ms
 8  216.239.49.250 (216.239.49.250)  6.434 ms  6.776 ms  7.420 ms
 9  64.233.174.119 (64.233.174.119)  7.809 ms  6.991 ms  7.846 ms
10  nuq04s09-in-f8.1e100.net (74.125.224.136)  7.219 ms  7.250 ms  6.704 ms
root@p14-frodo:~#
```

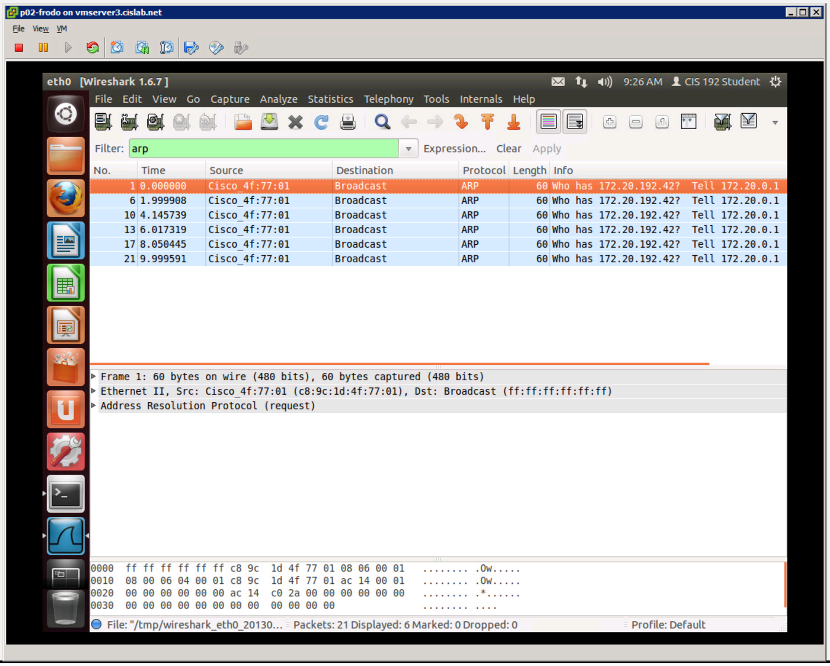
1) No wireshark command

```
cis90@p02-frodo:~$ wireshark &
[1] 3380
cis90@p02-frodo:~$ The program 'wireshark' is currently not installed.  To
run 'wireshark' please ask your administrator to install the package
'wireshark'
```

2) Install wireshark package (must be logged in as root)

```
root@p02-frodo:~# apt-get install wireshark
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libc-ares2 libsmi2ldbl libwireshark-data libwireshark1 libwiretap1
  libwsutil1 wireshark-common
Suggested packages:
  snmp-mibs-downloader wireshark-doc
The following NEW packages will be installed:
  libc-ares2 libsmi2ldbl libwireshark-data libwireshark1 libwiretap1
  libwsutil1 wireshark wireshark-common
0 upgraded, 8 newly installed, 0 to remove and 355 not upgraded.
Need to get 14.9 MB of archives.
After this operation, 62.2 MB of additional disk space will be used.
Do you want to continue [Y/n]? y
Get:1 http://us.archive.ubuntu.com/ubuntu/ precise/main libc-ares2 amd64 1.7.5-1 [36.6 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu/ precise/universe libsmi2ldbl amd64 0.4.8+dfsg2-4build1 [335 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu/ precise/universe libwireshark-data all 1.6.7-1 [1,155 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu/ precise/universe libwsutil1 amd64 1.6.7-1 [12.1 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu/ precise/universe libwiretap1 amd64 1.6.7-1 [144 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu/ precise/universe libwireshark1 amd64 1.6.7-1 [12.2 MB]
Get:7 http://us.archive.ubuntu.com/ubuntu/ precise/universe wireshark-common amd64 1.6.7-1 [186 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu/ precise/universe wireshark amd64 1.6.7-1 [848 kB]
Fetched 14.9 MB in 12s (1,218 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libc-ares2.
(Reading database ... 142443 files and directories currently installed.)
Unpacking libc-ares2 (from .../libc-ares2_1.7.5-1_amd64.deb) ...
Selecting previously unselected package libsmi2ldbl.
Unpacking libsmi2ldbl (from .../libsmi2ldbl_0.4.8+dfsg2-4build1_amd64.deb) ...
Selecting previously unselected package libwireshark-data.
Unpacking libwireshark-data (from .../libwireshark-data_1.6.7-1_all.deb) ...
Selecting previously unselected package libwsutil1.
Unpacking libwsutil1 (from .../libwsutil1_1.6.7-1_amd64.deb) ...
Selecting previously unselected package libwiretap1.
Unpacking libwiretap1 (from .../libwiretap1_1.6.7-1_amd64.deb) ...
Selecting previously unselected package libwireshark1.
Unpacking libwireshark1 (from .../libwireshark1_1.6.7-1_amd64.deb) ...
Selecting previously unselected package wireshark-common.
Unpacking wireshark-common (from .../wireshark-common_1.6.7-1_amd64.deb) ...
Selecting previously unselected package wireshark.
Unpacking wireshark (from .../wireshark_1.6.7-1_amd64.deb) ...
Processing triggers for man-db ...
Processing triggers for desktop-file-utils ...
Processing triggers for bamfdaemon ...
Rebuilding /usr/share/applications/bamf.index...
Processing triggers for gnome-menus ...
Processing triggers for hicolor-icon-theme ...
Setting up libc-ares2 (1.7.5-1) ...
Setting up libsmi2ldbl (0.4.8+dfsg2-4build1) ...
Setting up libwireshark-data (1.6.7-1) ...
Setting up libwsutil1 (1.6.7-1) ...
Setting up libwiretap1 (1.6.7-1) ...
Setting up libwireshark1 (1.6.7-1) ...
Setting up wireshark-common (1.6.7-1) ...
Setting up wireshark (1.6.7-1) ...
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
root@p02-frodo:~#
```

*3) Run **wireshark &** from the command line*


p02-frodo on vmserver3.cislab.net
File View VM
eth0 [Wireshark 1.6.7]
9:26 AM
CIS 192 Student
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply

Filter: arp

```
No. Time     Source         Destination Protocol Length Info
  1 0.000000 Cisco_4f:77:01 Broadcast   ARP          60 Who has 172.20.192.42?  Tell 172.20.0.1
  6 1.999908 Cisco_4f:77:01 Broadcast   ARP          60 Who has 172.20.192.42?  Tell 172.20.0.1
 10 4.145739 Cisco_4f:77:01 Broadcast   ARP          60 Who has 172.20.192.42?  Tell 172.20.0.1
 13 6.017319 Cisco_4f:77:01 Broadcast   ARP          60 Who has 172.20.192.42?  Tell 172.20.0.1
 17 8.050445 Cisco_4f:77:01 Broadcast   ARP          60 Who has 172.20.192.42?  Tell 172.20.0.1
 21 9.999591 Cisco_4f:77:01 Broadcast   ARP          60 Who has 172.20.192.42?  Tell 172.20.0.1

▸ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
▸ Ethernet II, Src: Cisco_4f:77:01 (c8:9c:1d:4f:77:01), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸ Address Resolution Protocol (request)

0000  ff ff ff ff ff ff c8 9c  1d 4f 77 01 08 06 00 01   ........ .Ow.....
0010  08 00 06 04 00 01 c8 9c  1d 4f 77 01 ac 14 00 01   ........ .Ow.....
0020  00 00 00 00 00 00 ac 14  c0 2a 00 00 00 00 00 00   ........ .*......
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....

File: "/tmp/wireshark_eth0_20130...  Packets: 21 Displayed: 6 Marked: 0 Dropped: 0  Profile: Default
```

Class Activity - Install mail on Elrond

1) Install sendmail and mailx packages

```
rpm -qa | grep sendmail
yum install sendmail
service sendmail start

rpm -qa | grep mailx
yum install mailx
```

2) Test that you can send and receive mail

```
root@p14-elrond ~]# mail root
Subject: test email
Mailx has been installed
.
EOT
[root@p14-elrond ~]# mail
Heirloom Mail version 12.4 7/29/08.  Type ? for help.
"/var/spool/mail/root": 9 messages 8 new
    1 CIS 192 Student       Sun Feb 17 10:07  20/802   "test email"
& q
```

arpwatch

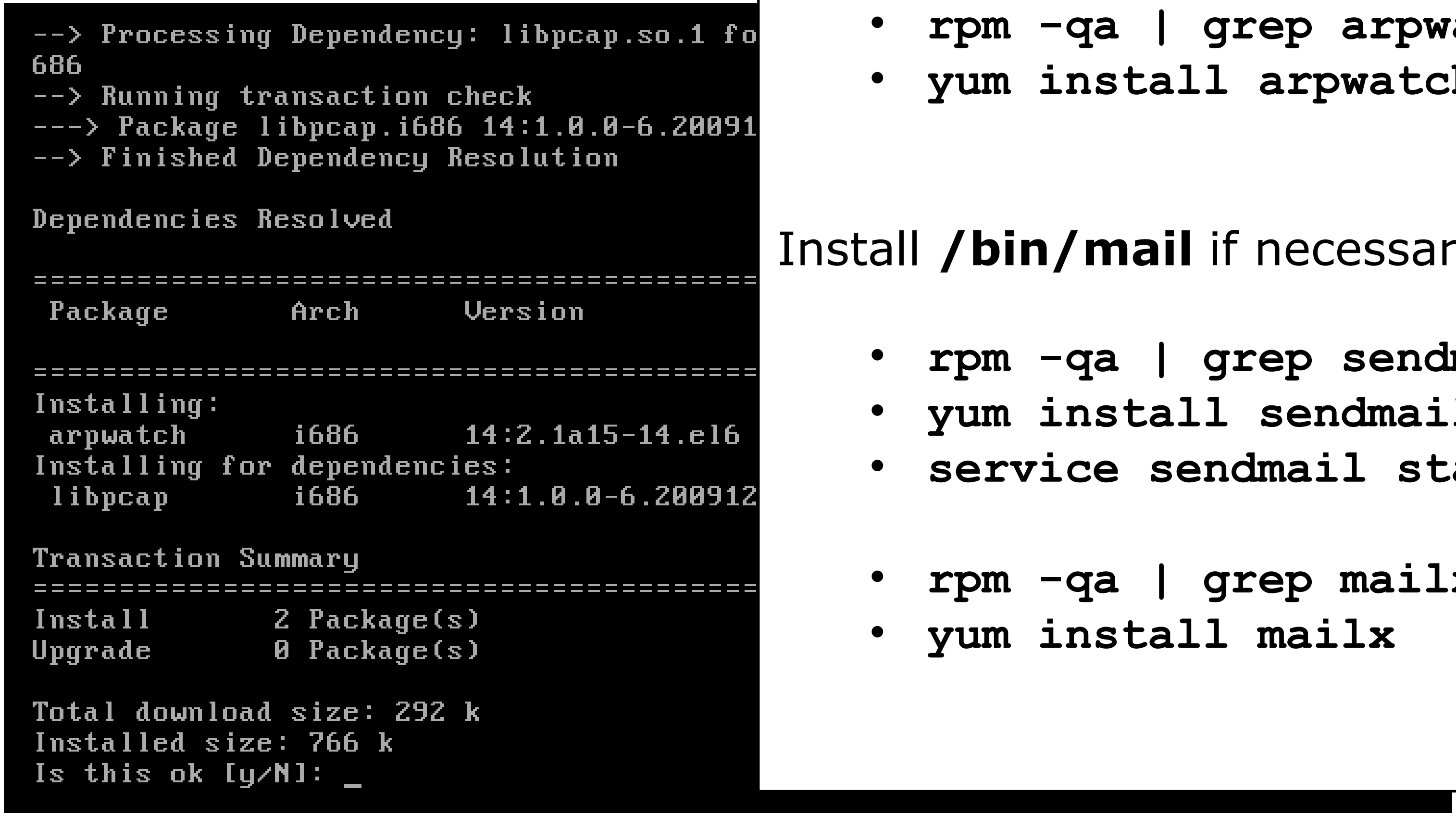
arpwatch

Track IP/MAC pairs

The arpwatch daemon

- Collects IP/MAC address pairs
- Save pairs in a log file: arp.dat
- Emails root as pairs are found
- Great way to inventory MAC addresses or monitor for fraudulent activity

arpwatch installation (Red Hat family)



```
--> Processing Dependency: libpcap.so.1 fo
686
--> Running transaction check
---> Package libpcap.i686 14:1.0.0-6.20091
--> Finished Dependency Resolution

Dependencies Resolved

============================================
 Package        Arch       Version

============================================
Installing:
 arpwatch       i686       14:2.1a15-14.el6
Installing for dependencies:
 libpcap        i686       14:1.0.0-6.200912

Transaction Summary
============================================
Install       2 Package(s)
Upgrade       0 Package(s)

Total download size: 292 k
Installed size: 766 k
Is this ok [y/N]: _
```

Install **arpwatch** if necessary:

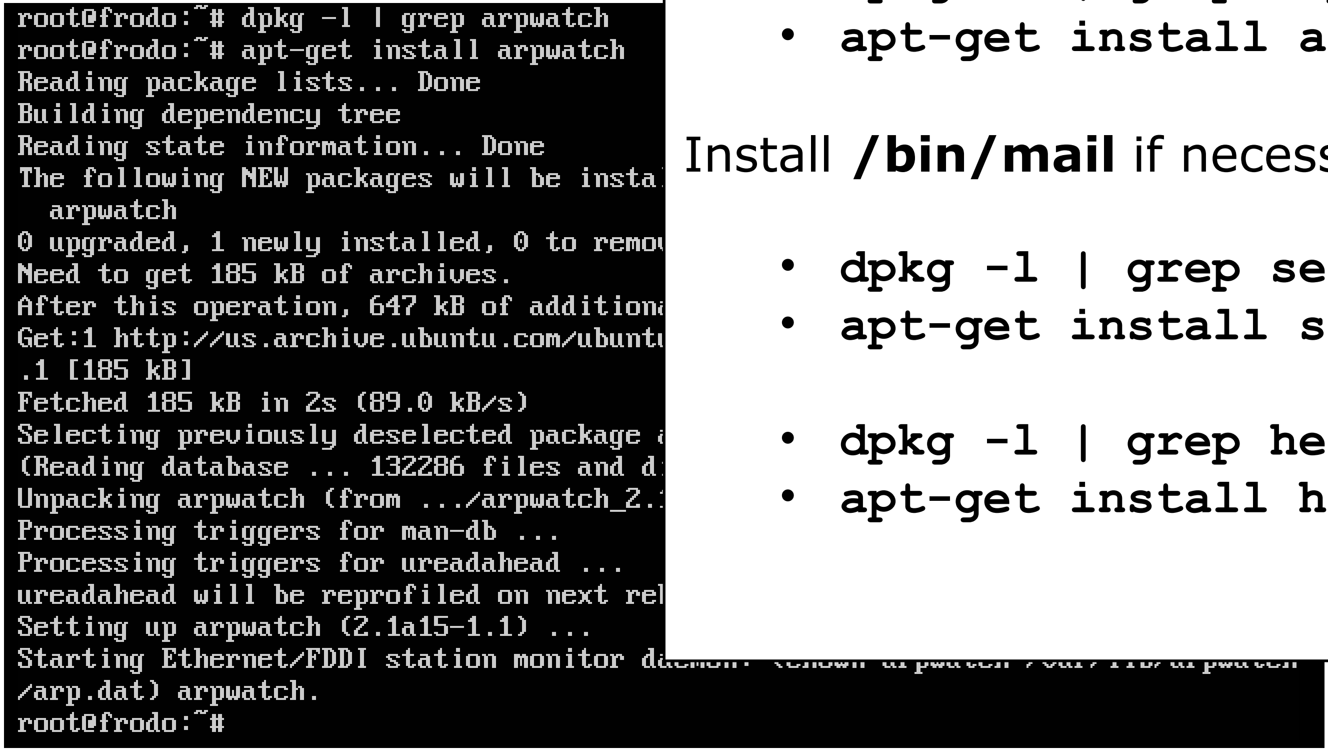
- **rpm -qa | grep arpwatch**
- **yum install arpwatch**

Install **/bin/mail** if necessary:

- **rpm -qa | grep sendmail**
- **yum install sendmail**
- **service sendmail start**

- **rpm -qa | grep mailx**
- **yum install mailx**

arpwatch installation (Debian family)



```
root@frodo:~# dpkg -l | grep arpwatch
root@frodo:~# apt-get install arpwatch
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be insta
  arpwatch
0 upgraded, 1 newly installed, 0 to remov
Need to get 185 kB of archives.
After this operation, 647 kB of additiona
Get:1 http://us.archive.ubuntu.com/ubuntu
.1 [185 kB]
Fetched 185 kB in 2s (89.0 kB/s)
Selecting previously deselected package a
(Reading database ... 132286 files and di
Unpacking arpwatch (from .../arpwatch_2.1
Processing triggers for man-db ...
Processing triggers for ureadahead ...
ureadahead will be reprofiled on next rel
Setting up arpwatch (2.1a15-1.1) ...
Starting Ethernet/FDDI station monitor daemon: (chown arpwatch /var/lib/arpwatch
/arp.dat) arpwatch.
root@frodo:~#
```

Install **arpwatch** if necessary:

- **`dpkg -l | grep arpwatch`**
- **`apt-get install arpwatch`**

Install **/bin/mail** if necessary:

- **`dpkg -l | grep sendmail`**
- **`apt-get install sendmail`**

- **`dpkg -l | grep heirloom-mailx`**
- **`apt-get install heirloom-mailx`**

arpwatch

Collect MAC / IP pairs

[Red Hat family] **service arpwatch start**

or [Red Hat or Debian family] **/etc/init.d/arpwatch start**

> *The collection starts now. As new pairs are detected they get emailed.*
> *arp.dat file is not updated till arpwatch is restarted*

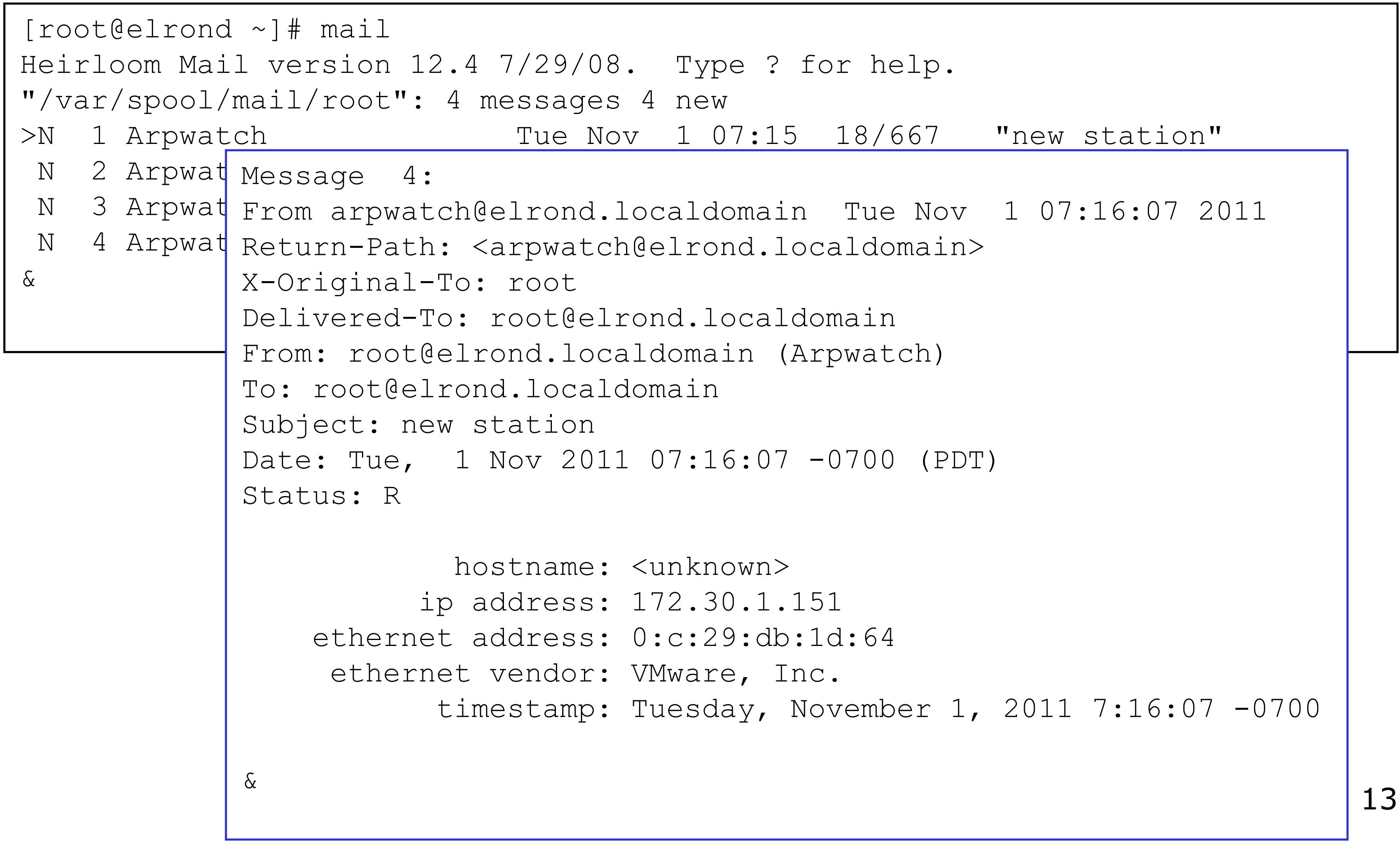
[Red Hat family] **service arpwatch restart**

or [Red Hat or Debian family] **/etc/init.d/arpwatch restart**

