Lesson Module Status

- Slides – draft
- Properties - done
- Flashcards - na
- 1st minute quiz – na, test today
- Web Calendar summary – done
- Web book pages –
- Commands – done
- Howtos – na
- Skills pacing - na
- Lab – na
- Depot (VMs) – added VBox VMs
- Labs published – na, no new labs this week
- Real test publish – done
- Real test copies made – done
- Set up mystery server for test driver question - done
- New quiz ?'s for next week – NA
- Add email option for all lesson quizzes

Course history and credits

Jim Griffin

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

Rick Graziani

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

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

STUDENT LOG IN

View Teach & Confer Archives

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


Joe A.

Joe P.


Chris B.


Chuck


Rich

Ryan


Josh

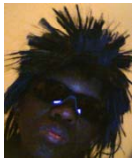

John

Robert

Patrick


Casady

Chris H.


Lieven


Jack

Julio

Edwin

Kay

Aaron


Joe B.

Junious


Drew


Edgar

Brynden

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

Quiz

No Quiz Today

The Application Layer

Objectives	Agenda
• 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. • Configure a network service with security restrictions for its use using either TCP Wrappers or a superdaemon.	• No Quiz today • Questions on previous material • Housekeeping • Review • TCP continued • Security issues • Application layer • xinetd and Telnet • Very Secure FTP • Wrap • Test 1

Questions on previous material

Questions?

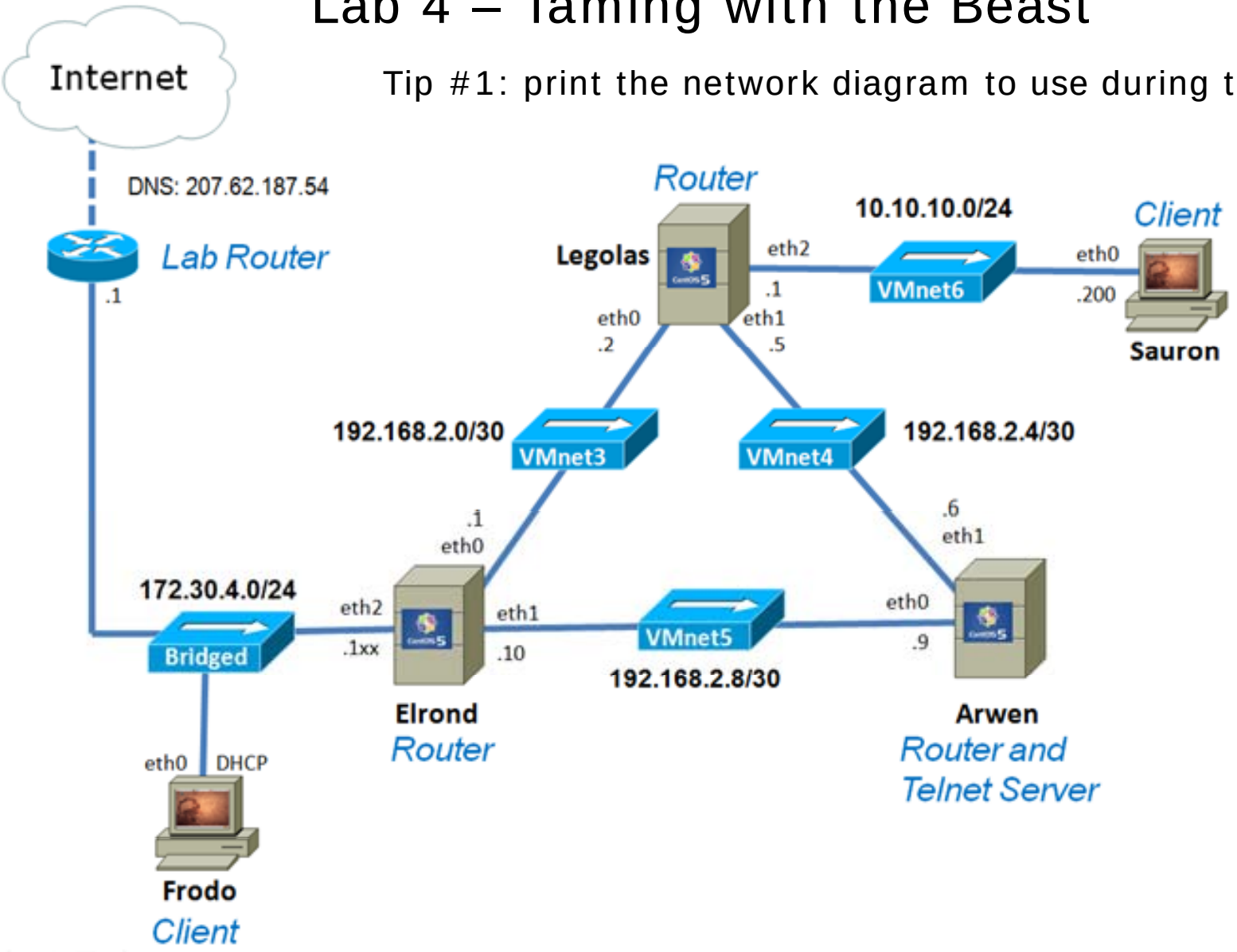
- Previous lesson material
- Lab assignment
- How this class works