```
[root@elrond ~]# cat /var/lib/arpwatch/arp.dat
0:b:fc:28:41:0      172.30.1.5      1234303973
0:c:29:a4:83:bc     172.30.1.126    1234303772
0:13:7f:55:f9:0     172.30.1.4      1234303973
0:3:e3:6c:77:80     172.30.1.3      1234303973
0:b0:64:53:42:1     172.30.1.1      1234303772
0:18:8b:28:ac:50    172.30.1.121    1234304404
0:19:b9:3:71:f5     172.30.1.120    1234304072
0:90:27:76:97:ab    172.30.1.10     1234304341
0:c:29:e4:be:d3     172.30.1.152    1234303463
0:19:b9:3:71:cc     172.30.1.103    1234303636
0:c:29:46:5:73      172.30.1.153    1234303945
```

arpwatch

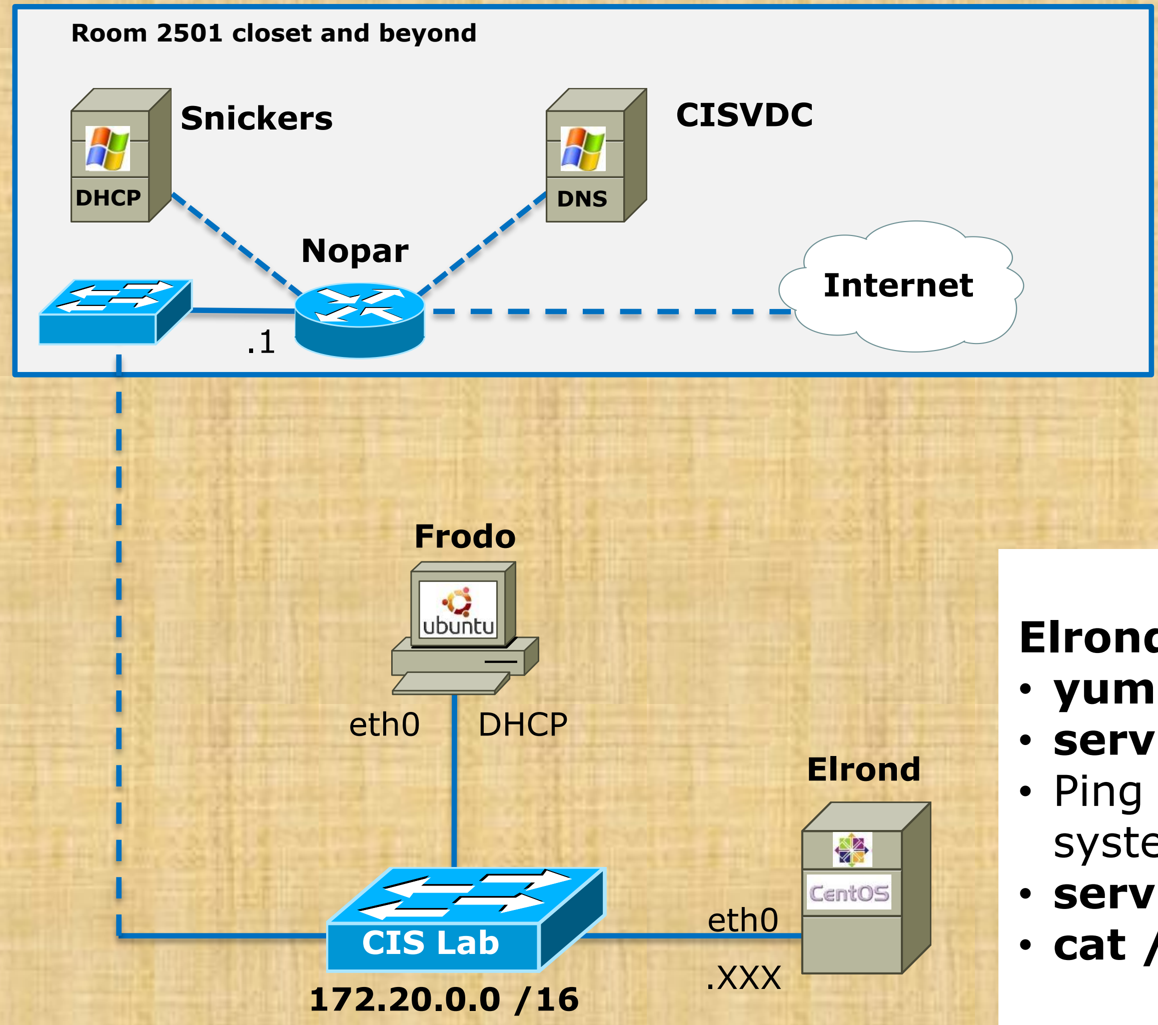
New pairs are emailed



```
[root@elrond ~]# mail
Heirloom Mail version 12.4 7/29/08.  Type ? for help.
"/var/spool/mail/root": 4 messages 4 new
>N  1 Arpwatch              Tue Nov  1 07:15  18/667   "new station"
 N  2 Arpwat
 N  3 Arpwat
 N  4 Arpwat
&
```

```
Message  4:
From arpwatch@elrond.localdomain  Tue Nov  1 07:16:07 2011
Return-Path: <arpwatch@elrond.localdomain>
X-Original-To: root
Delivered-To: root@elrond.localdomain
From: root@elrond.localdomain (Arpwatch)
To: root@elrond.localdomain
Subject: new station
Date: Tue,  1 Nov 2011 07:16:07 -0700 (PDT)
Status: R

            hostname: <unknown>
          ip address: 172.30.1.151
    ethernet address: 0:c:29:db:1d:64
     ethernet vendor: VMware, Inc.
           timestamp: Tuesday, November 1, 2011 7:16:07 -0700

&
```

Class Activity - Setting up arpwatch on Elrond



Try it!

Elrond
- **yum install arpwatch**
- **service arpwatch start**
- Ping some other 172.20.x.x systems
- **service arpwatch restart**
- **cat /var/arpwatch/arp.dat**

Q3

Viewing Packets with tcpdump

Viewing Network Packets

Some sniffer options:

- Use tcpdump command on the Linux systems
- Run Wireshark on the Classroom or Lab PCs
- Run Wireshark on the William VM (has Wireshark installed)
- Install and run Wireshark on the Ubuntu VMs (which have graphics mode)

Sniffer software like Wireshark puts the NIC in promiscuous mode so it will see all the packets on the line rather than just its own.

Viewing Network Packets
tcpdump

```
[root@elrond ~]# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol
decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535
bytes
08:48:35.555899 IP 172.30.1.125.ssh > 172.30.1.100.49326: Flags [P.],
seq 1215753462:1215753658, ack 2360465031, win 317, length 196
08:48:35.556202 IP 172.30.1.100.49326 > 172.30.1.125.ssh: Flags [.],
ack 196, win 254, length 0
08:48:35.557680 IP 172.30.1.125.48727 > cisvdc1.cisvlab.net.domain:
6647+ PTR? 100.1.30.172.in-addr.arpa. (43)
08:48:35.558483 IP cisvdc1.cisvlab.net.domain > 172.30.1.125.48727:
6647 NXDomain* 0/1/0 (130)
08:48:35.558704 ARP, Request who-has snickers.cisvlab.net
(00:13:20:c6:a4:16 (oui Unknown)) tell 172.30.1.100, length 46
08:48:35.558768 ARP, Reply snickers.cisvlab.net is-at
00:13:20:c6:a4:16 (oui Unknown), length 46
<continues like this>
```

***Ctrl-s** to pause **Ctrl-q** to continue **Ctrl-c** to end*

Viewing Network Packets
tcpdump

```
[root@elrond ~]# tcpdump -c5 arp or icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
08:55:58.135729 IP snickers.cisvlab.net > 172.30.1.100: ICMP echo request, id
1280, seq 13402, length 80
08:55:58.135742 IP 172.30.1.100 > snickers.cisvlab.net: ICMP echo reply, id
1280, seq 13402, length 80
08:55:58.139540 ARP, Request who-has 172.30.1.1 tell 172.30.1.125, length 28
08:55:58.140088 ARP, Reply 172.30.1.1 is-at c8:9c:1d:4f:77:01 (oui Unknown),
length 46
08:55:58.359346 IP snickers.cisvlab.net > 172.30.1.100: ICMP echo request, id
1280, seq 13658, length 80
5 packets captured
8 packets received by filter
0 packets dropped by kernel
[root@elrond ~]#
```

Using the -c option to limit the capture to 5 packets and filter out anything but arp or icmp packets

Viewing Network Packets
tcpdump on Elrond

```
[root@elrond ~]# tcpdump -c5 arp or icmp and host 172.30.1.125
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
08:59:12.957730 ARP, Request who-has 172.30.1.125 tell 172.30.1.150, length 46
08:59:12.958153 ARP, Reply 172.30.1.125 is-at 00:0c:29:d8:84:7f (oui Unknown),
length 28
08:59:12.958444 IP 172.30.1.150 > 172.30.1.125: ICMP echo request, id 2428,
seq 1, length 64
08:59:12.958612 IP 172.30.1.125 > 172.30.1.150: ICMP echo reply, id 2428, seq
1, length 64
08:59:13.940973 IP 172.30.1.150 > 172.30.1.125: ICMP echo request, id 2428,
seq 2, length 64
5 packets captured
13 packets received by filter
0 packets dropped by kernel
[root@elrond ~]#
```

Using the -c option to limit the capture to 5 packets and filter out anything but arp or icmp packets for host 172.30.1.125

Viewing Network Packets
tcpdump

```
[root@elrond ~]# tcpdump -c5 arp or icmp and host 172.30.1.125 > capture
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
5 packets captured
6 packets received by filter
0 packets dropped by kernel

[root@elrond ~]# cat capture
09:01:01.943495 IP 172.30.1.150 > 172.30.1.125: ICMP echo request, id 2428,
seq 110, length 64
09:01:01.943564 IP 172.30.1.125 > 172.30.1.150: ICMP echo reply, id 2428, seq
110, length 64
09:01:02.943255 IP 172.30.1.150 > 172.30.1.125: ICMP echo request, id 2428,
seq 111, length 64
09:01:02.943332 IP 172.30.1.125 > 172.30.1.150: ICMP echo reply, id 2428, seq
111, length 64
09:01:03.943654 IP 172.30.1.150 > 172.30.1.125: ICMP echo request, id 2428,
seq 112, length 64
[root@elrond ~]#
```

Same as before but saving the captured packets in a file

Viewing Network Packets
tcpdump

```
[root@elrond ~]# tcpdump src 172.30.1.150 or dst 172.30.1.150
tcpdump: verbose output suppressed, use -v or -vv for full protocol
  decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535
  bytes
09:05:35.763345 IP 172.30.1.150 > 172.30.1.125: ICMP echo request, id
  2469, seq 93, length 64
09:05:35.763413 IP 172.30.1.125 > 172.30.1.150: ICMP echo reply, id
  2469, seq 93, length 64
09:05:35.767609 IP 172.30.1.150.ssh > 172.30.1.100.49329: Flags [P.],
  seq 3250995165:3250995265, ack 256292814, win 591, length 100
09:05:35.972475 IP 172.30.1.100.49329 > 172.30.1.150.ssh: Flags [.],
  ack 100, win 255, length 0
^C
8 packets captured
9 packets received by filter
0 packets dropped by kernel
[root@elrond ~]#
```

View all packets coming or going from 172.30.4.125

Viewing Network Packets
tcpdump

Provide link-level header
Buffer stdout
Don't convert addresses to names

```
[root@elrond ~]# tcpdump -eln src 172.30.1.105 or dst 172.30.1.105
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
11:23:35.938846 00:0c:29:a4:83:bc > 00:19:b9:03:70:b3, ethertype IPv4 (0x0800),
  length 98: 172.30.1.126 > 172.30.1.105: ICMP echo request, id 54547, seq 1,
  length 64
11:23:35.939741 00:19:b9:03:70:b3 > Broadcast, ethertype ARP (0x0806), length 60:
  arp who-has 172.30.1.126 tell 172.30.1.105
11:23:35.939769 00:0c:29:a4:83:bc > 00:19:b9:03:70:b3, ethertype ARP (0x0806),
  length 42: arp reply 172.30.1.126 is-at 00:0c:29:a4:83:bc
11:23:35.940051 00:19:b9:03:70:b3 > 00:0c:29:a4:83:bc, ethertype IPv4 (0x0800),
  length 98: 172.30.1.105 > 172.30.1.126: ICMP echo reply, id 54547, seq 1,
  length 64
```

Ctrl-C to end

```
4 packets captured
12 packets received by filter
0 packets dropped by kernel
[root@elrond ~]#
```

Show all packets with a source and destination IP address of 172.30.1.105

Class Activity
tcpdump

Try it!

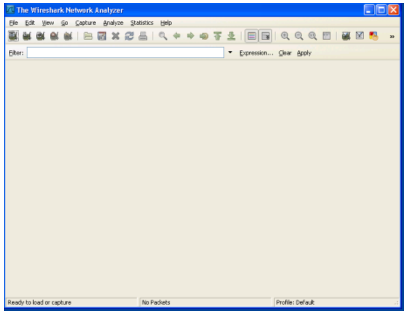
- [Elrond] **yum install tcpdump**
- [Elrond] **tcpdump**
- [Elrond] **tcpdump -c10 icmp or arp**

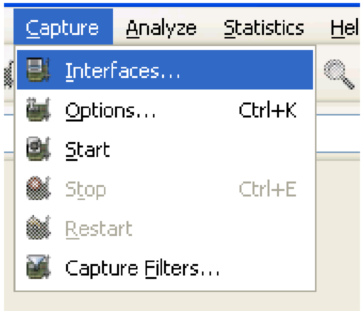
Ctrl-s to pause **Ctrl-q** to continue **Ctrl-c** to end

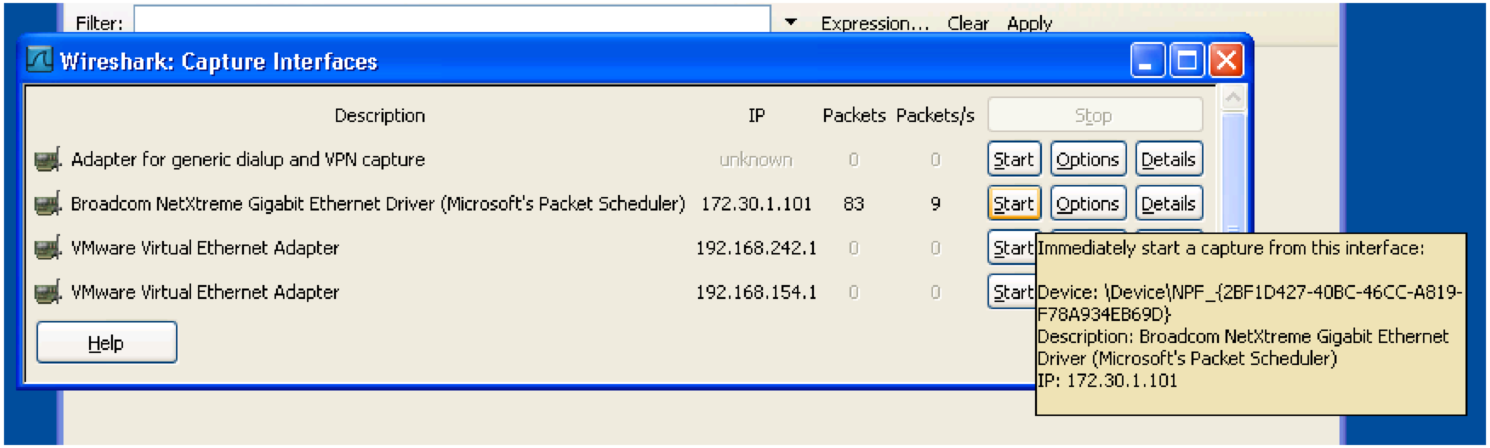
Viewing Packets with wireshark

Viewing Network Packets with Wireshark

Wireshark


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply
Ready to load or capture
No Packets
Profile: Default





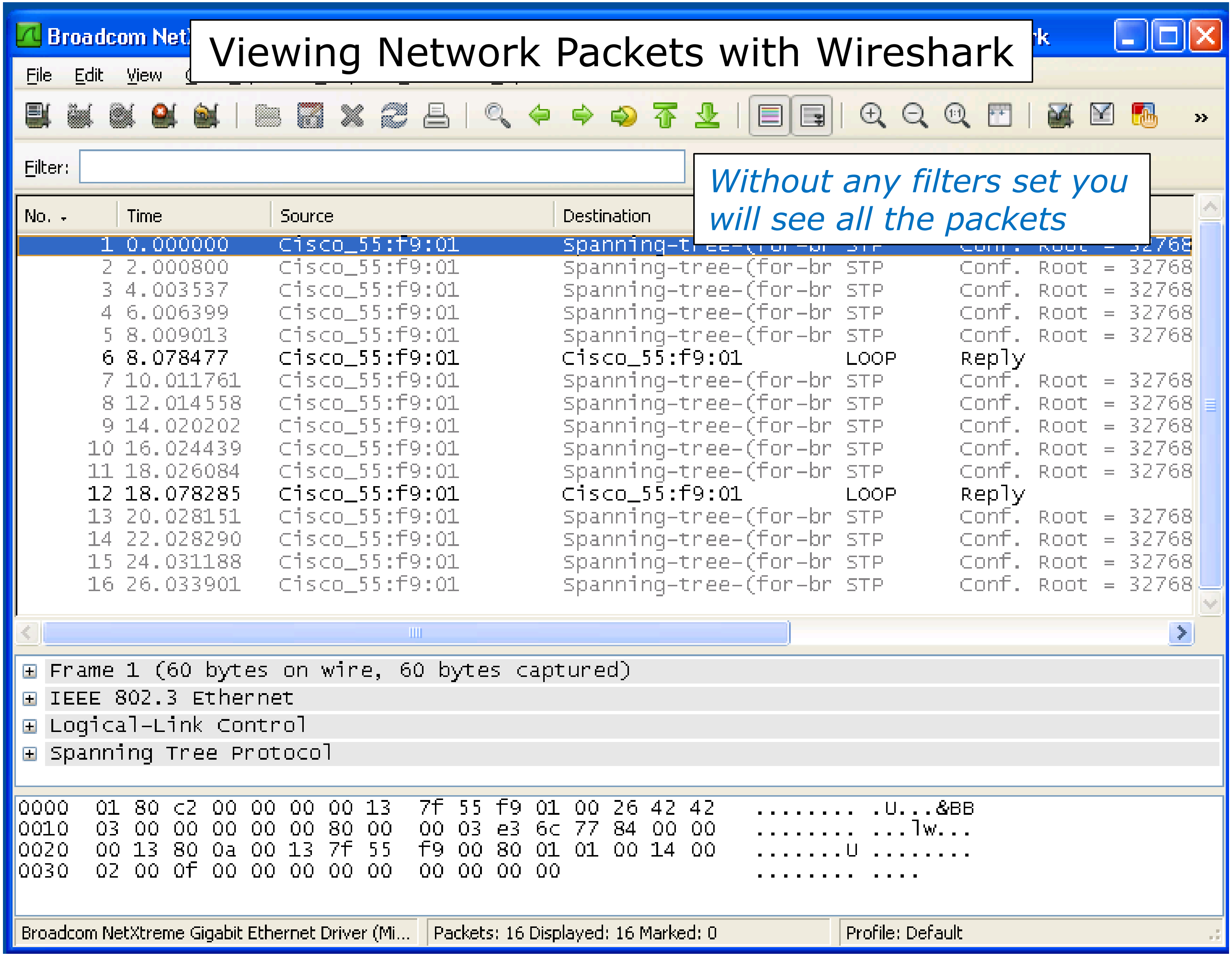
Description	IP	Packets	Packets/s			
Adapter for generic dialup and VPN capture	unknown	0	0	Start	Options	Details
Broadcom NetXtreme Gigabit Ethernet Driver (Microsoft's Packet Scheduler)	172.30.1.101	83	9	Start	Options	Details
VMware Virtual Ethernet Adapter	192.168.242.1	0	0	Start		
VMware Virtual Ethernet Adapter	192.168.154.1	0	0	Start		

Immediately start a capture from this interface:

Device: \Device\NPF_{2BF1D427-40BC-46CC-A819-F78A934EB69D}
Description: Broadcom NetXtreme Gigabit Ethernet Driver (Microsoft's Packet Scheduler)
IP: 172.30.1.101

Select the interface to listen on and click Start button

Viewing Network Packets with Wireshark



Without any filters set you will see all the packets

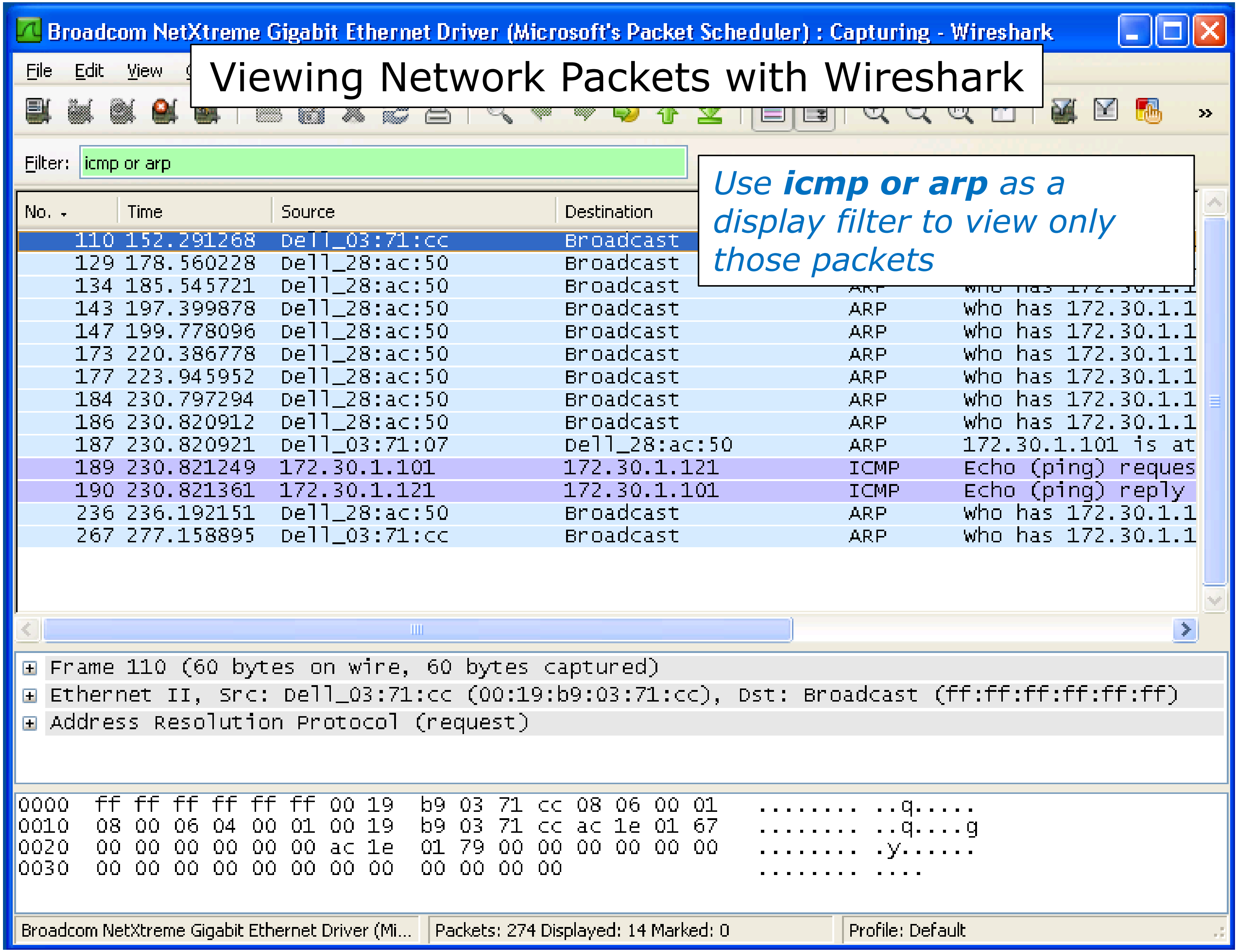
```
No. -    Time         Source            Destination
   1 0.000000    Cisco_55:f9:01    Spanning-tr                              Conf. Root = 32768
   2 2.000800    Cisco_55:f9:01    Spanning-tree-(for-br STP    Conf. Root = 32768
   3 4.003537    Cisco_55:f9:01    Spanning-tree-(for-br STP    Conf. Root = 32768
   4 6.006399    Cisco_55:f9:01    Spanning-tree-(for-br STP    Conf. Root = 32768
   5 8.009013    Cisco_55:f9:01    Spanning-tree-(for-br STP    Conf. Root = 32768
   6 8.078477    Cisco_55:f9:01    Cisco_55:f9:01        LOOP   Reply
   7 10.011761   Cisco_55:f9:01    Spanning-tree-(for-br STP    Conf. Root = 32768
   8 12.014558   Cisco_55:f9:01    Spanning-tree-(for-br STP    Conf. Root = 32768
   9 14.020202   Cisco_55:f9:01    Spanning-tree-(for-br STP    Conf. Root = 32768
  10 16.024439   Cisco_55:f9:01    Spanning-tree-(for-br STP    Conf. Root = 32768
  11 18.026084   Cisco_55:f9:01    Spanning-tree-(for-br STP    Conf. Root = 32768
  12 18.078285   Cisco_55:f9:01    Cisco_55:f9:01        LOOP   Reply
  13 20.028151   Cisco_55:f9:01    Spanning-tree-(for-br STP    Conf. Root = 32768
  14 22.028290   Cisco_55:f9:01    Spanning-tree-(for-br STP    Conf. Root = 32768
  15 24.031188   Cisco_55:f9:01    Spanning-tree-(for-br STP    Conf. Root = 32768
  16 26.033901   Cisco_55:f9:01    Spanning-tree-(for-br STP    Conf. Root = 32768
```

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

```
0000  01 80 c2 00 00 00 00 13  7f 55 f9 01 00 26 42 42   ........ .U...&BB
0010  03 00 00 00 00 00 80 00  00 03 e3 6c 77 84 00 00   ........ ...lw...
0020  00 13 80 0a 00 13 7f 55  f9 00 80 01 01 00 14 00   .......U ........
0030  02 00 0f 00 00 00 00 00  00 00 00 00               ........ ....
```

Broadcom NetXtreme Gigabit Ethernet Driver (Mi... | Packets: 16 Displayed: 16 Marked: 0 | Profile: Default

Viewing Network Packets with Wireshark



Broadcom NetXtreme Gigabit Ethernet Driver (Microsoft's Packet Scheduler) : Capturing - Wireshark
File Edit View

Filter: icmp or arp

Use ***icmp or arp*** *as a display filter to view only those packets*

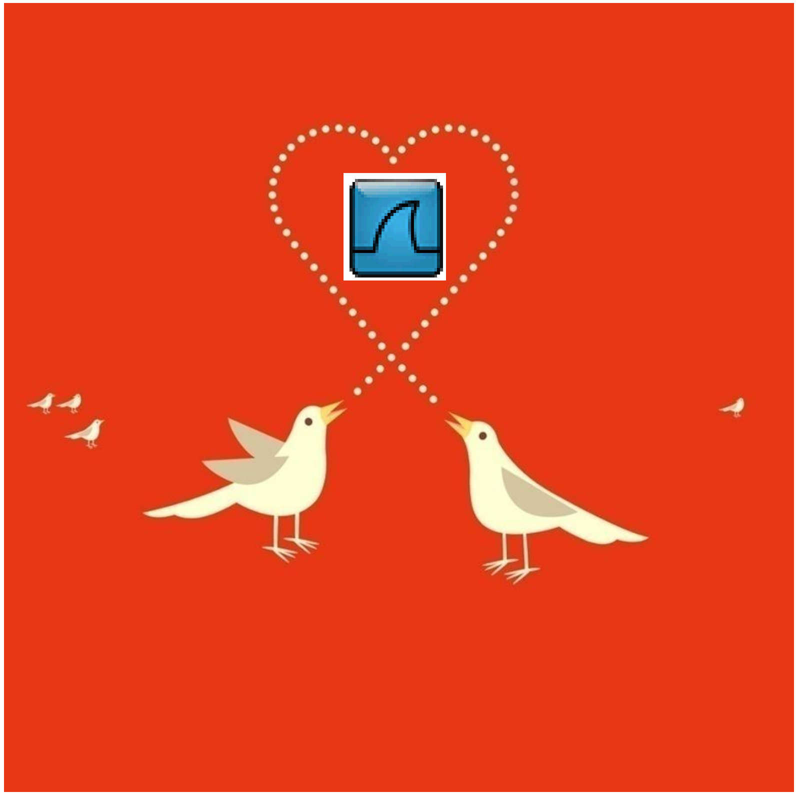
```
No.   Time        Source          Destination
110 152.291268  Dell_03:71:cc   Broadcast
129 178.560228  Dell_28:ac:50   Broadcast
134 185.545721  Dell_28:ac:50   Broadcast
143 197.399878  Dell_28:ac:50   Broadcast       ARP   Who has 172.30.1.1
147 199.778096  Dell_28:ac:50   Broadcast       ARP   Who has 172.30.1.1
173 220.386778  Dell_28:ac:50   Broadcast       ARP   Who has 172.30.1.1
177 223.945952  Dell_28:ac:50   Broadcast       ARP   Who has 172.30.1.1
184 230.797294  Dell_28:ac:50   Broadcast       ARP   Who has 172.30.1.1
186 230.820912  Dell_28:ac:50   Broadcast       ARP   Who has 172.30.1.1
187 230.820921  Dell_03:71:07   Dell_28:ac:50   ARP   172.30.1.101 is at
189 230.821249  172.30.1.101    172.30.1.121    ICMP  Echo (ping) reques
190 230.821361  172.30.1.121    172.30.1.101    ICMP  Echo (ping) reply
236 236.192151  Dell_28:ac:50   Broadcast       ARP   Who has 172.30.1.1
267 277.158895  Dell_03:71:cc   Broadcast       ARP   Who has 172.30.1.1
```

```
⊞ Frame 110 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: Dell_03:71:cc (00:19:b9:03:71:cc), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Address Resolution Protocol (request)
```

```
0000  ff ff ff ff ff ff 00 19  b9 03 71 cc 08 06 00 01   ........ ..q.....
0010  08 00 06 04 00 01 00 19  b9 03 71 cc ac 1e 01 67   ........ ..q....g
0020  00 00 00 00 00 00 ac 1e  01 79 00 00 00 00 00 00   ........ .y......
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

Broadcom NetXtreme Gigabit Ethernet Driver (Mi... | Packets: 274 Displayed: 14 Marked: 0 | Profile: Default

Viewing Network Packets with Wireshark

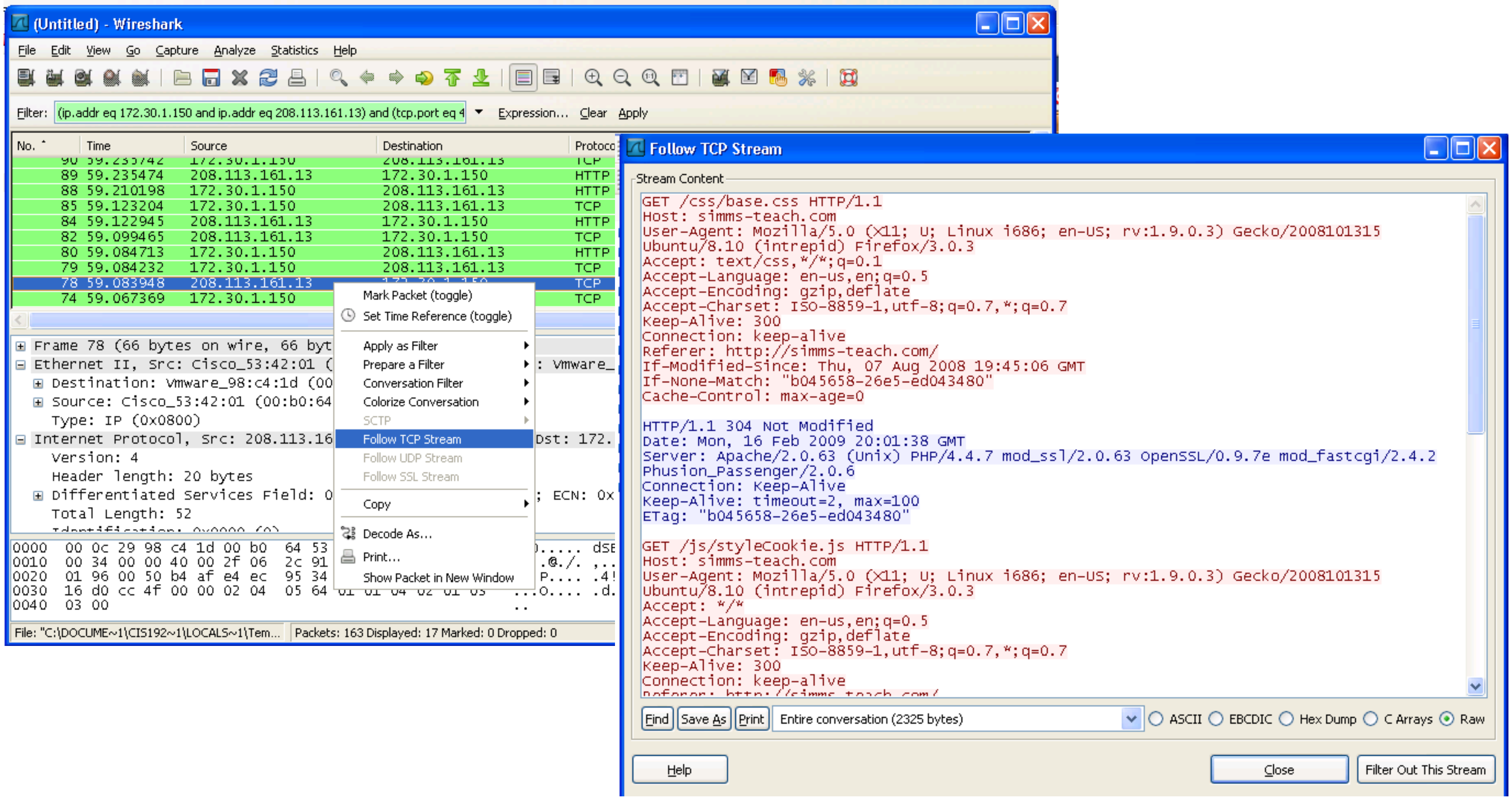


Some really nice options:

- Follow TCP stream
- Prepare a filter

Use icmp or arp as a display filter to view only those packets

Wireshark - Follow TCP Stream



Filter: (ip.addr eq 172.30.1.150 and ip.addr eq 208.113.161.13) and (tcp.port eq 4

```
No.   Time        Source           Destination      Protocol
 90 59.235742   172.30.1.150     208.113.161.13   TCP
 89 59.235474   208.113.161.13   172.30.1.150     HTTP
 88 59.210198   172.30.1.150     208.113.161.13   HTTP
 85 59.123204   172.30.1.150     208.113.161.13   TCP
 84 59.122945   208.113.161.13   172.30.1.150     HTTP
 82 59.099465   208.113.161.13   172.30.1.150     TCP
 80 59.084713   172.30.1.150     208.113.161.13   HTTP
 79 59.084232   172.30.1.150     208.113.161.13   TCP
 78 59.083948   208.113.161.13   172.30.1.150     TCP
 74 59.067369   172.30.1.150                      TCP
```

```
⊞ Frame 78 (66 bytes on wire, 66 byt
⊟ Ethernet II, Src: Cisco_53:42:01 (                : Vmware_
  ⊞ Destination: Vmware_98:c4:1d (00
  ⊞ Source: Cisco_53:42:01 (00:b0:64
    Type: IP (0x0800)
⊟ Internet Protocol, Src: 208.113.16                Dst: 172.
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0               ; ECN: 0x
    Total Length: 52
```

```
0000  00 0c 29 98 c4 1d 00 b0  64 53                  ).....  dSE
0010  00 34 00 00 40 00 2f 06  2c 91                  .@./.  ,..
0020  01 96 00 50 b4 af e4 ec  95 34                  P....  .4!
0030  16 d0 cc 4f 00 00 02 04  05 64 01 01 04 02 01 03  ...O....  .d.
0040  03 00                                            ..
```

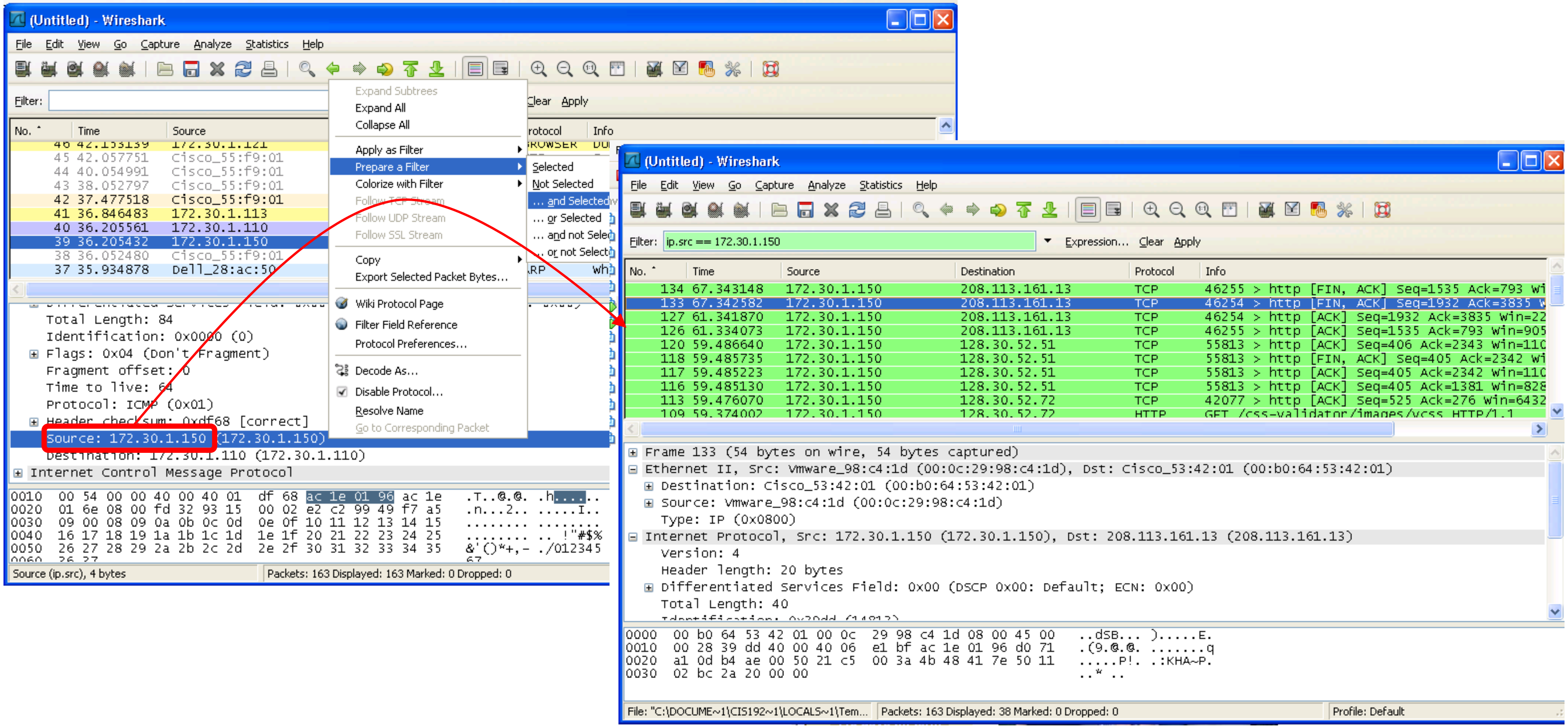
File: "C:\DOCUME~1\CIS192~1\LOCALS~1\Tem... Packets: 163 Displayed: 17 Marked: 0 Dropped: 0

```
GET /css/base.css HTTP/1.1
Host: simms-teach.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.3) Gecko/2008101315
Ubuntu/8.10 (intrepid) Firefox/3.0.3
Accept: text/css,*/*;q=0.1
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://simms-teach.com/
If-Modified-Since: Thu, 07 Aug 2008 19:45:06 GMT
If-None-Match: "b045658-26e5-ed043480"
Cache-Control: max-age=0

HTTP/1.1 304 Not Modified
Date: Mon, 16 Feb 2009 20:01:38 GMT
Server: Apache/2.0.63 (Unix) PHP/4.4.7 mod_ssl/2.0.63 OpenSSL/0.9.7e mod_fastcgi/2.4.2
Phusion_Passenger/2.0.6
Connection: Keep-Alive
Keep-Alive: timeout=2, max=100
ETag: "b045658-26e5-ed043480"

GET /js/styleCookie.js HTTP/1.1
Host: simms-teach.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.3) Gecko/2008101315
Ubuntu/8.10 (intrepid) Firefox/3.0.3
Accept: */*
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://simms-teach.com/
```

Entire conversation (2325 bytes)

Following the TCP stream of viewing a web page

Wireshark - Prepare a filter



```
No.   Time        Source
 45 42.057751   Cisco_55:f9:01
 44 40.054991   Cisco_55:f9:01
 43 38.052797   Cisco_55:f9:01
 42 37.477518   Cisco_55:f9:01
 41 36.846483   172.30.1.113
 40 36.205561   172.30.1.110
 39 36.205432   172.30.1.150
 38 36.052480   Cisco_55:f9:01
 37 35.934878   Dell_28:ac:50
```

```
    Total Length: 84
    Identification: 0x0000 (0)
  ⊞ Flags: 0x04 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (0x01)
  ⊞ Header checksum: 0xdf68 [correct]
    Source: 172.30.1.150 (172.30.1.150)
    Destination: 172.30.1.110 (172.30.1.110)
⊞ Internet Control Message Protocol
```

```
0010  00 54 00 00 40 00 40 01  df 68 ac 1e 01 96 ac 1e   .T..@.@. .h......
0020  01 6e 08 00 fd 32 93 15  00 02 e2 c2 99 49 f7 a5   .n...2.. .....I..
0030  09 00 08 09 0a 0b 0c 0d  0e 0f 10 11 12 13 14 15   ........ ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ .. !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
```