Taming the Beast

(Lab 4)

Lab 4 – Taming with the Beast

Tip #1: print the network diagram to use during the lab



Lab 4 – Taming the Beast

Tip #2: Use permanent network settings

On Legolas …

```
[root@legolas ~]# head /etc/sysconfig/{network,network-scripts/ifcfg-eth[0-3]}
==> /etc/sysconfig/network <==
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=legolas.localdomain
GATEWAY=192.168.2.1

==> /etc/sysconfig/network-scripts/ifcfg-eth0 <==
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.2.2
NETMASK=255.255.255.252

==> /etc/sysconfig/network-scripts/ifcfg-eth1 <==
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.2.5
NETMASK=255.255.255.252

==> /etc/sysconfig/network-scripts/ifcfg-eth2 <==
DEVICE=eth2
ONBOOT=yes
BOOTPROTO=static
IPADDR=10.10.10.1
NETMASK=255.255.255.0
[root@legolas ~]#
```

You can restart the VM or the network services without having to reconfigure interfaces

… do the same on the other VMs

Lab 4 – Taming the Beast

Tip #3: Populate /etc/hosts files with names used in Lab 4

On Elrond …

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

172.30.4.1 router
192.168.2.2 legolas
192.168.2.9 arwen
10.10.10.200 sauron

[root@elrond ~]#
```

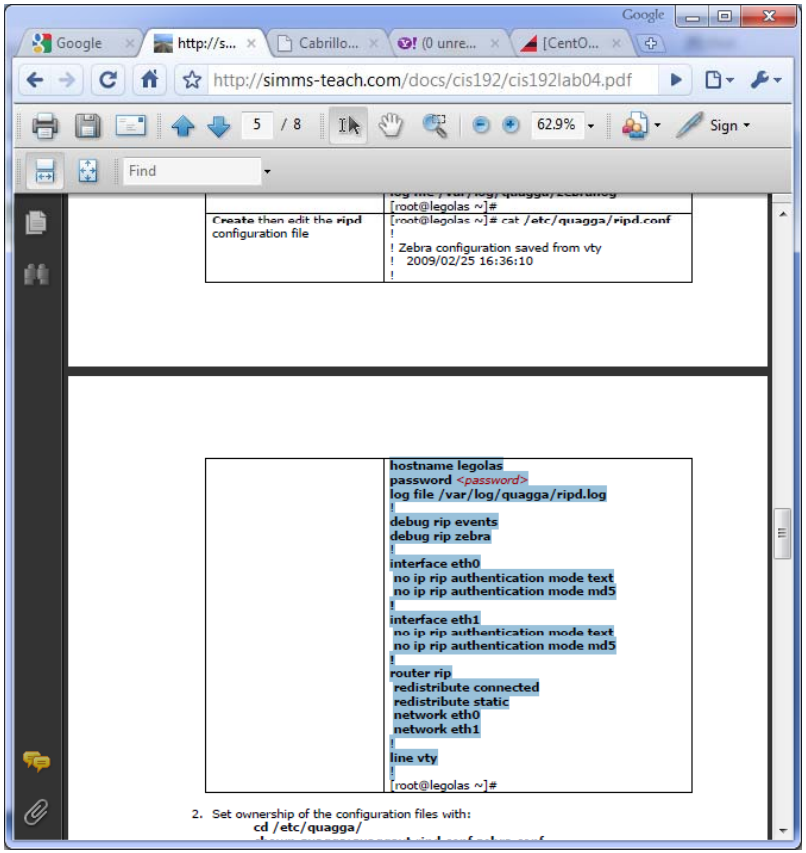
Do the same for Arwen, Frodo, and Sauron and then you can use names rather than IP address for testing and troubleshooting

On Legolas …

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

192.168.2.6 arwen
192.168.2.1 elrond
172.30.4.1 router
10.10.10.200 sauron

[root@legolas ~]#
```