Source (ip.src), 4 bytes | Packets: 163 Displayed: 163 Marked: 0 Dropped: 0

Filter: ip.src == 172.30.1.150

```
No.   Time        Source          Destination      Protocol  Info
134 67.343148   172.30.1.150    208.113.161.13   TCP       46255 > http [FIN, ACK] Seq=1535 Ack=793 Wi
133 67.342582   172.30.1.150    208.113.161.13   TCP       46254 > http [FIN, ACK] Seq=1932 Ack=3835 W
127 61.341870   172.30.1.150    208.113.161.13   TCP       46254 > http [ACK] Seq=1932 Ack=3835 Win=22
126 61.334073   172.30.1.150    208.113.161.13   TCP       46255 > http [ACK] Seq=1535 Ack=793 Win=905
120 59.486640   172.30.1.150    128.30.52.51     TCP       55813 > http [ACK] Seq=406 Ack=2343 Win=110
118 59.485735   172.30.1.150    128.30.52.51     TCP       55813 > http [FIN, ACK] Seq=405 Ack=2342 Wi
117 59.485223   172.30.1.150    128.30.52.51     TCP       55813 > http [ACK] Seq=405 Ack=2342 Win=110
116 59.485130   172.30.1.150    128.30.52.51     TCP       55813 > http [ACK] Seq=405 Ack=1381 Win=828
113 59.476070   172.30.1.150    128.30.52.72     TCP       42077 > http [ACK] Seq=525 Ack=276 Win=6432
109 59.374002   172.30.1.150    128.30.52.72     HTTP      GET /css-validator/images/vcss HTTP/1.1
```

```
⊞ Frame 133 (54 bytes on wire, 54 bytes captured)
⊟ Ethernet II, Src: Vmware_98:c4:1d (00:0c:29:98:c4:1d), Dst: Cisco_53:42:01 (00:b0:64:53:42:01)
  ⊞ Destination: Cisco_53:42:01 (00:b0:64:53:42:01)
  ⊞ Source: Vmware_98:c4:1d (00:0c:29:98:c4:1d)
    Type: IP (0x0800)
⊟ Internet Protocol, Src: 172.30.1.150 (172.30.1.150), Dst: 208.113.161.13 (208.113.161.13)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 40
```

```
0000  00 b0 64 53 42 01 00 0c  29 98 c4 1d 08 00 45 00   ..dSB... ).....E.
0010  00 28 39 dd 40 00 40 06  e1 bf ac 1e 01 96 d0 71   .(9.@.@. .......q
0020  a1 0d b4 ae 00 50 21 c5  00 3a 4b 48 41 7e 50 11   .....P!. .:KHA~P.
0030  02 bc 2a 20 00 00                                  ..* ..
```

File: "C:\DOCUME~1\CIS192~1\LOCALS~1\Tem... | Packets: 163 Displayed: 38 Marked: 0 Dropped: 0 | Profile: Default

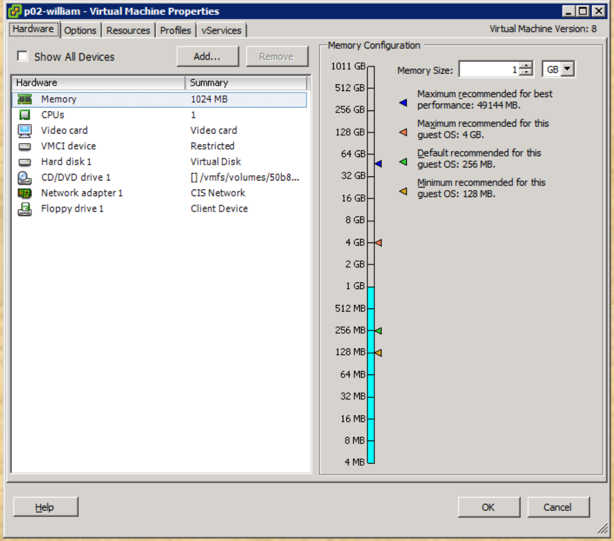
Select the source IP address of a packet and use it to make a display filter to only see packets from that IP address

Wireshark - example filters

- arp *will only show ARP packets*
- arp || icmp *will only show ARP and ICMP packets*
- http *will only show HTTP packets*
- bootp *will only show bootp and DHCP packets*
- (ip.src == 172.30.1.107 || ip.dst == 172.30.1.107) *will only show packets going to or from 172.30.1.107*
- icmp && (ip.src == 172.30.1.107 || ip.dst == 172.30.1.107) *will only show ARP packets going to or from 172.30.1.107*
- !ssh *will hide any SSH packets*
- ip.src == 172.30.1.0/24 *will only show packets with a source IP address in the 172.30.1.0/24 subnet*
- ip.host == 172.30.1.125

Class Activity

Wireshark



p02-william - Virtual Machine Properties

Virtual Machine Version: 8

□ Show All Devices

Hardware	Summary
Memory	1024 MB
CPUs	1
Video card	Video card
VMCI device	Restricted
Hard disk 1	Virtual Disk
CD/DVD drive 1	[] /vmfs/volumes/50b8...
Network adapter 1	CIS Network
Floppy drive 1	Client Device

Memory Configuration

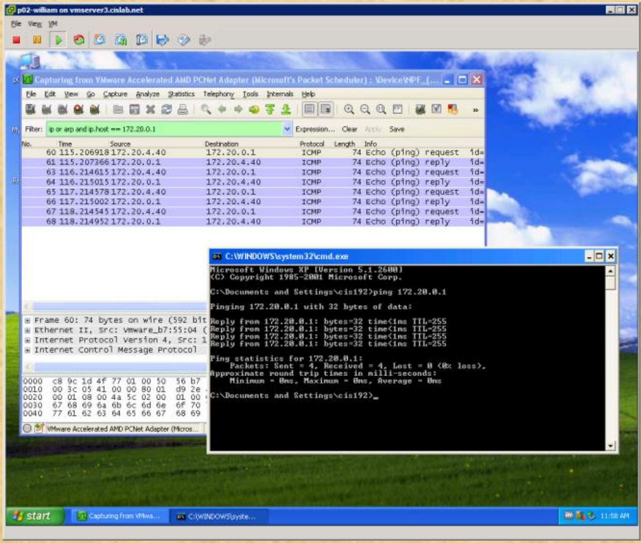
Memory Size: 1 GB

Maximum recommended for best performance: 49144 MB.

Maximum recommended for this guest OS: 4 GB.

Default recommended for this guest OS: 256 MB.

Minimum recommended for this guest OS: 128 MB.



p02-william on vmserver3.cislab.net

Capturing from VMware Accelerated AMD PCNet Adapter (Microsoft's Packet Scheduler) : \Device\NPF_{...

Filter: ip or arp and ip.host == 172.20.0.1

```
No.   Time       Source        Destination   Protocol Length Info
   60 115.206918 172.20.4.40   172.20.0.1    ICMP         74 Echo (ping) request  id=
   61 115.207366 172.20.0.1    172.20.4.40   ICMP         74 Echo (ping) reply    id=
   63 116.214615 172.20.4.40   172.20.0.1    ICMP         74 Echo (ping) request  id=
   64 116.215015 172.20.0.1    172.20.4.40   ICMP         74 Echo (ping) reply    id=
   65 117.214578 172.20.4.40   172.20.0.1    ICMP         74 Echo (ping) request  id=
   66 117.215002 172.20.0.1    172.20.4.40   ICMP         74 Echo (ping) reply    id=
   67 118.214545 172.20.4.40   172.20.0.1    ICMP         74 Echo (ping) request  id=
   68 118.214952 172.20.0.1    172.20.4.40   ICMP         74 Echo (ping) reply    id=
```

```
⊞ Frame 60: 74 bytes on wire (592 bit
⊞ Ethernet II, Src: Vmware_b7:55:04 (
⊞ Internet Protocol Version 4, Src: 1
⊞ Internet Control Message Protocol

0000  c8 9c 1d 4f 77 01 00 50  56 b7
0010  00 3c 05 41 00 00 80 01  d9 2e
0020  00 01 08 00 4a 5c 02 00  01 00
0030  67 68 69 6a 6b 6c 6d 6e  6f 70
0040  77 61 62 63 64 65 66 67  68 69
```

VMware Accelerated AMD PCNet Adapter (Micros...

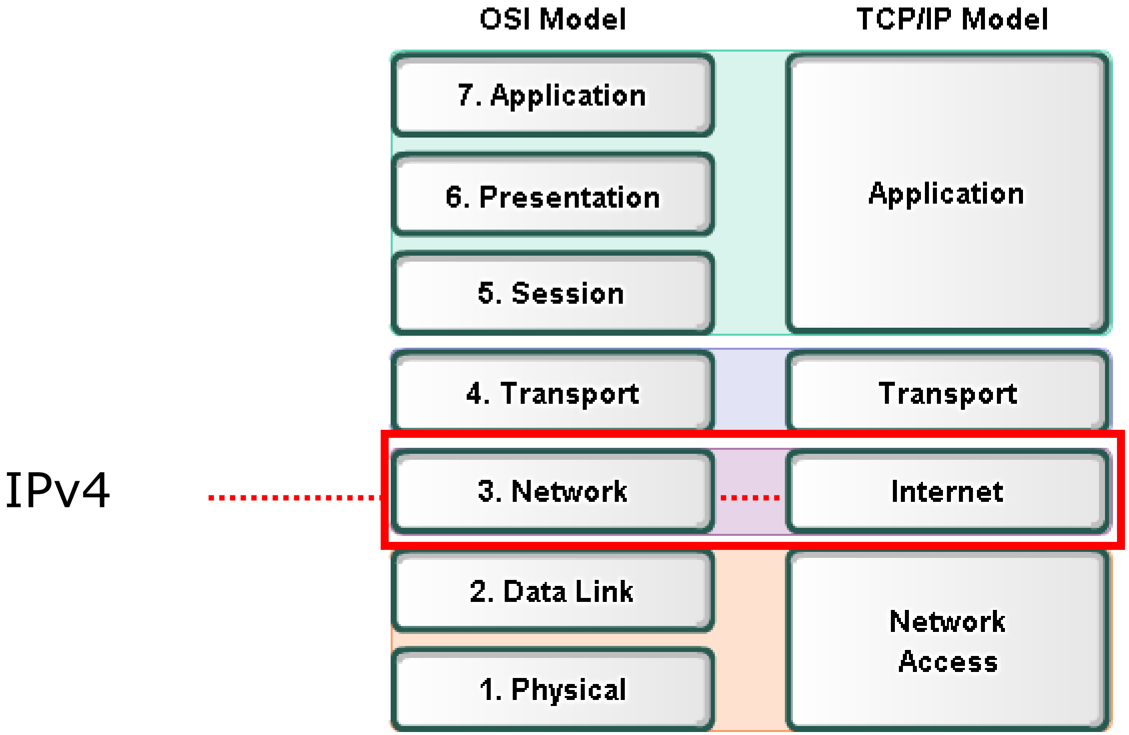
C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\cis192>ping 172.20.0.1

Pinging 172.20.0.1 with 32 bytes of data:

Reply from 172.20.0.1: bytes=32 time<1ms TTL=255
Reply from 172.20.0.1: bytes=32 time<1ms TTL=255
Reply from 172.20.0.1: bytes=32 time<1ms TTL=255
Reply from 172.20.0.1: bytes=32 time<1ms TTL=255

Ping statistics for 172.20.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\cis192>_
```

- Edits settings on Wiliam VM and boost RAM to 1GB
- [William] Run Wireshark
- [William] "ip or arp" filter
- [William] "ip or arp and ip.host == 172.20.0.1" filter
- [William] ping 172.20.0.1

Layer 3

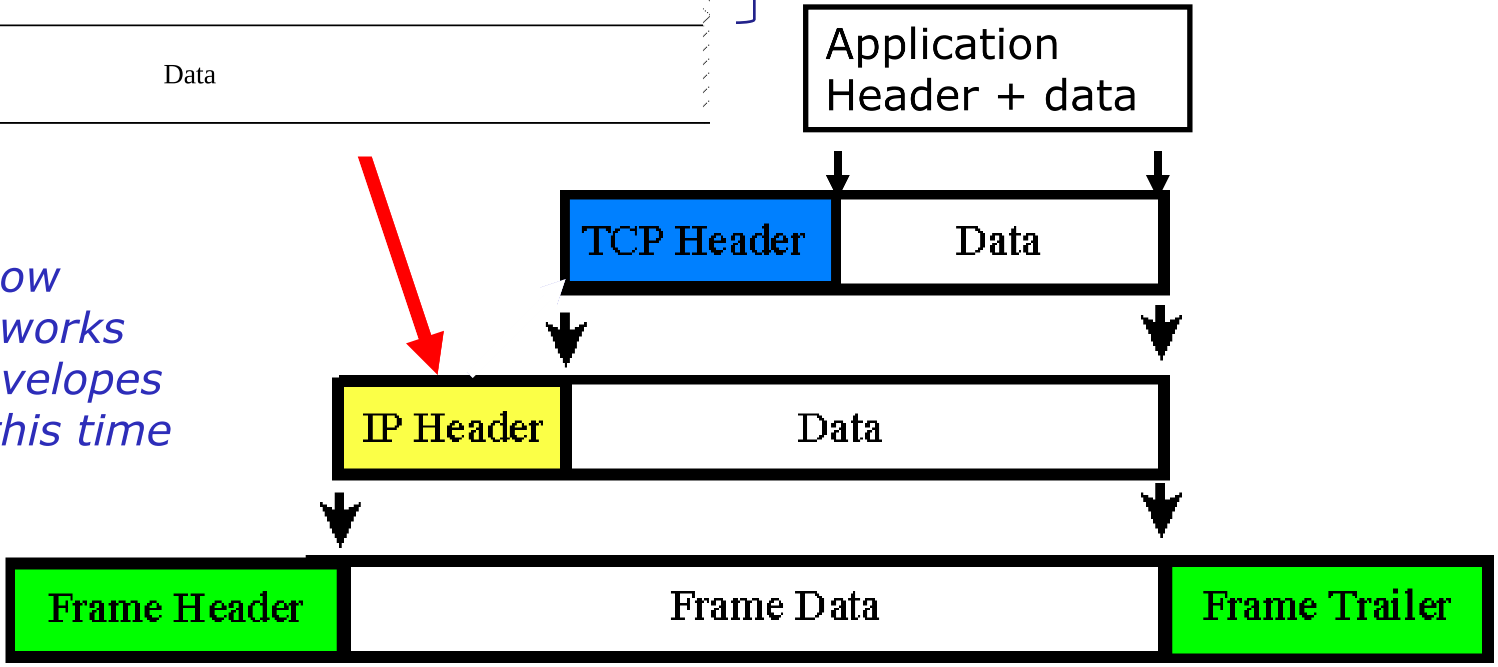
Network Layer



RS: More on Layer 3 tonight

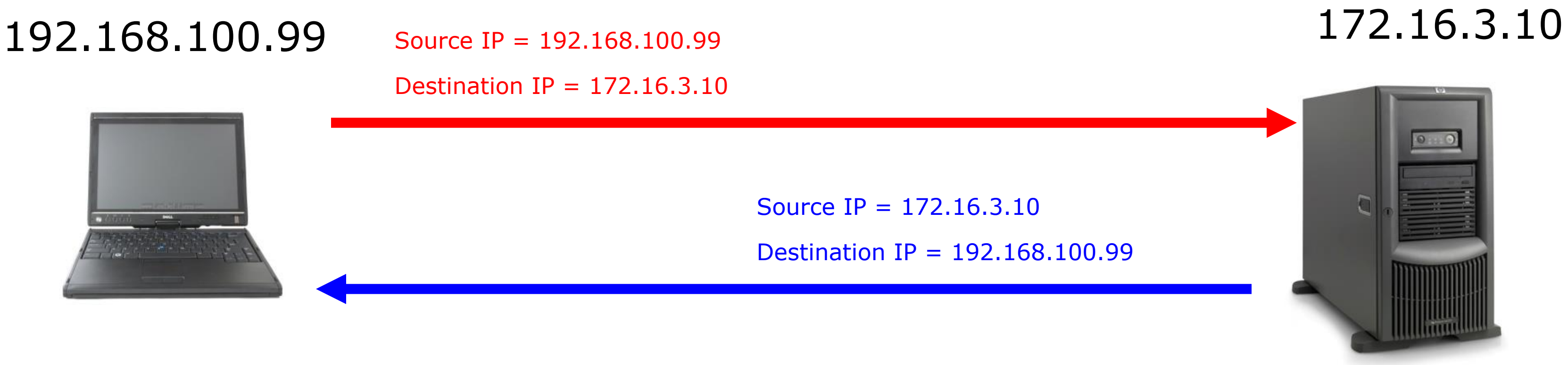
0		15	16	31
4-bit Version	4-bit Header Length	8-bit Type Of Service (TOS)	16-bit Total Length (in bytes)	
16-bit Identification			3-bit Flags	13-bit Fragment Offset
8 bit Time To Live TTL		8-bit Protocol	16-bit Header Checksum	
32-bit Source IP Address				
32-bit Destination IP Address				
Options (if any)				
Data				

IP Header



RS: showing how encapsulation works without the envelopes and postman this time

Addressing



192.168.100.99

172.16.3.10

Source IP = 192.168.100.99

Destination IP = 172.16.3.10

Source IP = 172.16.3.10

Destination IP = 192.168.100.99

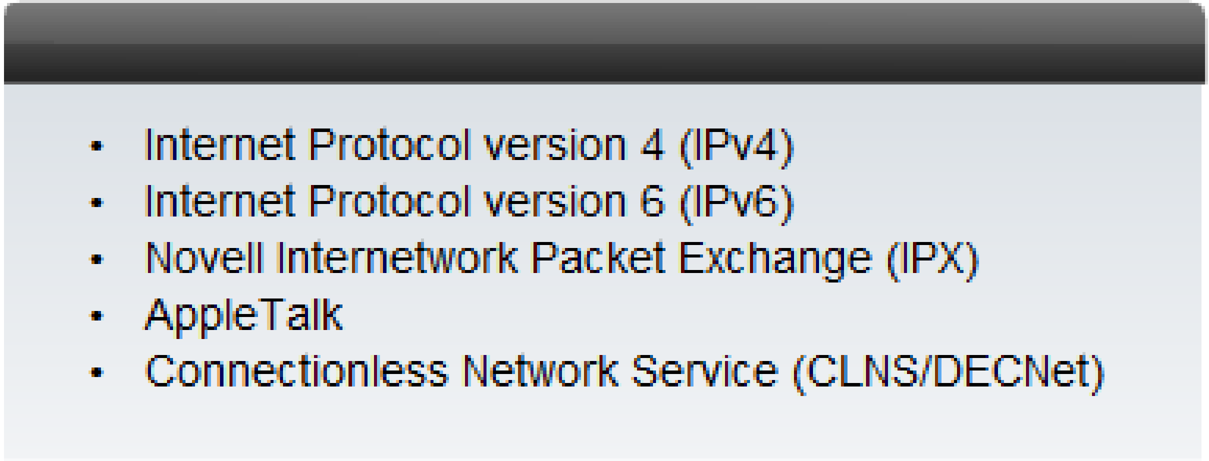
- Source IP Address
- Destination IP Address

- More later!

RS: Layer 3 is where IP addresses are used. They are put in the header of the layer three packets.

<table>
<tr><td colspan="3">0 15</td><td colspan="2">16 31</td></tr>
<tr><td>4-bit Version</td><td>4-bit Header Length</td><td>8-bit Type Of Service (TOS)</td><td colspan="2">16-bit Total Length (in bytes)</td></tr>
<tr><td colspan="3">16-bit Identification</td><td>3-bit Flags</td><td>13-bit Fragment Offset</td></tr>
<tr><td colspan="2">8 bit Time To Live TTL</td><td>8-bit Protocol</td><td colspan="2">16-bit Header Checksum</td></tr>
<tr><td colspan="5">32-bit Source IP Address</td></tr>
<tr><td colspan="5">32-bit Destination IP Address</td></tr>
<tr><td colspan="5">Options (if any)</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>

Network Layer Protocols

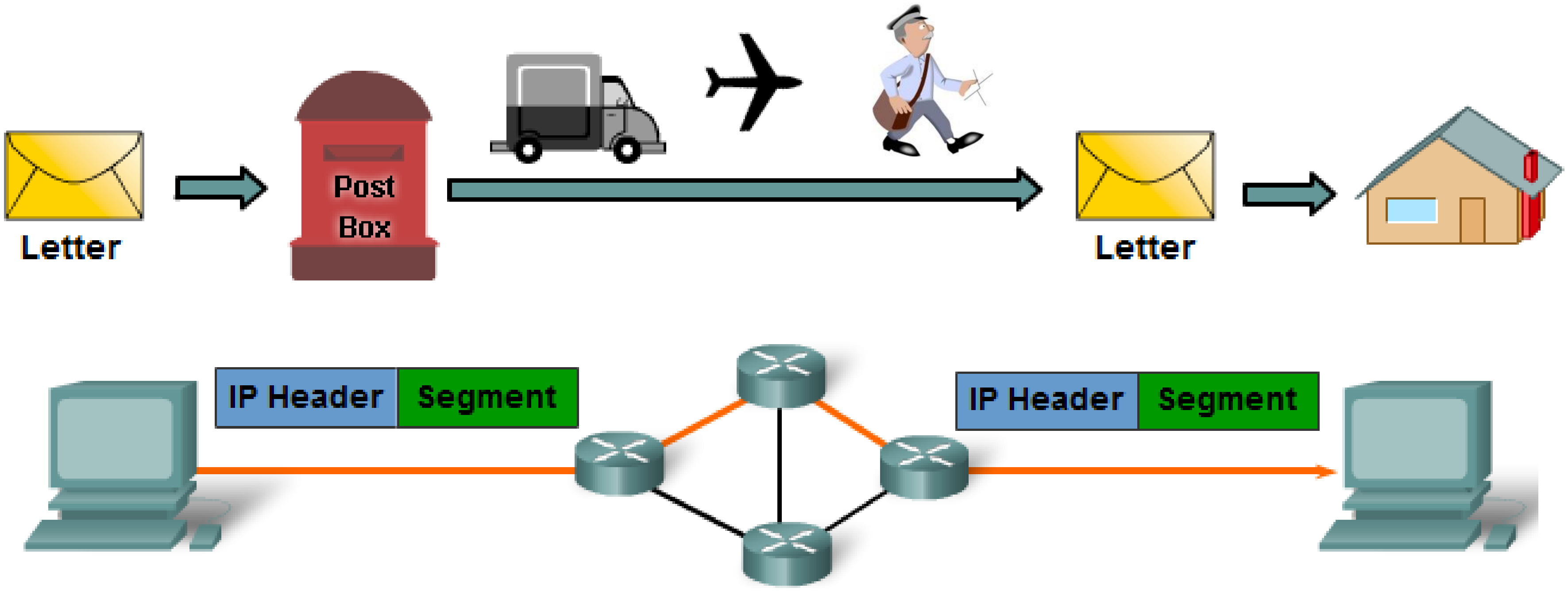


- Internet Protocol version 4 (IPv4)
- Internet Protocol version 6 (IPv6)
- Novell Internetwork Packet Exchange (IPX)
- AppleTalk
- Connectionless Network Service (CLNS/DECNet)

- The Internet Protocol (IPv4 and IPv6) is the most widely-used Layer 3 data carrying protocol and will be the focus of this course.

same goes for CIS 192!

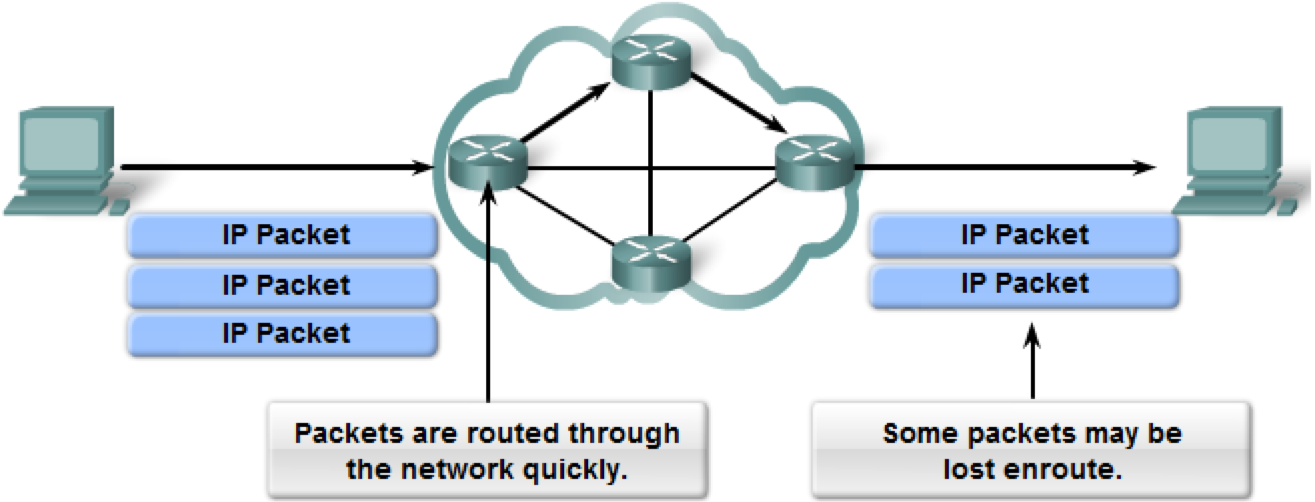
Connectionless



IP packets are sent without notifying the end host that they are coming. *(Layer 3)*

- **TCP**: A connection-oriented protocol does requires a connection to be established prior to sending TCP segments. *(Layer 4)*
- **UDP**: A connectionless protocol does not require a session to be established. *(Layer 4)*

Best Effort Service (unreliable)

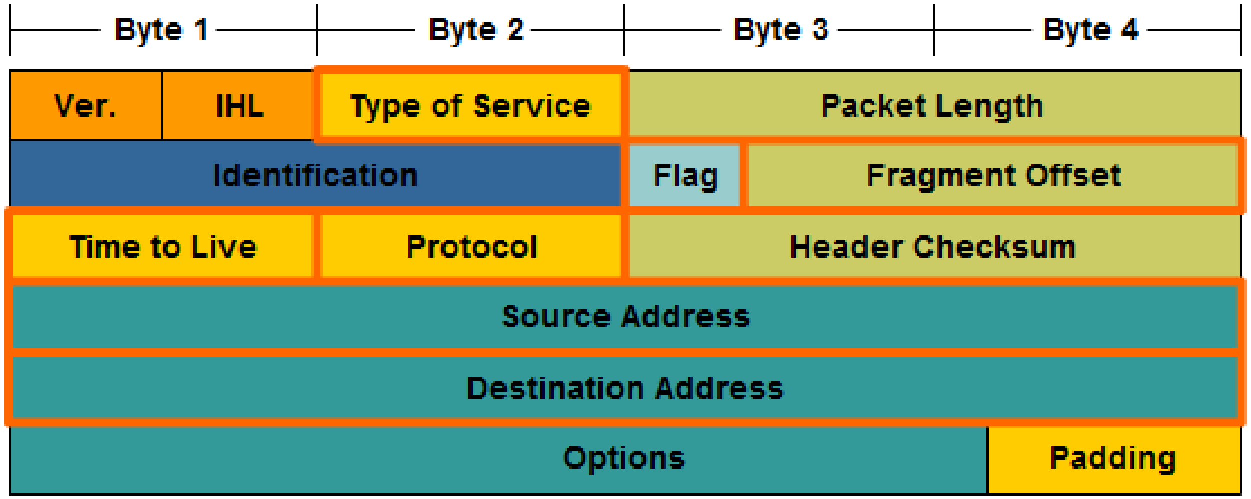


Packets are routed through the network quickly.

Some packets may be lost enroute.

- The mission of Layer 3 is to transport the packets between the hosts while placing as little burden on the network as possible.
  - Speed over reliability
- Layer 3 is not concerned with or even aware of the type of data contained inside of a packet.
  - This responsibility is the role of the upper layers as required.
- **Unreliable**: IP does not have the capability or responsibility to manage, and recover from, undelivered or corrupt packets.
  - TCP's responsibility at the end-to-end hosts

IP Header



Byte 1 — Byte 2 — Byte 3 — Byte 4

Ver. | IHL | Type of Service | Packet Length

Identification | Flag | Fragment Offset

Time to Live | Protocol | Header Checksum

Source Address

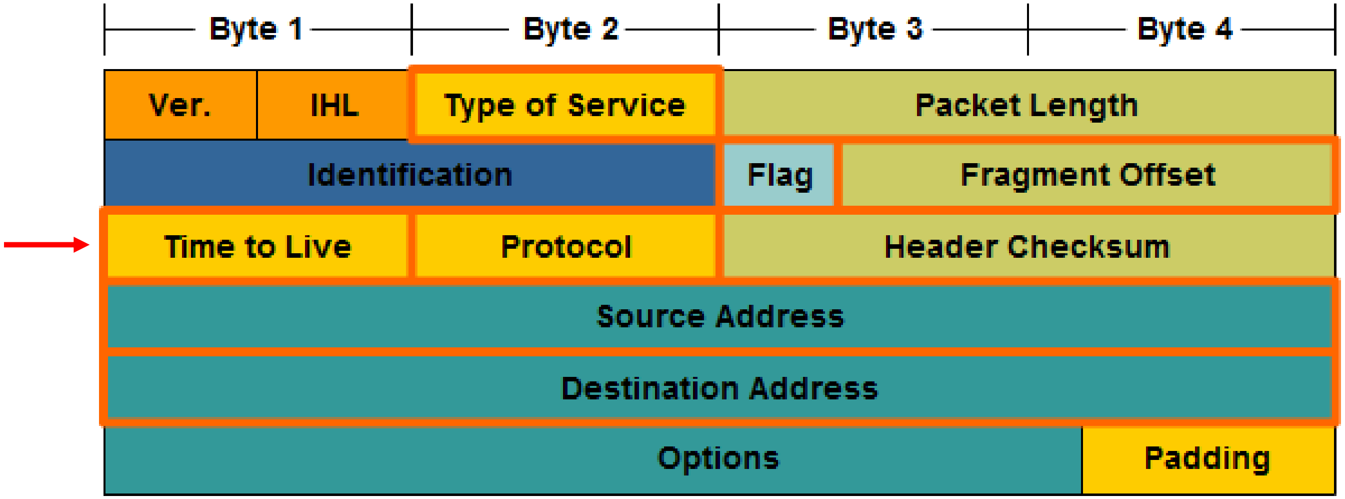
Destination Address

Options | Padding

- **IP Destination Address**
  - 32-bit binary value that represents the packet destination Network layer host address.
- **IP Source Address**
  - 32-bit binary value that represents the packet source Network layer host address.

IP’s TTL - Time To Live field

Decrement by 1, if 0 drop the packet.



Byte 1 | Byte 2 | Byte 3 | Byte 4

Ver. | IHL | Type of Service | Packet Length

Identification | Flag | Fragment Offset

Time to Live | Protocol | Header Checksum

Source Address

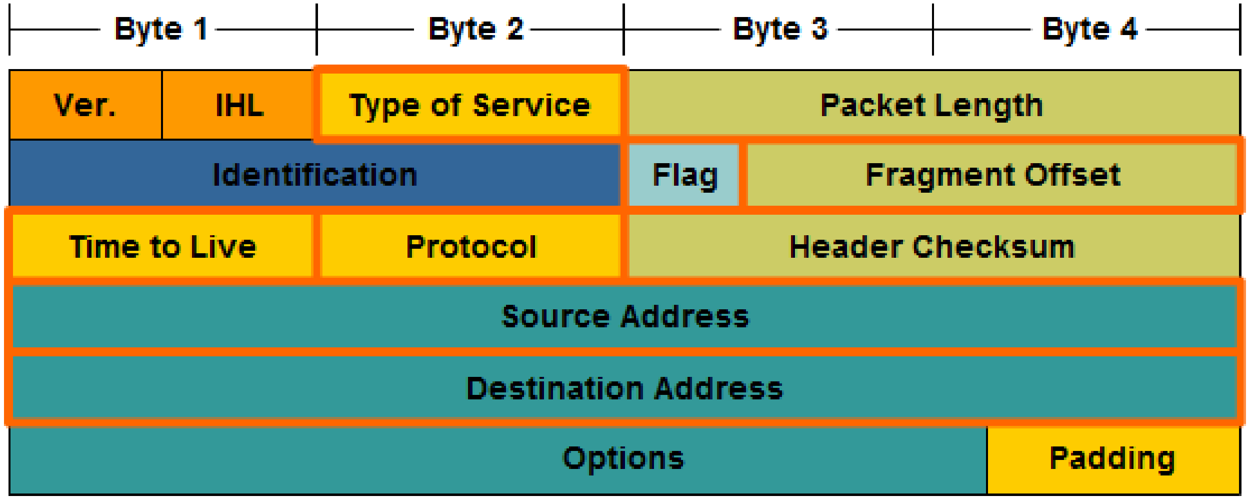
Destination Address

Options | Padding

- If the router decrements the TTL field to 0, it will then drop the packet (unless the packet is destined specifically for the router, i.e. ping, telnet, etc.).
- Common operating system TTL values are:
  - UNIX: **255**
  - Linux: **64 or 255** depending upon vendor and version
  - Microsoft Windows 95: **32**
  - Other Microsoft Windows operating systems: **128**

IP’s TTL - Time To Live field

Decrement by 1, if 0 drop the packet.



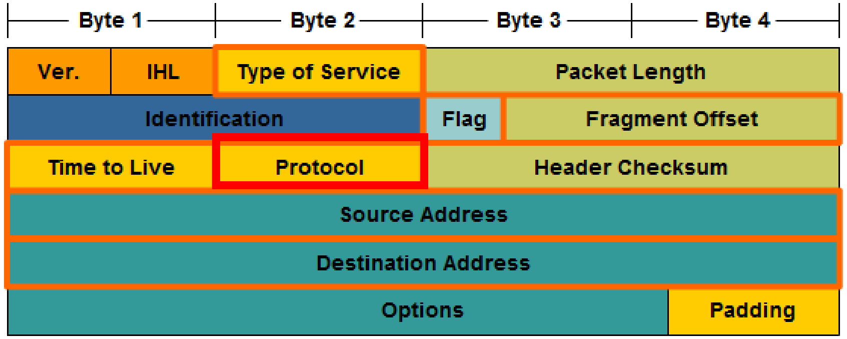
Byte 1		Byte 2	Byte 3	Byte 4
Ver.	IHL	Type of Service	Packet Length	
Identification			Flag	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source Address				
Destination Address				
Options				Padding

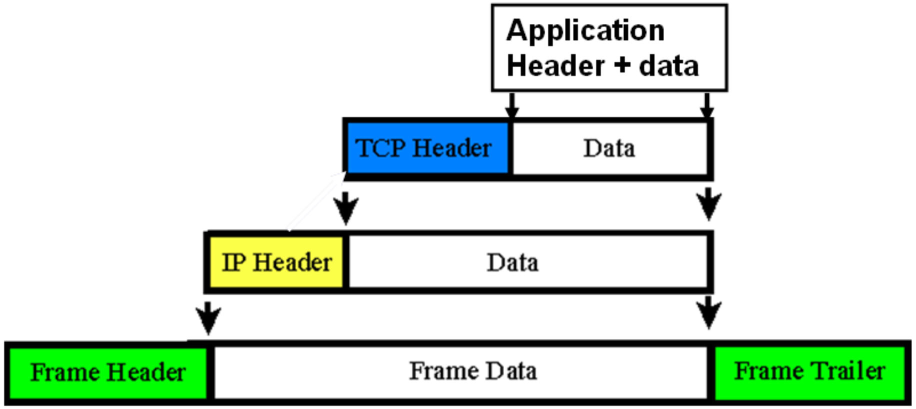
- The idea behind the TTL field is that IP packets can not travel around the Internet forever, from router to router.

- Eventually, the packet’s TTL which reach 0 and be dropped by the router, even if there is a routing loop somewhere in the network.

RS: TTL errors are used by traceroute and mtr to discover the path a packet takes

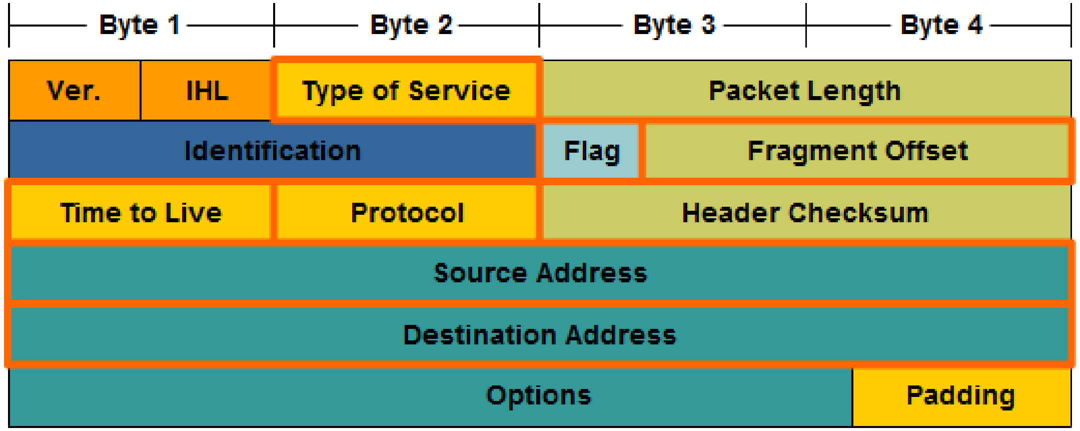
IP’s Protocol Field





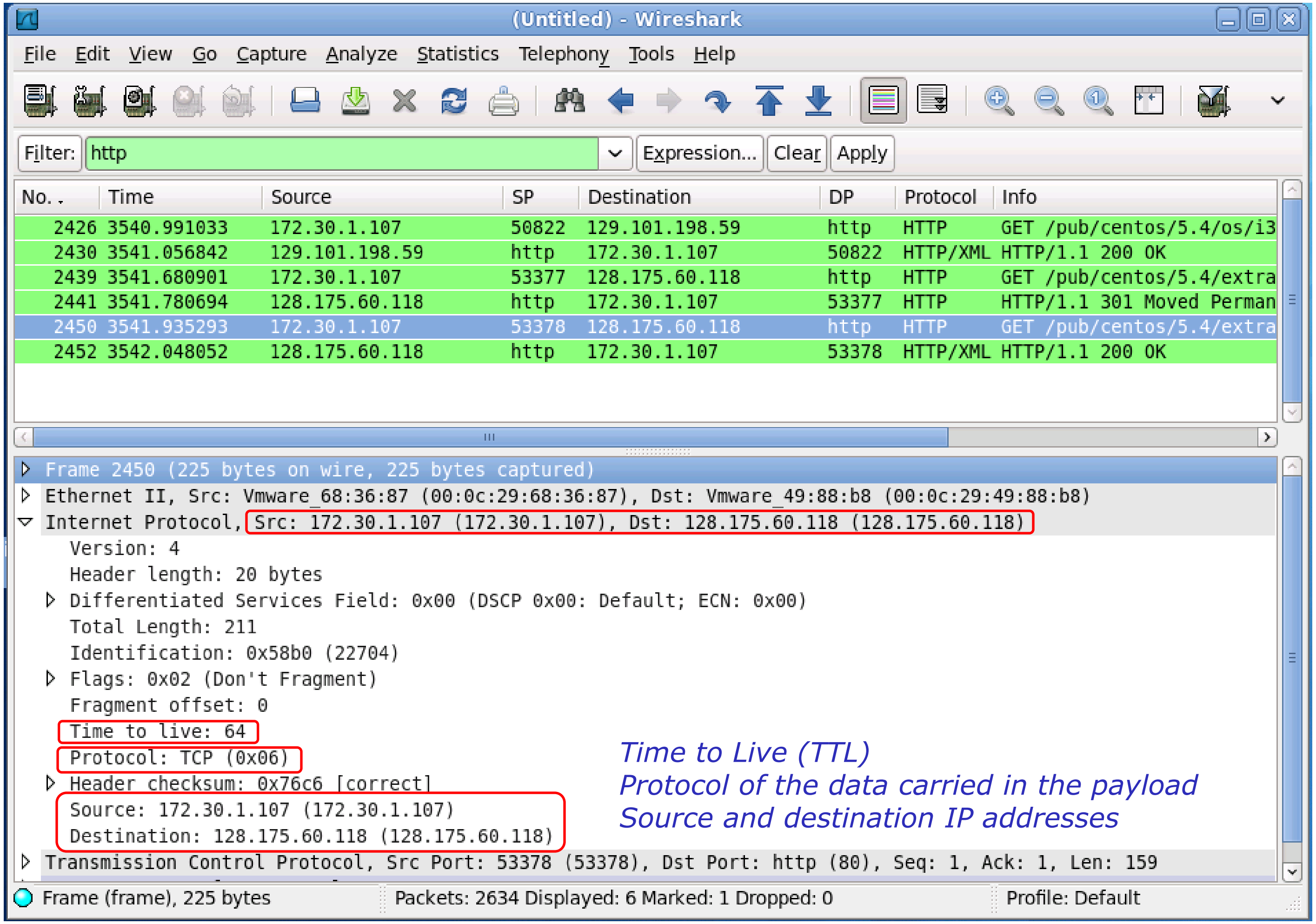
- **Protocol field** enables the Network layer to pass the data to the appropriate upper-layer protocol.
- Example values are:
  - 01 ICMP
  - 06 TCP
  - 17 UDP

Other IPv4 fields



- **Version** - Contains the IP version number (4)
- **Header Length (IHL)** - Specifies the size of the packet header.
- Packet Length - This field gives the entire packet size, including header and data, in bytes.
- **Identification** - This field is primarily used for uniquely identifying fragments of an original IP packet
- **Header Checksum** - The checksum field is used for error checking the packet header.
- **Options** - There is provision for additional fields in the IPv4 header to provide other services but these are rarely used.

Viewing Layer 3 IP Packets with Wireshark


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Expression... Clear Apply

Filter: http

```
No. .  Time         Source          SP     Destination     DP     Protocol Info
  2426 3540.991033  172.30.1.107    50822  129.101.198.59  http   HTTP     GET /pub/centos/5.4/os/i3
  2430 3541.056842  129.101.198.59  http   172.30.1.107    50822  HTTP/XML HTTP/1.1 200 OK
  2439 3541.680901  172.30.1.107    53377  128.175.60.118  http   HTTP     GET /pub/centos/5.4/extra
  2441 3541.780694  128.175.60.118  http   172.30.1.107    53377  HTTP     HTTP/1.1 301 Moved Perman
  2450 3541.935293  172.30.1.107    53378  128.175.60.118  http   HTTP     GET /pub/centos/5.4/extra
  2452 3542.048052  128.175.60.118  http   172.30.1.107    53378  HTTP/XML HTTP/1.1 200 OK
```