Lab 4 – Taming the Beast

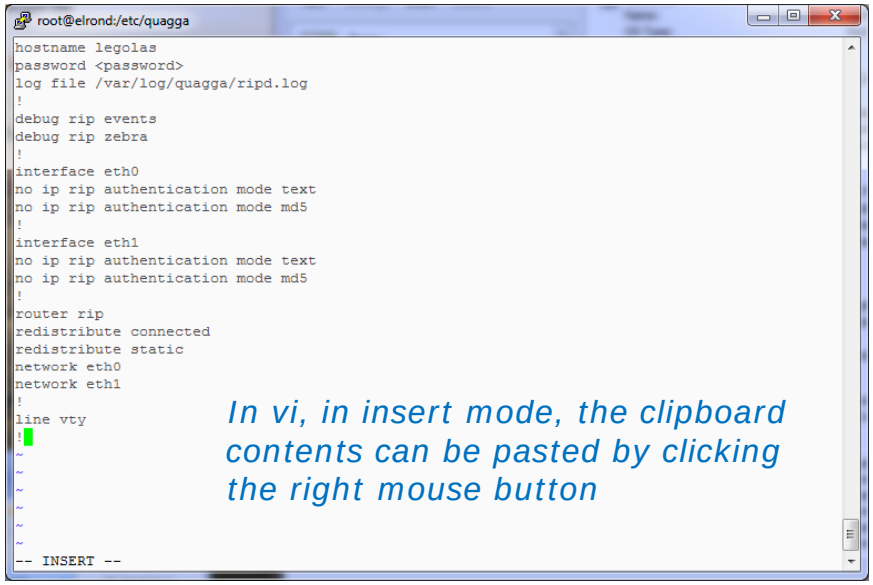
Tip #4: Use Putty to copy/paste Quagga configuration files



Lab 4 PDF

Putty into Elrond at 172.30.x.1xx …

```
login as: root
root@172.30.1.125's password:
Last login: Tue Mar  9 23:04:46 2010 from
172.30.1.150
[root@elrond ~]# cd /etc/quagga
[root@elrond quagga]# vi ripd.conf
```



In vi, in insert mode, the clipboard contents can be pasted by clicking the right mouse button

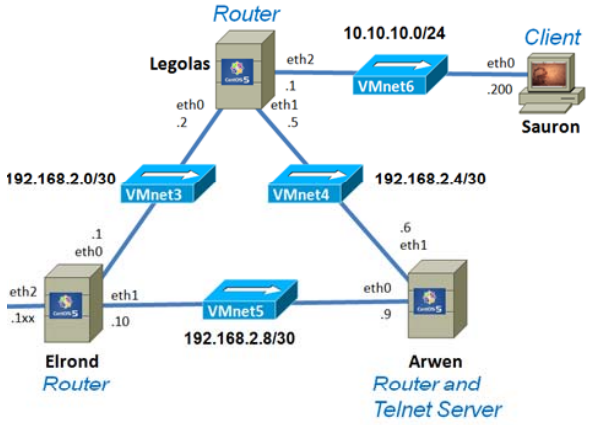
/etc/quagga/ripd.conf on Elrond

After copying, be sure and modify the hostname to match the system you are configuring