```
▷ Frame 2450 (225 bytes on wire, 225 bytes captured)
▷ Ethernet II, Src: Vmware_68:36:87 (00:0c:29:68:36:87), Dst: Vmware_49:88:b8 (00:0c:29:49:88:b8)
▽ Internet Protocol, Src: 172.30.1.107 (172.30.1.107), Dst: 128.175.60.118 (128.175.60.118)
    Version: 4
    Header length: 20 bytes
  ▷ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 211
    Identification: 0x58b0 (22704)
  ▷ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
  ▷ Header checksum: 0x76c6 [correct]
    Source: 172.30.1.107 (172.30.1.107)
    Destination: 128.175.60.118 (128.175.60.118)
▷ Transmission Control Protocol, Src Port: 53378 (53378), Dst Port: http (80), Seq: 1, Ack: 1, Len: 159
```

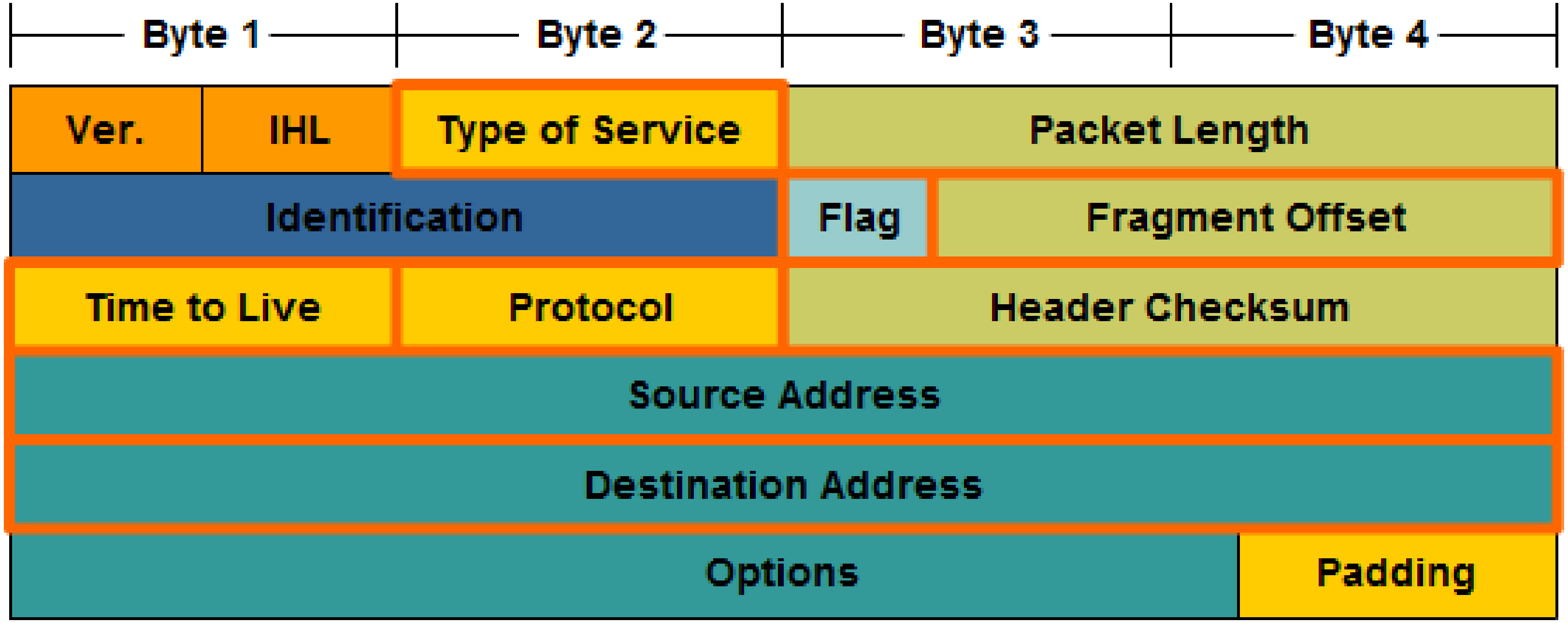
Frame (frame), 225 bytes | Packets: 2634 Displayed: 6 Marked: 1 Dropped: 0 | Profile: Default

Time to Live (TTL)
Protocol of the data carried in the payload
Source and destination IP addresses

Frodo is browsing google.com

IPv4 addressing & subnetting

IPv4 Addresses



Byte 1 | Byte 2 | Byte 3 | Byte 4

Ver. | IHL | Type of Service | Packet Length

Identification | Flag | Fragment Offset

Time to Live | Protocol | Header Checksum

Source Address

Destination Address

Options | Padding

- IPv4 addresses are 32 bit addresses

IPv4 Addresses

- IPv4 Addresses are 32 bit addresses:

1010100111000111010001011000100

10101001 11000111 01000101 10001001

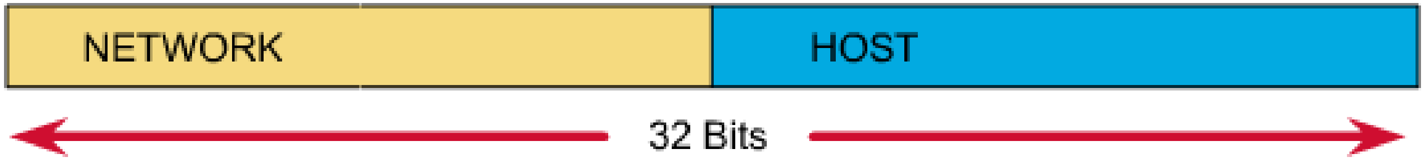
- We use dotted notation (or dotted decimal notation) to represent the value of each byte (octet) of the IP address in decimal.

```
10101001  11000111  01000101  10001001
   169    .    199    .    69    .    137
```

IPv4 Addresses

An IP address has two parts:

- **network number**
- **host number**



Which bits refer to the network number?

Which bits refer to the host number?

IPv4 Addresses

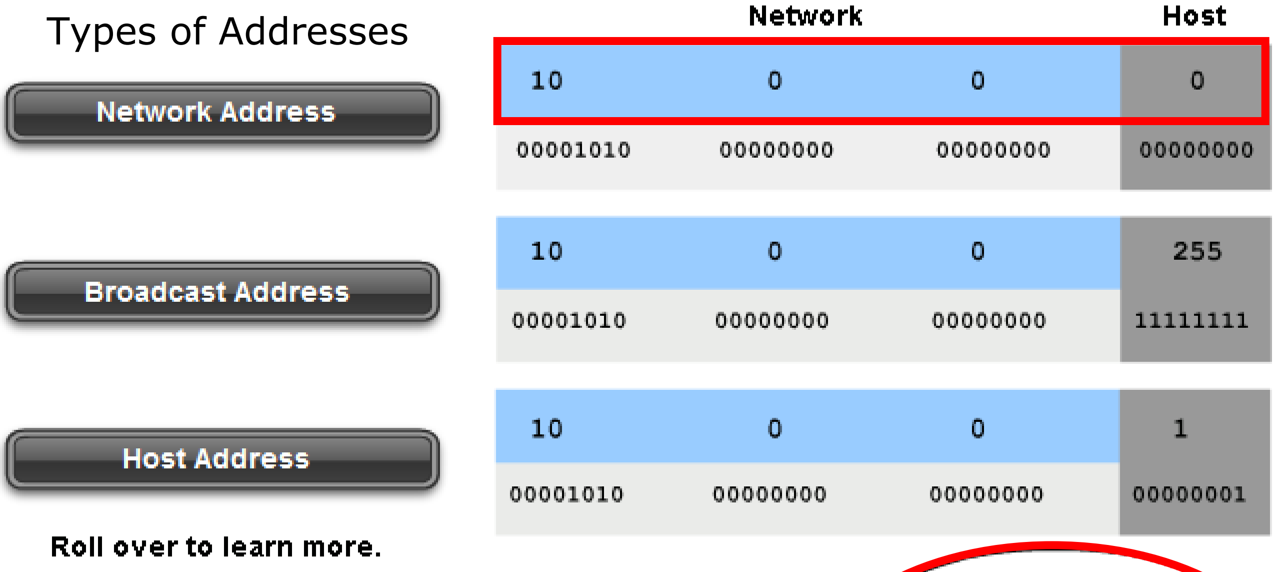
Answer:

- Newer technology - **Classless IP Addressing**
  - The **subnet mask** determines the network portion and the host portion.
  - Value of first octet does NOT matter (older classful IP addressing)
  - Hosts and Classless Inter-Domain Routing (**CIDR**).
  - Classless IP Addressing is what is used within the Internet and in most internal networks.

- Older technology - **Classful IP Addressing**
  - **Value of first octet** determines the network portion and the host portion.
  - Used with classful routing protocols like RIPv1.
  - The Cisco IP Routing Table is structured in a classful manner (CIS 82)

RS: We will be using Classless IP Addressing in CIS 192 which means we will always be specifying network masks on interfaces and genmasks in routing tables

Network Addresses have all 0’s in the host portion.

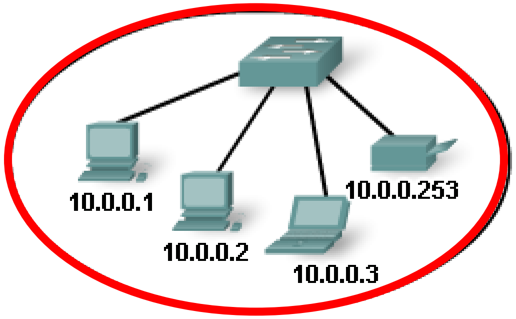


Types of Addresses

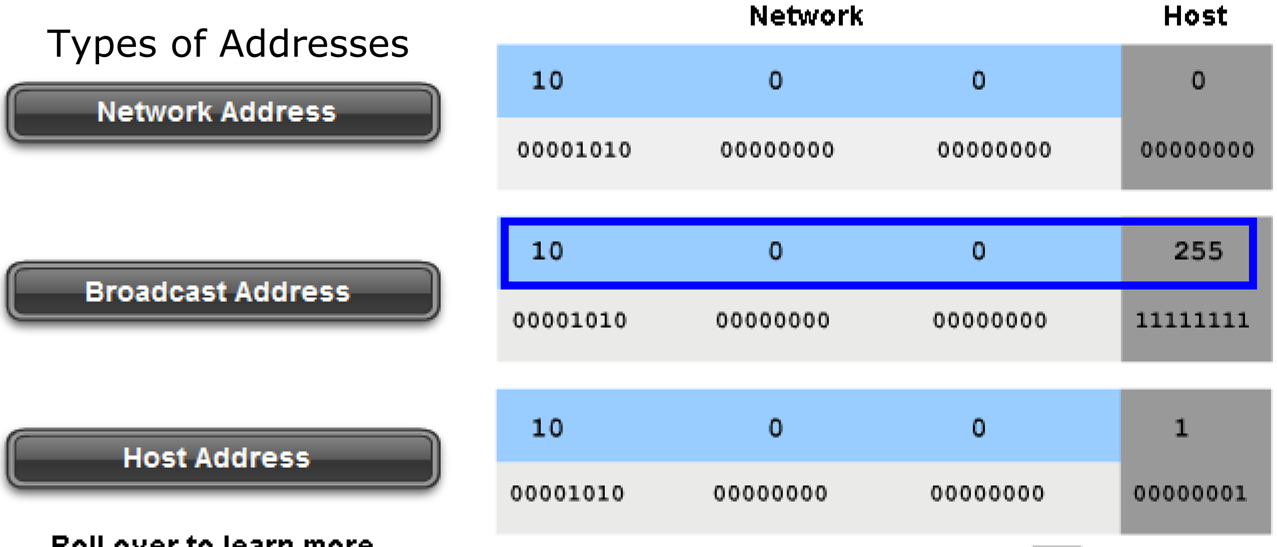
Network			Host
10	0	0	0
00001010	00000000	00000000	00000000
10	0	0	255
00001010	00000000	00000000	11111111
10	0	0	1
00001010	00000000	00000000	00000001

Roll over to learn more.

Subnet Mask: 255.255.255.0



- **Network address** - The address by which we refer to the network
- **Broadcast address** - A special address used to send data to all hosts in the network
- **Host addresses** - The addresses assigned to the end devices in the network



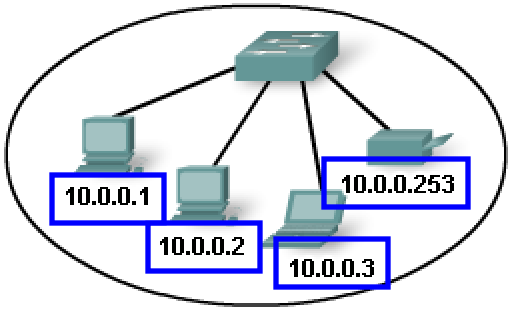
Types of Addresses

	Network			Host
Network Address	10	0	0	0
	00001010	00000000	00000000	00000000
Broadcast Address	10	0	0	255
	00001010	00000000	00000000	11111111
Host Address	10	0	0	1
	00001010	00000000	00000000	00000001

Roll over to learn more.

Broadcast Addresses have all 1's in the host portion.

Subnet Mask: 255.255.255.0



- **Network address** - The address by which we refer to the network
- **Broadcast address** - A special address used to send data to all hosts in the network
- **Host addresses** - The addresses assigned to the end devices in the network

Types of Addresses

Network Address

Broadcast Address

Host Address

Roll over to learn more.

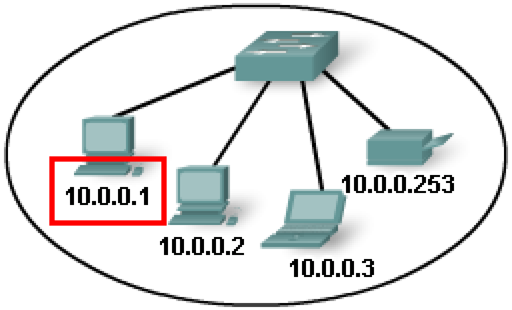
Network			Host
10	0	0	0
00001010	00000000	00000000	00000000

Network			Host
10	0	0	255
00001010	00000000	00000000	11111111

Network			Host
10	0	0	1
00001010	00000000	00000000	00000001

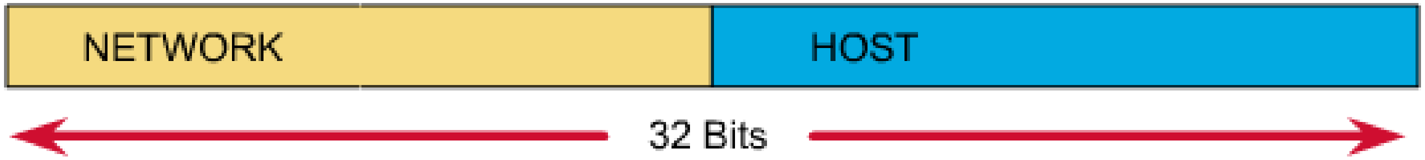
Host Addresses can <u>not</u> have all 0's or all 1's in the host portion.

Subnet Mask: 255.255.255.0



- **Network address** - The address by which we refer to the network
- **Broadcast address** - A special address used to send data to all hosts in the network
- **Host addresses** - The addresses assigned to the end devices in the network

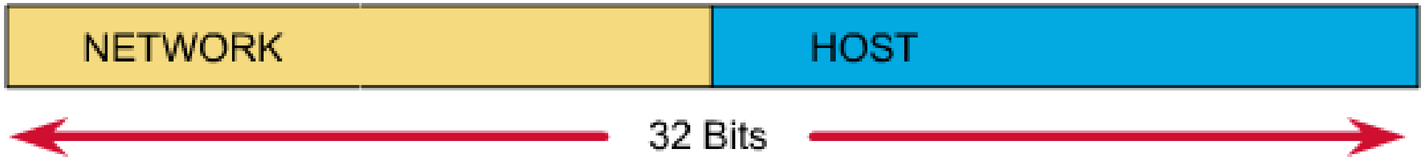
Dividing the Network and Host Portions



11111111111111110000000000000000

- **Subnet Mask**
  - Used to define the:
    - Network portion
    - Host portion
  - 32 bits
  - Contiguous set of 1’s followed by a contiguous set of 0’s
    - 1’s: Network portion
    - 0’s: Host portion

Dividing the Network and Host Portions



11111111.11111111.00000000.00000000

Dotted decimal: 255 . 255 . 0 . 0

Slash notation: /16

- Subnet mask expressed as:
  - Dotted decimal
    - Ex: 255.255.0.0
  - Slash notation or prefix length
    - /16 (the number of one bits)

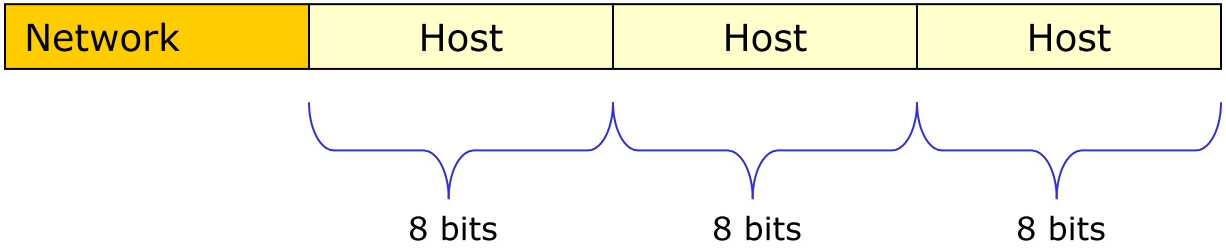
RS: We will use both dotted and slash notations in CIS 192

Why the mask matters: Number of hosts!

Subnet Mask:	1st octet	2nd octet	3rd octet	4th octet
255.0.0.0 or /8	Network	Host	Host	Host
255.255.0.0 or /16	Network	Network	Host	Host
255.255.255.0 or /24	Network	Network	Network	Host

- The more host bits in the subnet mask means the more hosts in the network.
- Subnet masks do not have to end on "natural octet boundaries"

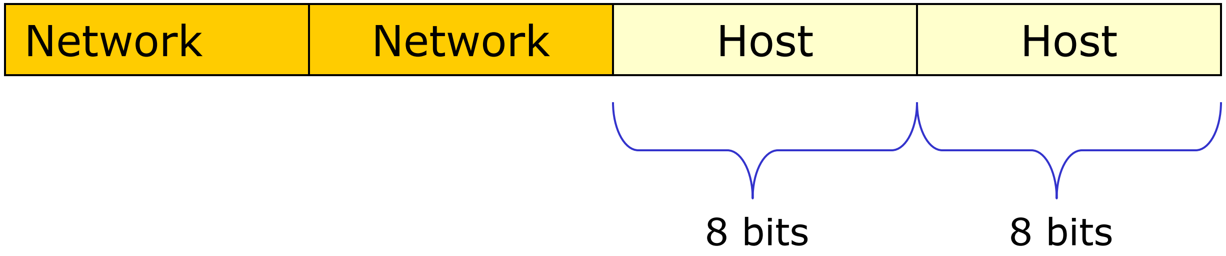
Subnet: 255.0.0.0 (/8)



With 24 bits available for hosts, there a 2^{24} possible addresses. That’s 16,777,216 nodes!

- Only large organizations such as the military, government agencies, universities, and large corporations have networks with these many addresses.
- Example: A certain cable modem ISP has 24.0.0.0 and a DSL ISP has 63.0.0.0

Subnet: 255.255.0.0 (/16)

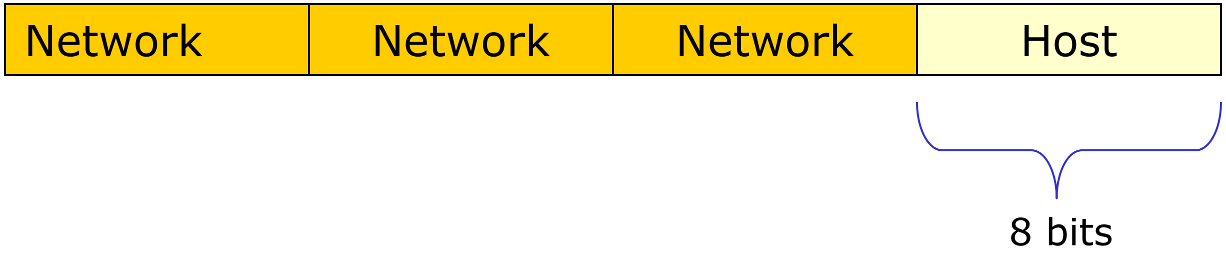


With 16 bits available for hosts, there a 2^{16} possible addresses. That’s 65,536 nodes!

- 65,534 host addresses, one for network address and one for broadcast address.

RS: We use this for the CIS Lab network

Subnet: 255.255.255.0 (/24)



Network | Network | Network | Host

8 bits

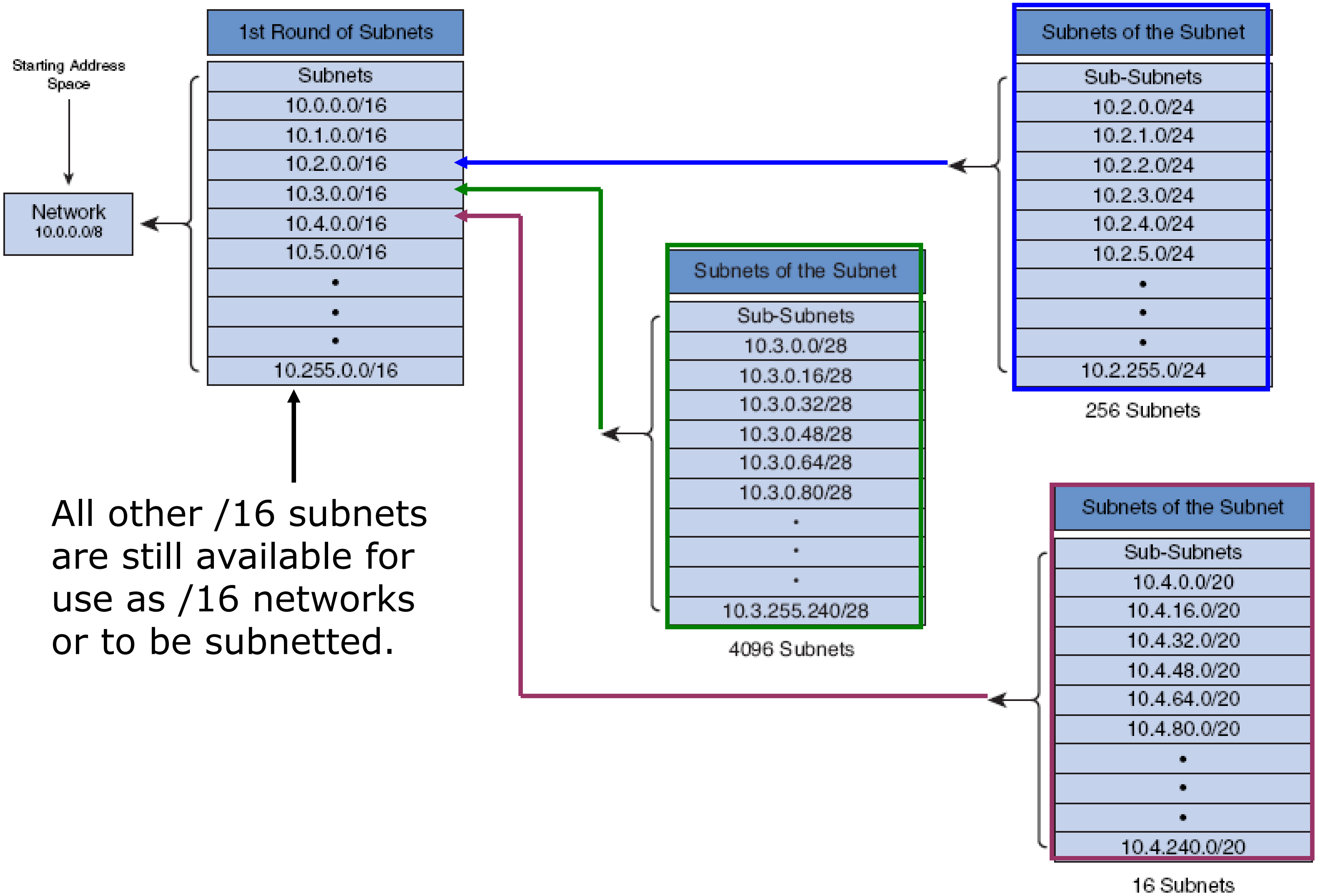
With 8 bits available for hosts, there a 2^8 possible addresses. That’s 256 nodes!

- 254 host addresses, one for network address and one for broadcast address.

RS: We are using a /24 network in room 2501.

That gives us 2^8 -2 (256 -2 = 254) host addresses. We drop by 2 because the first address (172.30.1.0) is the network address and the last address (172.30.1.255) is the broadcast address.

VLSM - Variable Length Subnet Masks
Subnet a subnet



Starting Address Space

Network
10.0.0.0/8

1st Round of Subnets
Subnets
10.0.0.0/16
10.1.0.0/16
10.2.0.0/16
10.3.0.0/16
10.4.0.0/16
10.5.0.0/16
•
•
•
10.255.0.0/16

Subnets of the Subnet
Sub-Subnets
10.2.0.0/24
10.2.1.0/24
10.2.2.0/24
10.2.3.0/24
10.2.4.0/24
10.2.5.0/24
•
•
•
10.2.255.0/24

256 Subnets

Subnets of the Subnet
Sub-Subnets
10.3.0.0/28
10.3.0.16/28
10.3.0.32/28
10.3.0.48/28
10.3.0.64/28
10.3.0.80/28
•
•
•
10.3.255.240/28

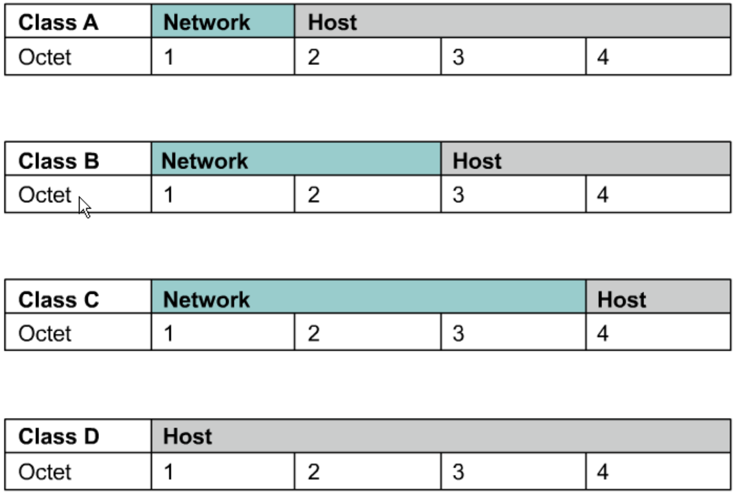
4096 Subnets

Subnets of the Subnet
Sub-Subnets
10.4.0.0/20
10.4.16.0/20
10.4.32.0/20
10.4.48.0/20
10.4.64.0/20
10.4.80.0/20
•
•
•
10.4.240.0/20

16 Subnets

All other /16 subnets are still available for use as /16 networks or to be subnetted.

Old Days:
Classful IP Addressing

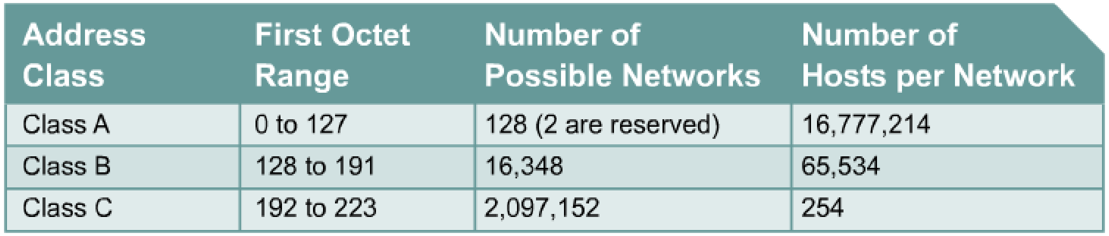


Class A	Network	Host		
Octet	1	2	3	4

Class B	Network		Host	
Octet	1	2	3	4

Class C	Network			Host
Octet	1	2	3	4

Class D	Host			
Octet	1	2	3	4



Address Class	First Octet Range	Number of Possible Networks	Number of Hosts per Network
Class A	0 to 127	128 (2 are reserved)	16,777,214
Class B	128 to 191	16,348	65,534
Class C	192 to 223	2,097,152	254

- In the early days of the Internet, IP addresses were allocated to organizations based on request rather than actual need.
- When an organization received an IP network address, that address was associated with a **"Class", A, B, or C.**
- This is known as **Classful IP Addressing**
- The **first octet** of the address determined what class the network belonged to and which bits were the network bits and which bits were the host bits.
- There were **no** subnet masks.
- It was not until 1992 when the IETF introduced CIDR (Classless Interdomain Routing), making the address class meaningless.
- This is known as **Classless IP Addressing**.

RS: We won’t be using classful IP addressing in CIS 192

Old days: Address Classes

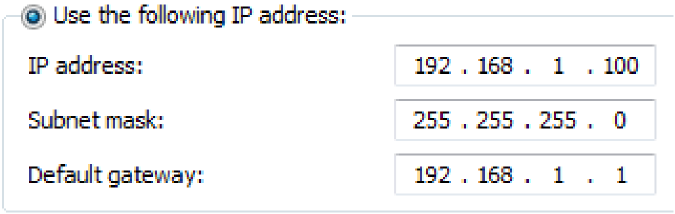
	1st octet	2nd octet	3rd octet	4th octet
Class A	Network	Host	Host	Host
Class B	Network	Network	Host	Host
Class C	Network	Network	Network	Host

N = Network number assigned by ARIN (American Registry for Internet Numbers)
H = Host number assigned by administrator

RS: HP has the 15 and 16 networks (or they used to). They got the 15 net in the early days. After buying Compaq (which bought DEC) they had the 16 net as well!

Special Unicast IPv4 Addresses

- **Default Route**



● Use the following IP address:

IP address: 192 . 168 . 1 . 100

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 1 . 1

- **Loopback Address**
  - Special address that hosts use to direct traffic to themselves.
  - 127.0.0.0 to 127.255.255.255

- **Link-Local Addresses (APIPA)**
  - 169.254.0.0 to 169.254.255.255 (169.254.0.0 /16)
  - Can be automatically assigned to the local host by the operating system in environments where no IP configuration is available.
  - Microsoft calls this APIPA (Automatic Private IP Addressing)

- **TEST-NET Addresses**
  - 192.0.2.0 to 192.0.2.255 (192.0.2.0 /24)
  - Set aside for teaching and learning purposes.
  - These addresses can be used in documentation and network examples.

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
```

When subnetting by hand I like to make these two tables first

subnetting using the ipcalc command

```
[root@elrond ~]# ipcalc -n 192.168.2.107 255.255.255.0
NETWORK=192.168.2.0

[root@elrond ~]# ipcalc -b 192.168.2.107 255.255.255.0
BROADCAST=192.168.2.255

[root@elrond ~]# ipcalc -p 192.168.2.107 255.255.255.0
PREFIX=24

[root@elrond ~]# ipcalc -nbp 172.30.1.0/24
PREFIX=24
BROADCAST=172.30.1.255
NETWORK=172.30.1.0
```

The ipcalc on Ubuntu is nicer but you have to install it with: ***apt-get install ipcalc***

```
cis192@frodo:~$ ipcalc 172.30.4.0/24
Address:   172.30.4.0           10101100.00011110.00000100. 00000000
Netmask:   255.255.255.0 = 24   11111111.11111111.11111111. 00000000
Wildcard:  0.0.0.255            00000000.00000000.00000000. 11111111
=>
Network:   172.30.4.0/24        10101100.00011110.00000100. 00000000
HostMin:   172.30.4.1           10101100.00011110.00000100. 00000001
HostMax:   172.30.4.254         10101100.00011110.00000100. 11111110
Broadcast: 172.30.4.255         10101100.00011110.00000100. 11111111
Hosts/Net: 254                   Class B, Private Internet
```

subnetting example problem - by hand

Given the following IP address and network mask, what is the network address?

IP: 192.168.30.100
Netmask: 255.255.240.0

The first two octets of the mask are 255 so we will start the network address as 192.168.?.0. This mask indicates a /20 network (8 + 8 + 4). Next we need to apply the decimal 240 mask (1111 0000) to decimal 30 (0001 1110) which gives us binary 0001 0000 or decimal 16. Our network address is 192.168.16.0.

a) 192.168.30.0
b) 192.168.24.0
c) 192.168.15.0
d) 192.168.16.0

subnetting example problem - by CentOS ipcalc

Given the following IP address and network mask, what is the network address?

IP: 192.168.30.100
Netmask: 255.255.240.0

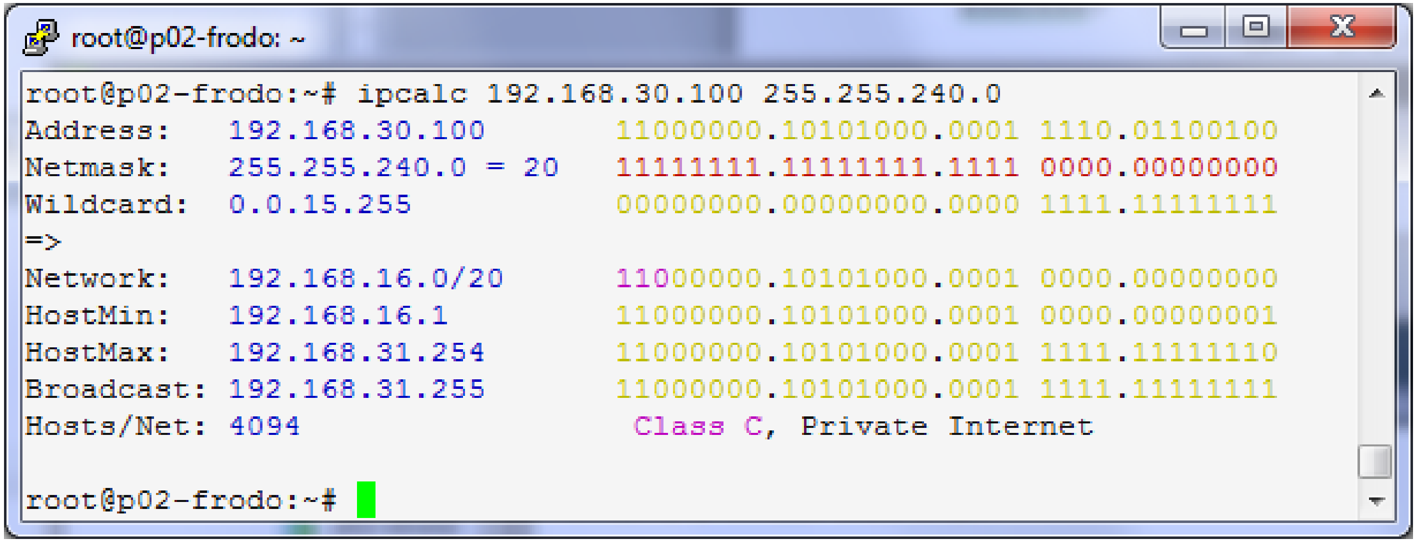
```
[root@elrond ~]# ipcalc -n 192.168.30.100 255.255.240.0
NETWORK=192.168.16.0
```

a) 192.168.30.0
b) 192.168.24.0
c) 192.168.15.0
d) 192.168.16.0

subnetting example problem - by Ubuntu ipcalc

Given the following IP address and network mask, what is the network address?

IP: 192.168.30.100
Netmask: 255.255.240.0



```
root@p02-frodo:~# ipcalc 192.168.30.100 255.255.240.0
Address:   192.168.30.100        11000000.10101000.0001 1110.01100100
Netmask:   255.255.240.0 = 20    11111111.11111111.1111 0000.00000000
Wildcard:  0.0.15.255            00000000.00000000.0000 1111.11111111
=>
Network:   192.168.16.0/20       11000000.10101000.0001 0000.00000000
HostMin:   192.168.16.1          11000000.10101000.0001 0000.00000001
HostMax:   192.168.31.254        11000000.10101000.0001 1111.11111110
Broadcast: 192.168.31.255        11000000.10101000.0001 1111.11111111
Hosts/Net: 4094                   Class C, Private Internet

root@p02-frodo:~#
```

a) 192.168.30.0
b) 192.168.24.0
c) 192.168.15.0
d) 192.168.16.0

Team Exercise - IPv4 Addressing

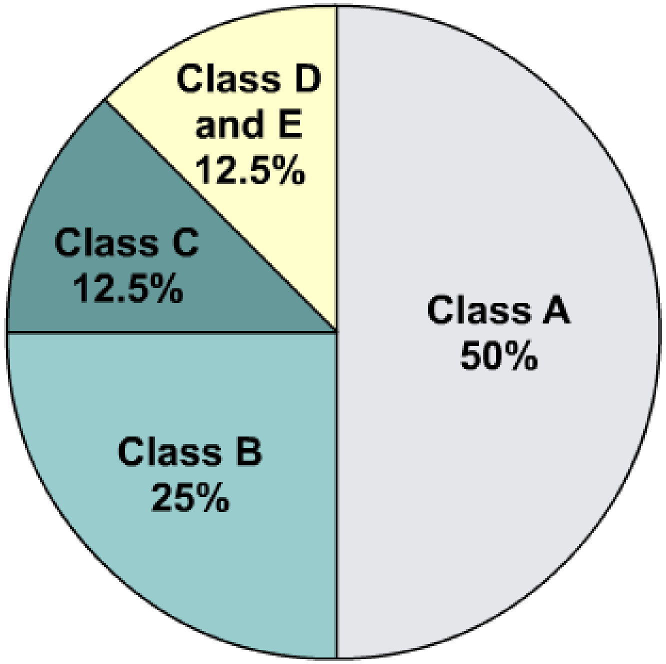
http://simms-teach.com/docs/cis192/ip-exercise.pdf

Q1, Q5, Q9 - Breakout room 1
Q2, Q6, Q10 - Breakout room 2
Q3, Q7, Q11 - Breakout room 3
Q4, Q8, Q12 - Breakout room 4

Q4

NAT/PAT
and
IPv6

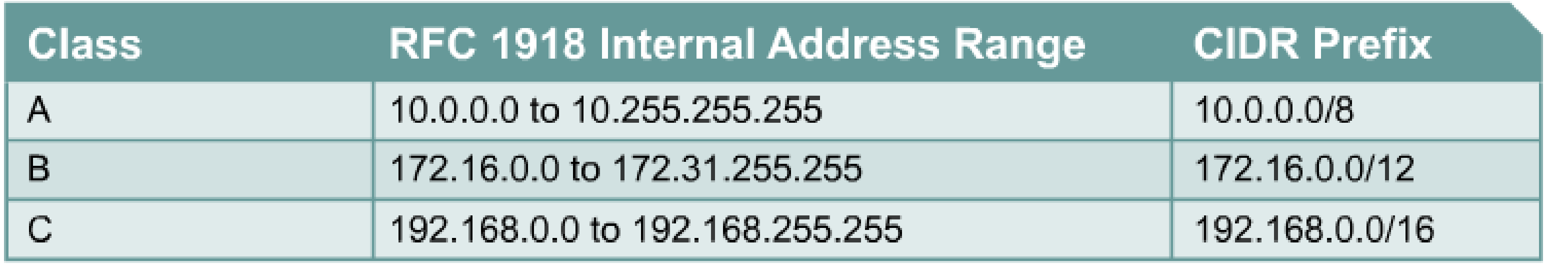
IP addressing crisis



With Class A and B addresses virtually exhausted, Class C addresses (12.5 percent of the total space) are left to assign to new networks.

- Address Depletion
- Internet Routing Table Explosion

Short Term Solutions: IPv4 Enhancements

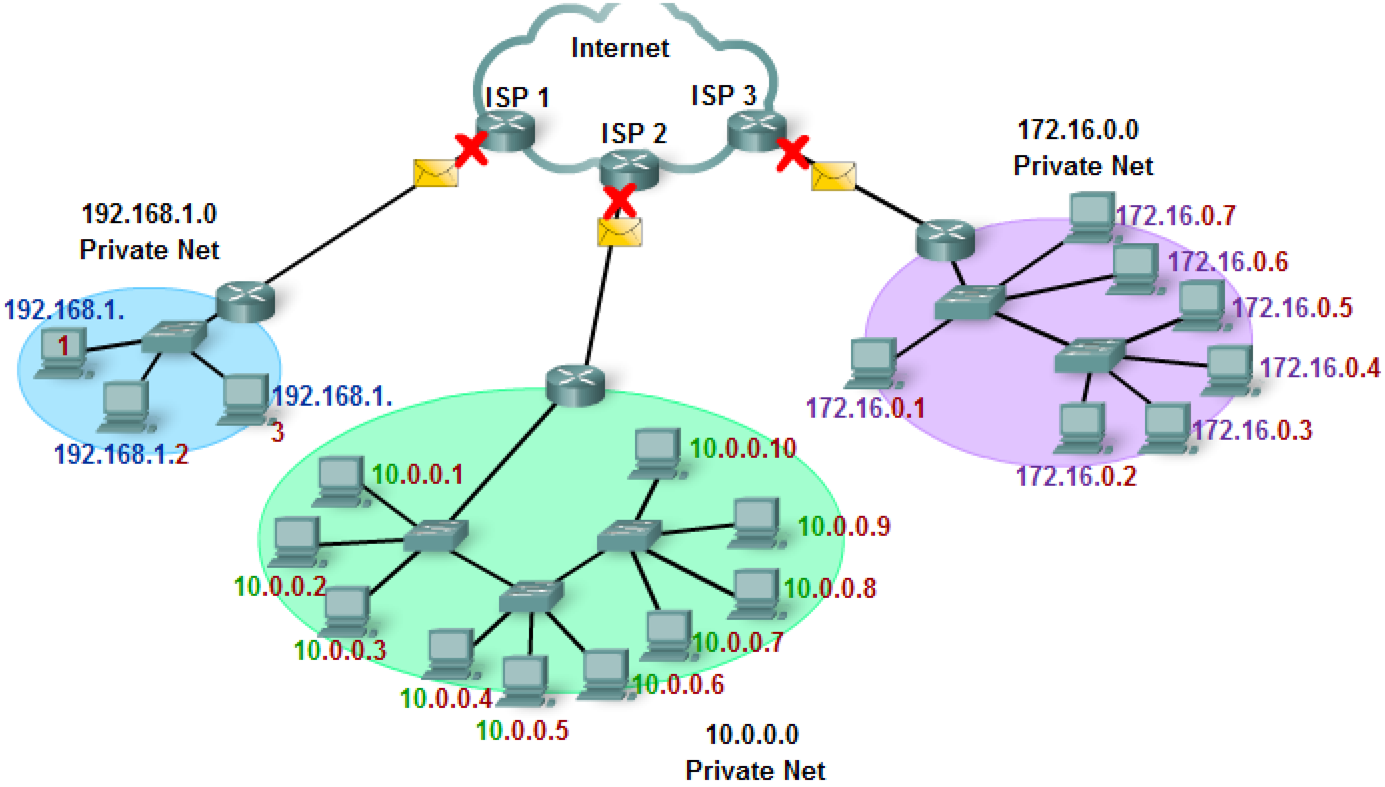


Class	RFC 1918 Internal Address Range	CIDR Prefix
A	10.0.0.0 to 10.255.255.255	10.0.0.0/8
B	172.16.0.0 to 172.31.255.255	172.16.0.0/12
C	192.168.0.0 to 192.168.255.255	192.168.0.0/16