Lab 4 – Taming the Beast

Tip #5: Use scp command to copy Quagga configuration files

After Elrond is configured, ssh into Legolas and use scp to copy Quagga configuration files from Elrond. Repeat with Arwen.




```
Using username "root".
root@172.30.1.125's password:
Last login: Tue Mar  9 22:43:57 2010
[root@elrond ~]# ssh legolas
root@legolas's password:
Last login: Tue Mar  9 22:04:50 2010
[root@legolas ~]# cd /etc/quagga
[root@legolas quagga]# scp elrond:/etc/quagga/[rz]*conf .
The authenticity of host 'elrond (192.168.2.1)' can't be established.
RSA key fingerprint is ec:e3:71:94:41:bb:9a:c6:61:a8:06:2c:ed:f2:d7:7c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'elrond,192.168.2.1' (RSA) to the list of known hosts.
root@elrond's password:
ripd.conf                                     100%  400     0.4KB/s   00:00
zebra.conf                                    100%   70     0.1KB/s   00:00
[root@legolas quagga]#
```

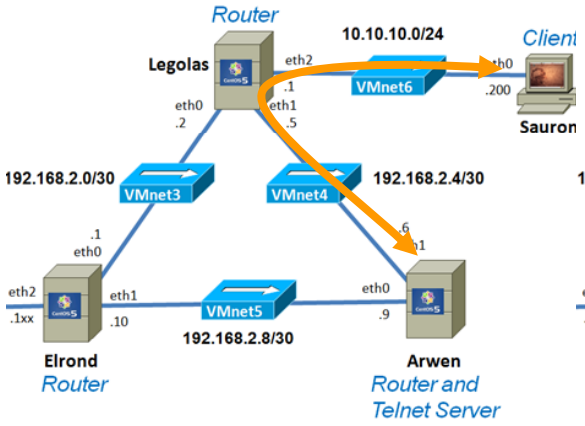
After copying, be sure and modify the hostname to match the system you are configuring

Playing with the Beast

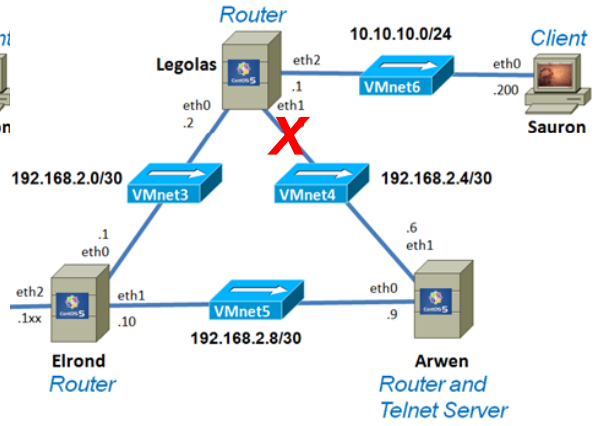
(Lab 4)

Lab 4 – Playing with the Beast

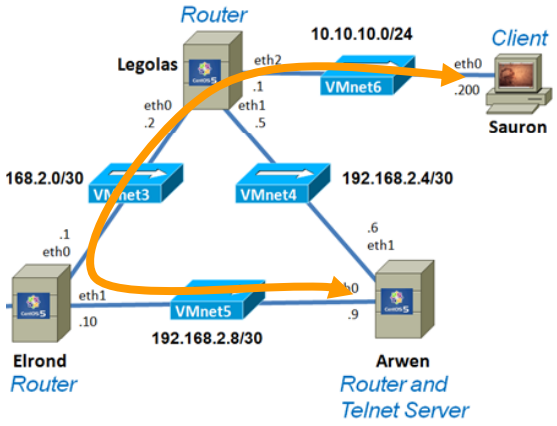
Playing #1: Force routing table to adapt to network changes you make



Pinging Sauron from Arwen via Legolas



Making trouble: The cable attached to the second Ethernet Controller on Legolas (eth1) is disconnected



After 180 failed pings, routing tables adjust and a new, longer route via Elrond and Legolas is used