- CIDR (Classless Inter-Domain Routing) - RFCs 1517, 1518, 1519, 1520
- VLSM (Variable Length Subnet Mask) - RFC 1009

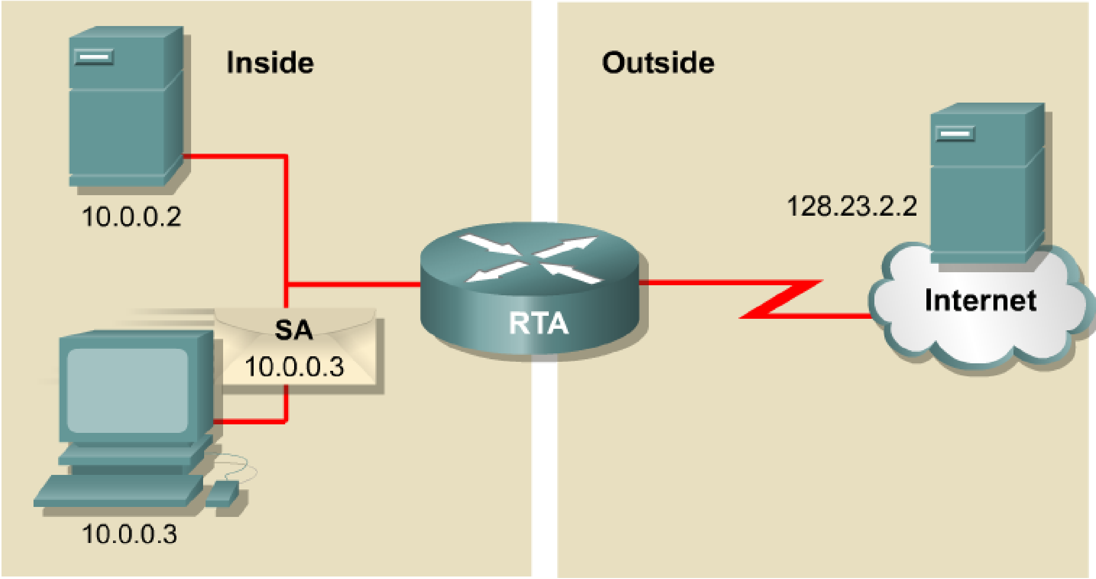
- Private Addressing - RFC 1918
- NAT/PAT (Network Address Translation / Port Address Translation)
  - More later when we discuss TCP

Private IP Addresses



- RFC 1918
  - 10.0.0.0 to
  - 172.16.0.0 to 172.31.255.255 (172.16.0.0 /12)
  - 192.168.0.0 to 192.168.255.255 (192.168.0.0 /16)
- The addresses will not be routed in the Internet
  - Need NAT/PAT (next)
- Should be blocked by your ISP
- Allows for any network to have up to 16,777,216 hosts (/8)

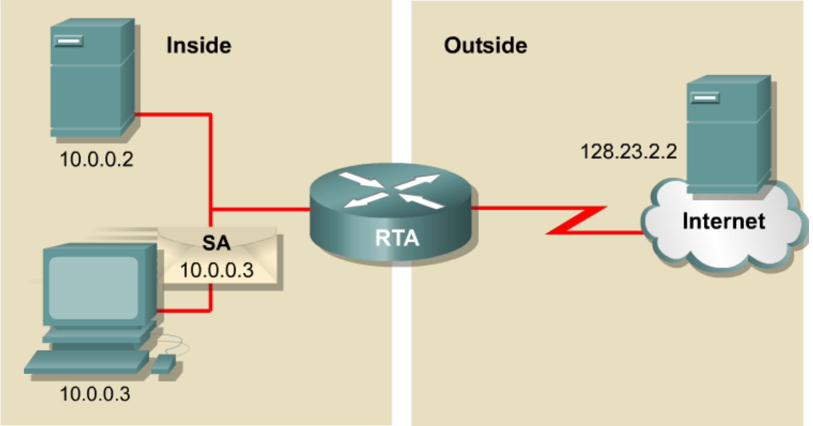
Introducing NAT and PAT



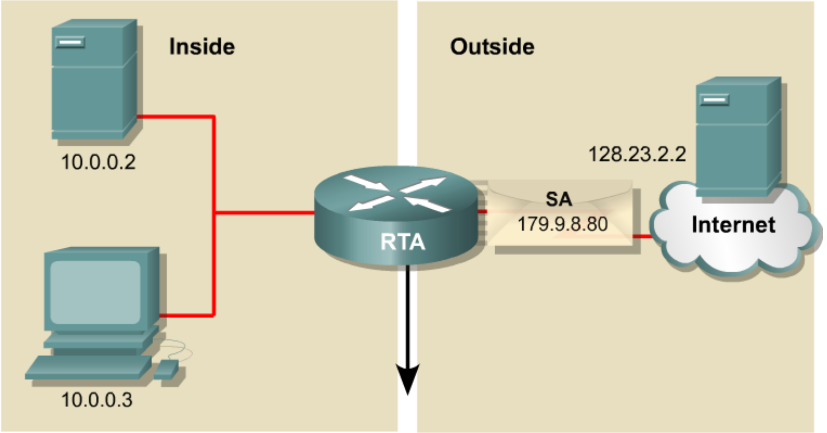
- NAT is designed to conserve IP addresses and enable networks to use private IP addresses on internal networks.
- These private, internal addresses are translated to routable, public addresses.
- IPv4 addresses are almost depleted.
- NAT/PAT has allowed IPv4 to be the predominant network protocol, keeping IPv6 at-bay (for now).

NAT Example

1



2



NAT Table		
Inside Local IP Address	Inside Global IP Address	Outside Global IP Address
10.0.0.3	179.9.8.80	128.23.2.2

DA	SA		
128.23.2.2	10.0.0.3	....	Data

1 **IP Header**

→

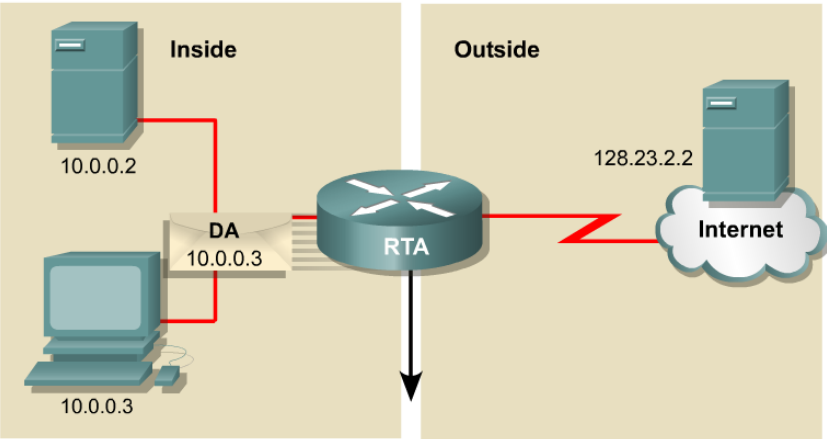
DA	SA		
128.23.2.2	179.9.8.80	....	Data

2 **IP Header**

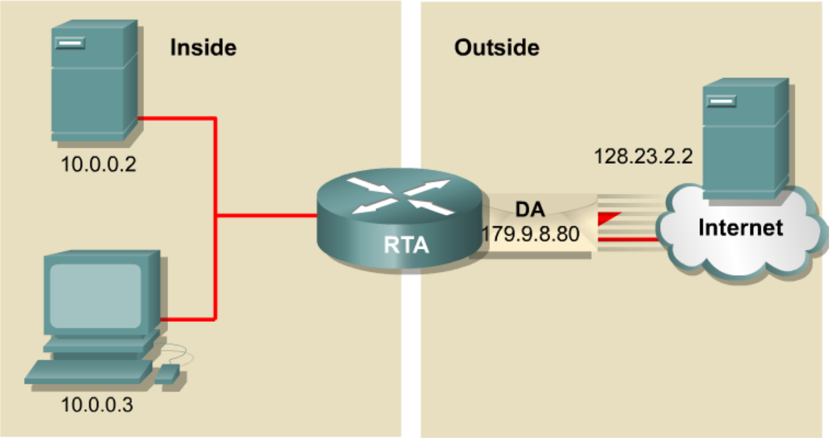
The translation from Private <u>source</u> IP address to Public <u>source</u> IP address.

NAT Example

4



3



NAT Table

Inside Local IP Address	Inside Global IP Address	Outside Global IP Address
10.0.0.2	179.9.8.80	128.23.2.2
10.0.0.3	179.9.8.80	128.23.2.2

DA	SA		
10.0.0.3	128.23.2.2	....	Data

4 **IP Header**

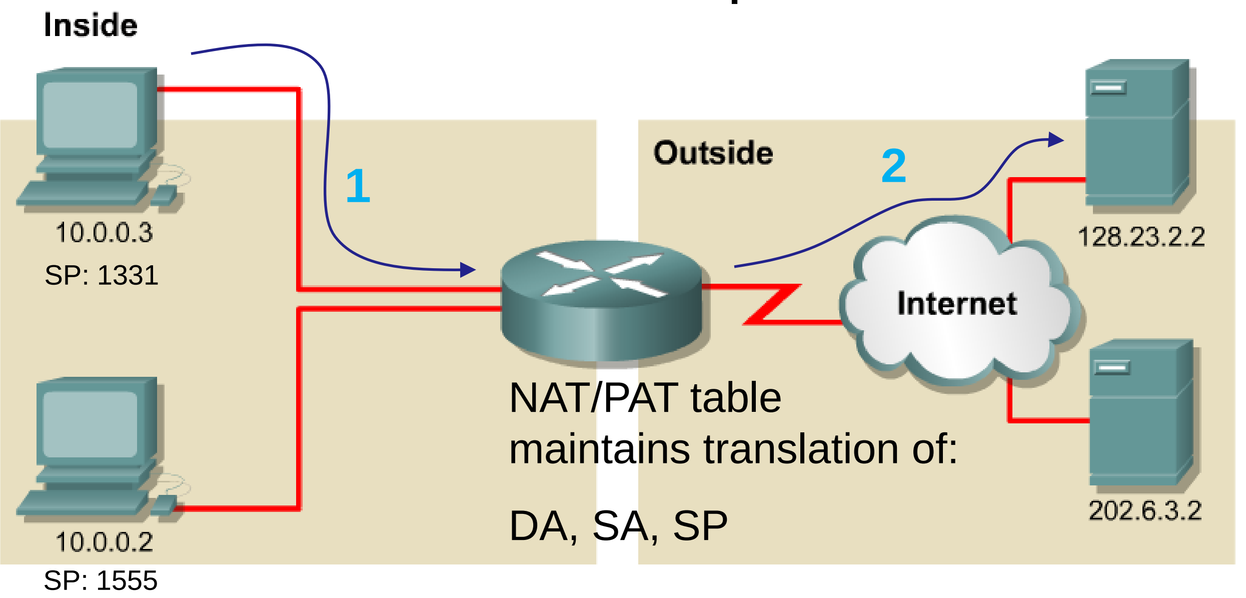
DA	SA		
179.9.8.80	128.23.2.2	....	Data

3 **IP Header**

Translation back, from Public destination IP address to Private destination IP address.

RS: The main downfall of NAT is that you may not have a big enough pool of public addresses for every internal host needing to use the Internet at the same time.

PAT Example



NAT/PAT table maintains translation of:

DA, SA, SP

1

DA	SA	DP	SP	
128.23.2.2	10.0.0.3	80	1331	Data

IP Header | TCP/UDP Header

→ *translated*

2

DA	SA	DP	SP	
128.23.2.2	179.9.8.80	80	3333	Data

IP Header | TCP/UDP Header

DA	SA	DP	SP	
128.23.2.2	10.0.0.2	80	1555	Data

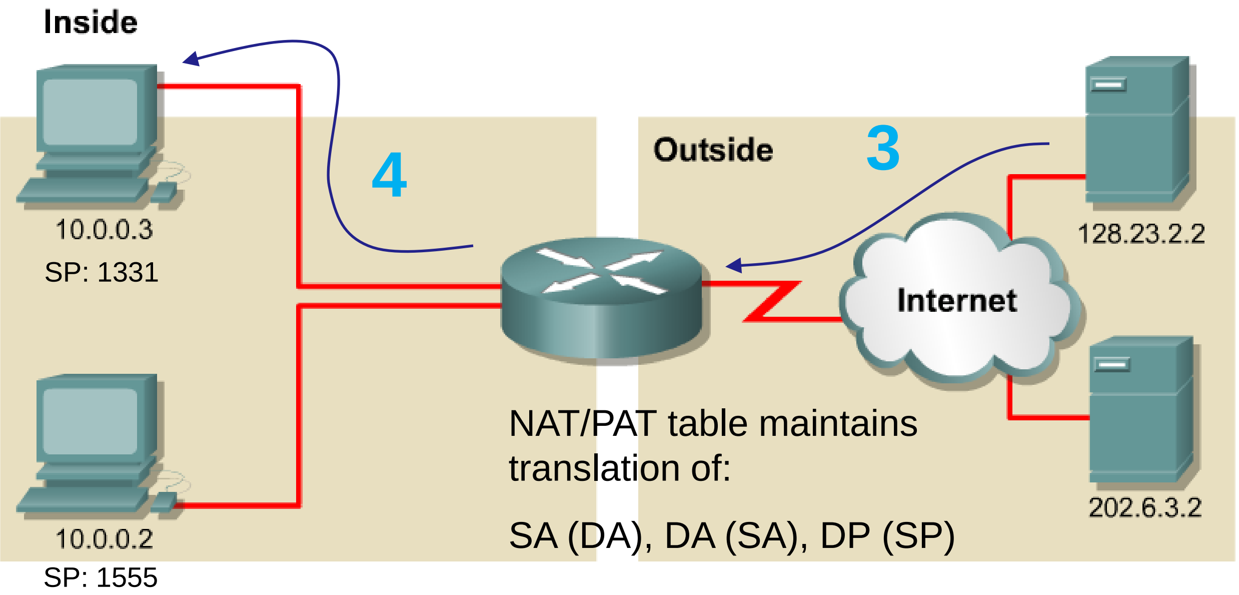
IP Header | TCP/UDP Header

→

DA	SA	DP	SP	
128.23.2.2	179.9.8.80	80	2222	Data

IP Header | TCP/UDP Header

PAT Example



NAT/PAT table maintains translation of:

SA (DA), DA (SA), DP (SP)

DA	SA	DP	SP	
10.0.0.3	128.23.2.2	1331	80	Data

4 IP Header TCP/UDP Header

← *translated*

DA	SA	DP	SP	
179.9.8.80	128.23.2.2	3333	80	Data

3 IP Header TCP/UDP Header

DA	SA	DP	SP	
10.0.0.2	128.23.2.2	1555	80	Data

IP Header TCP/UDP Header

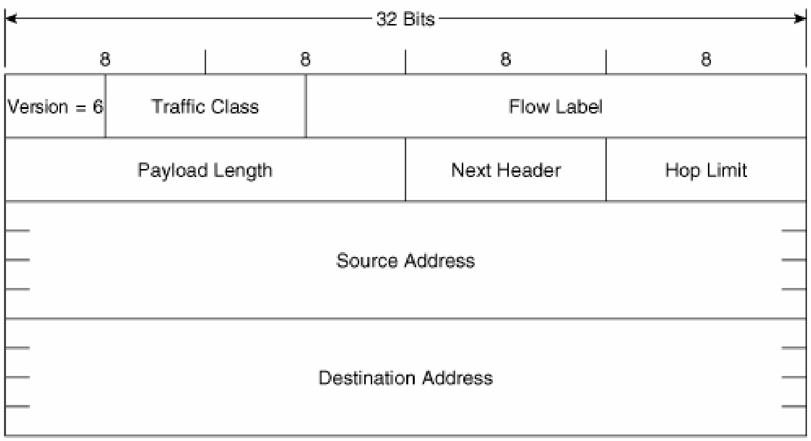
←

DA	SA	DP	SP	
179.9.8.80	128.23.2.2	2222	80	Data

IP Header TCP/UDP Header

Long Term Solution: IPv6

Figure 2-5. The IPv6 packet header.



- IPv6 replaces the 32-bit IPv4 address with a **128-bit address**, making ***340** **trillion trillion trillion*** **IP addresses** available.

  340,282,366,920,938,463,463,374,607,431,768,211,456 addresses

  - Represented by breaking them up into **eight 16-bit segments.**
  - **Each segment** is written in **hexadecimal** between 0x0000 and 0xFFFF, separated by colons.

- An example of a written IPv6 address is

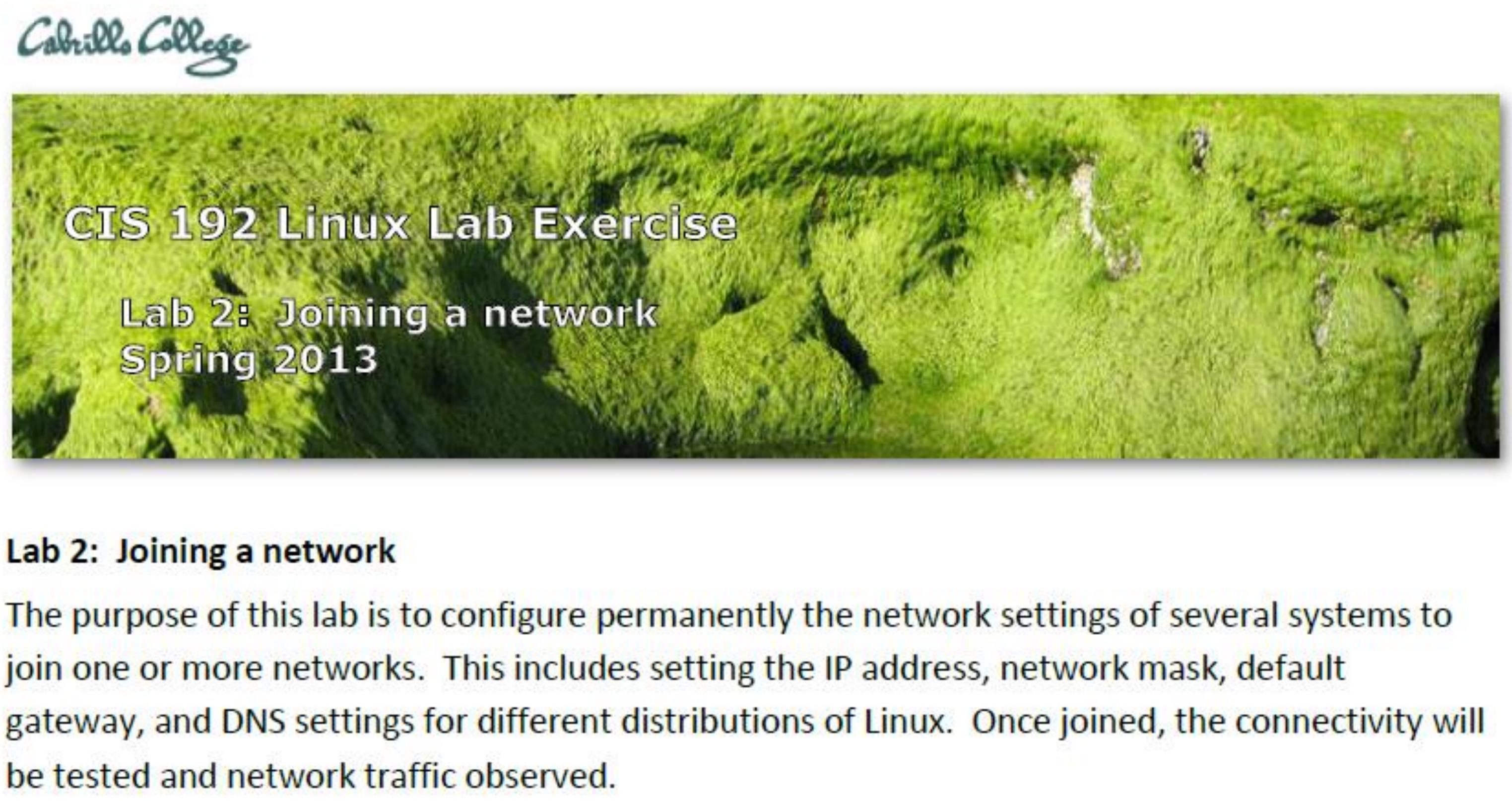
  3ffe:1944:0100:000a:0000:00bc:2500:0d0b

Long Term Solution: IPv6 (coming)

- IPv6 has been slow to arrive
- IPv6 requires new software; IT staffs must be retrained
- IPv6 will most likely coexist with IPv4 for years to come.
- Some experts believe IPv4 will remain for more than 10 years.

See Rick’s presentation on IPv6 for an excellent overview

Lab



CIS 192 Linux Lab Exercise

Lab 2: Joining a network
Spring 2013

Lab 2: Joining a network

The purpose of this lab is to configure permanently the network settings of several systems to join one or more networks. This includes setting the IP address, network mask, default gateway, and DNS settings for different distributions of Linux. Once joined, the connectivity will be tested and network traffic observed.

Pearls of Wisdom:

- Don't start CIS Lab Assignments at the last minute!
- The slower you go, the sooner you will be finished!
- Use the forum!

Some essentials for doing labs

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

- Troubleshoot starting at Layer 1 and work up
- 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.

Wrap

New commands, tools and services:

arp
ifconfig
ip
ipcalc
mtr
netconfig or system-config-network
netstat
ping
ping6
tcpdump
traceroute

service network restart
/etc/init.d/networking start (Ubuntu)

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:

Next Class

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

Quiz questions for next class:

- What does the C flag mean when viewing ARP cache entries with arp -n?
- What Wireshark display filter would only show ARP and 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?

Backup