In Lab 4 you can observe routing tables update themselves as the network changes

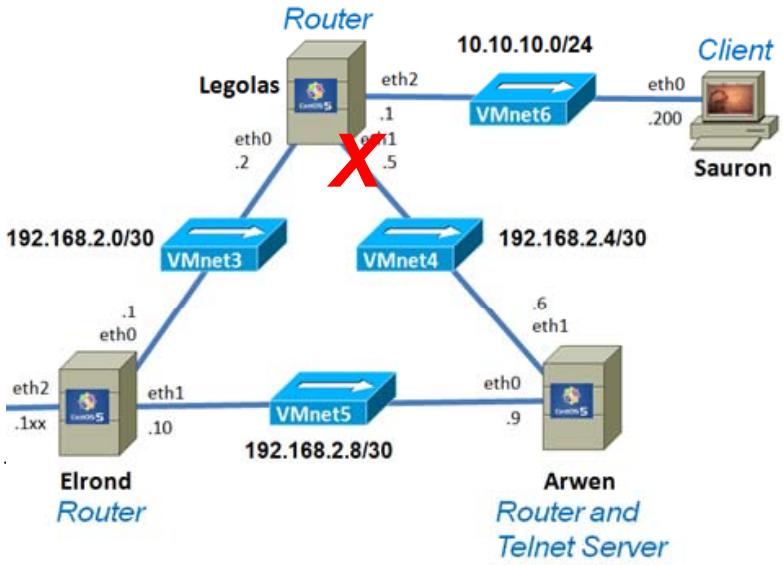
Lab 4 – Playing with the Beast

```
[root@arwen ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.2.8     0.0.0.0         255.255.255.252 U     0      0        0 eth0
192.168.2.0     192.168.2.5     255.255.255.252 UG    2      0        0 eth1
192.168.2.4     0.0.0.0         255.255.255.252 U     0      0        0 eth1
172.30.1.0      192.168.2.10    255.255.255.0   UG    2      0        0 eth0
10.10.10.0      192.168.2.5     255.255.255.0   UG    2      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
```

Pinging Sauron from Arwen via Legolas

Trouble: cable on Legolas eth1 is disconnected

```
[root@arwen ~]# ping sauron
PING sauron (10.10.10.200) 56(84) bytes of data.
64 bytes from sauron (10.10.10.200): icmp_seq=1 ttl=63 time=5.60 ms
< snipped >
64 bytes from sauron (10.10.10.200): icmp_seq=7 ttl=63 time=6.20 ms
From 192.168.2.6 icmp_seq=53 Destination Host Unreachable
< snipped >
From 192.168.2.6 icmp_seq=181 Destination Host Unreachable
ping: sendmsg: Network is unreachable
< snipped >
ping: sendmsg: Network is unreachable
64 bytes from sauron (10.10.10.200): icmp_seq=188 ttl=62 time=7.16 ms
< snipped >
64 bytes from sauron (10.10.10.200): icmp_seq=226 ttl=62 time=6.27 ms

--- sauron ping statistics ---
226 packets transmitted, 46 received, +126 errors, 79% packet
rtt min/avg/max/mdev = 0.142/5.446/21.506/3.115 ms, pipe 4
[root@arwen ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.2.8     0.0.0.0         255.255.255.252 U     0      0        0 eth0
192.168.2.0     192.168.2.10    255.255.255.252 UG    2      0        0 eth0
192.168.2.4     0.0.0.0         255.255.255.252 U     0      0        0 eth1
172.30.1.0      192.168.2.10    255.255.255.0   UG    2      0        0 eth0
10.10.10.0      192.168.2.10    255.255.255.0   UG    3      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
[root@arwen ~]#
```



After 180 failed pings, routing tables adjust and now use longer route via Elrond and Legolas

Lab 4 – Playing with the Beast

Playing #2: Debug RIP events and packets

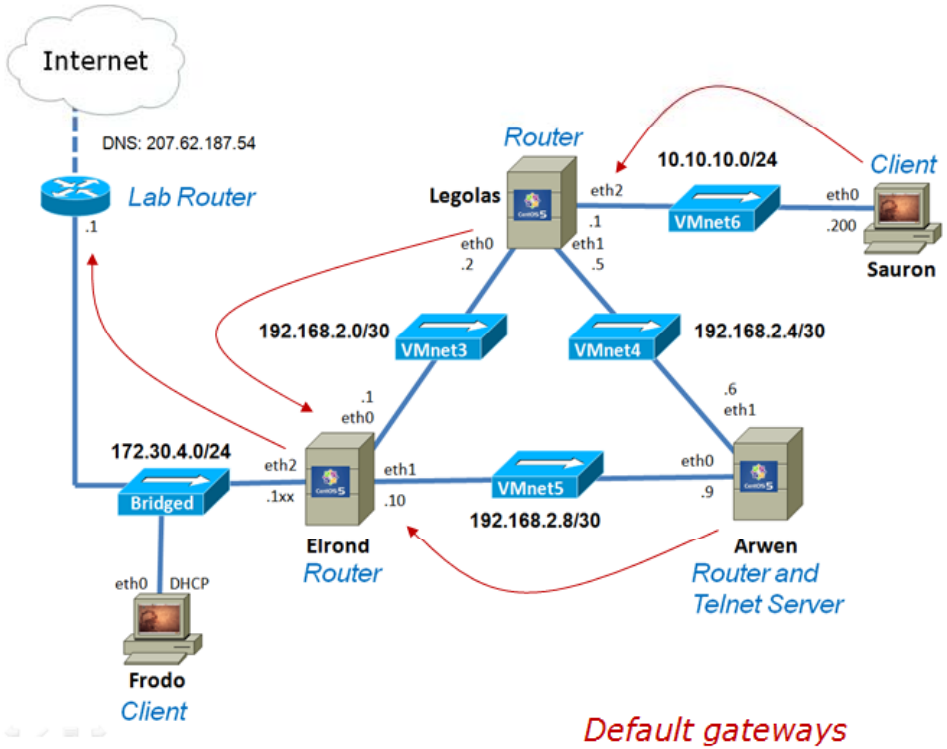
```
[root@legolas ~]# vtysh

Hello, this is Quagga (version 0.98.6).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

legolas.localdomain# debug rip events
legolas.localdomain# debug rip packet
legolas.localdomain# exit
[root@legolas ~]# tail -f /var/log/quagga/ripd.log
2010/03/10 00:39:43 RIP: ignore packet comes from myself
2010/03/10 00:39:47 RIP: RECV packet from 192.168.2.1 port 520 on eth0
2010/03/10 00:39:47 RIP: RECV RESPONSE version 2 packet size 64
2010/03/10 00:39:47 RIP:   0.0.0.0/0 -> 0.0.0.0 family 2 tag 0 metric 1
2010/03/10 00:39:47 RIP:   172.30.1.0/24 -> 0.0.0.0 family 2 tag 0 metric 1
2010/03/10 00:39:47 RIP:   192.168.2.8/30 -> 0.0.0.0 family 2 tag 0 metric 1
2010/03/10 00:39:55 RIP: RECV packet from 192.168.2.6 port 520 on eth1
2010/03/10 00:39:55 RIP: RECV RESPONSE version 2 packet size 84
[root@legolas ~]#
```

Use the debug command to enable debugging

*Use **tail** with the **–f** option to monitor debug messages as they are written to **/var/quagga/ripd.conf***

Lab 4 – Playing with the Beast

Playing #3: Use RIP to set default gateways on Legolas and Arwen



In Lab 4 we manually set up default routes before configuring dynamic routing on Elrond, Legolas, and Arwen

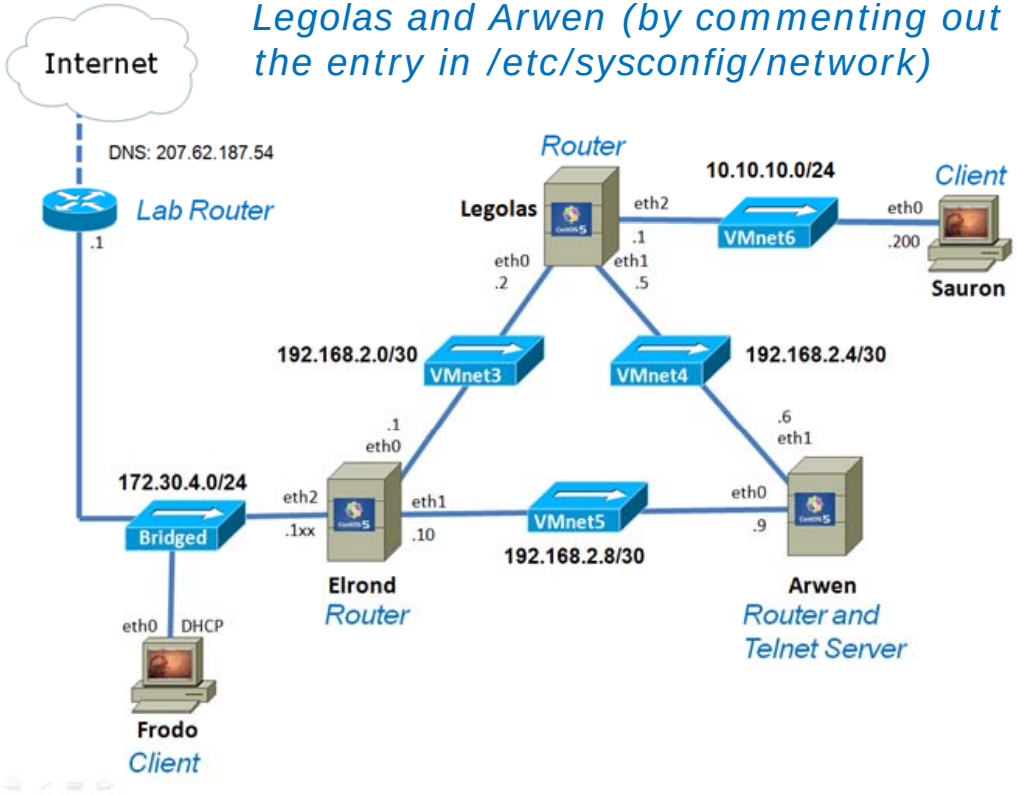
Rather than manually configuring the default gateway on Legolas and Arwen, Elrond can propagate its default gateway using the RIP ***default-information originate*** *command to Legolas and Arwen*

Lab 4 – Playing with the Beast

Playing #3: Use RIP to set default gateways on Legolas and Arwen

```
[root@elrond ~]# cat /etc/quagga/ripd.conf
hostname elrond
password Cabri11o
log file /var/log/quagga/ripd.log
!
debug rip events
debug rip zebra
!
interface eth0
no ip rip authentication mode text
no ip rip authentication mode md5
!
interface eth1
no ip rip authentication mode text
no ip rip authentication mode md5
!
router rip
redistribute connected
redistribute static
network eth0
network eth1
default-information originate
!
line vty
!
[root@elrond ~]#
```

The default gateways were removed on Legolas and Arwen (by commenting out the entry in /etc/sysconfig/network)



*The **default-information originate** command is added just on Elrond (not Legolas or Arwen) to propagate the default gateway using RIP to Legolas and Arwen*

Lab 4 – Playing with the Beast

Playing #3: Use RIP to set default gateways on Legolas and Arwen

... on Legolas

```
[root@elrond ~]# ssh legolas
root@legolas's password:
Last login: Tue Mar  9 23:36:55 2010 from elrond
[root@legolas ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.2.8     192.168.2.1     255.255.255.252 UG    2      0        0 eth0
192.168.2.0     0.0.0.0         255.255.255.252 U     0      0        0 eth0
192.168.2.4     0.0.0.0         255.255.255.252 U     0      0        0 eth1
172.30.1.0      192.168.2.1     255.255.255.0   UG    2      0        0 eth0
10.10.10.0      0.0.0.0         255.255.255.0   U     0      0        0 eth2
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth2
0.0.0.0         192.168.2.1     0.0.0.0         UG    2      0        0 eth0
[root@legolas ~]# vtysh

Hello, this is Quagga (version 0.98.6).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

legolas.localdomain# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

R>* 0.0.0.0/0 [120/2] via 192.168.2.1, eth0, 00:55:43
C>* 10.10.10.0/24 is directly connected, eth2
C>* 127.0.0.0/8 is directly connected, lo
K>* 169.254.0.0/16 is directly connected, eth2
R>* 172.30.1.0/24 [120/2] via 192.168.2.1, eth0, 00:55:43
C>* 192.168.2.0/30 is directly connected, eth0
C>* 192.168.2.4/30 is directly connected, eth1
R>* 192.168.2.8/30 [120/2] via 192.168.2.1, eth0, 00:55:43
legolas.localdomain#
```

Legolas has learned the default gateway from Elrond using RIP because of the ***default-information originate*** *command added on Elrond*

Lab 4 – Playing with the Beast

Playing #3: Use RIP to set default gateways on Legolas and Arwen

```
[root@legolas ~]# vtysh

Hello, this is Quagga (version 0.98.6).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

legolas.localdomain# debug rip events
legolas.localdomain# debug rip packet
legolas.localdomain# exit
[root@legolas ~]# tail -f /var/log/quagga/ripd.log
2010/03/10 00:39:43 RIP: ignore packet comes from myself
2010/03/10 00:39:47 RIP: RECV packet from 192.168.2.1 port 520 on eth0
2010/03/10 00:39:47 RIP: RECV RESPONSE version 2 packet size 64
2010/03/10 00:39:47 RIP:   0.0.0.0/0 -> 0.0.0.0 family 2 tag 0 metric 1
2010/03/10 00:39:47 RIP:   172.30.1.0/24 -> 0.0.0.0 family 2 tag 0 metric 1
2010/03/10 00:39:47 RIP:   192.168.2.8/30 -> 0.0.0.0 family 2 tag 0 metric 1
2010/03/10 00:39:55 RIP: RECV packet from 192.168.2.6 port 520 on eth1
2010/03/10 00:39:55 RIP: RECV RESPONSE version 2 packet size 84
[root@legolas ~]# vtysh

Hello, this is Quagga (version 0.98.6).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

legolas.localdomain# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

R>* 0.0.0.0/0 [120/2] via 192.168.2.1, eth0, 01:19:50
C>* 10.10.10.0/24 is directly connected, eth2
C>* 127.0.0.0/8 is directly connected, lo
K>* 169.254.0.0/16 is directly connected, eth2
R>* 172.30.1.0/24 [120/2] via 192.168.2.1, eth0, 01:19:50
C>* 192.168.2.0/30 is directly connected, eth0
C>* 192.168.2.4/30 is directly connected, eth1
R>* 192.168.2.8/30 [120/2] via 192.168.2.1, eth0, 01:19:50
legolas.localdomain#
```

*With the **default-information originate** command added on Elrond ….*

On Legolas …
RIP updates from Elrond include the default gateway

Legolas adds a default gateway

Lab 4 – Playing with the Beast

Playing #3: Use RIP to set default gateways on Legolas and Arwen

```
[root@legolas ~]# vtysh

Hello, this is Quagga (version 0.98.6).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

legolas.localdomain# debug rip events
legolas.localdomain# debug rip packet
legolas.localdomain# exit
[root@legolas ~]# tail -f /var/log/quagga/ripd.log
2010/03/10 00:48:46 RIP: RECV packet from 192.168.2.1 port 520 on eth0
2010/03/10 00:48:46 RIP: RECV RESPONSE version 2 packet size 44
2010/03/10 00:48:46 RIP:   172.30.1.0/24 -> 0.0.0.0 family 2 tag 0 metric 1
2010/03/10 00:48:46 RIP:   192.168.2.8/30 -> 0.0.0.0 family 2 tag 0 metric 1
2010/03/10 00:49:01 RIP: update timer fire!
[root@legolas ~]# vtysh

Hello, this is Quagga (version 0.98.6).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

legolas.localdomain# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 10.10.10.0/24 is directly connected, eth2
C>* 127.0.0.0/8 is directly connected, lo
K>* 169.254.0.0/16 is directly connected, eth2
R>* 172.30.1.0/24 [120/2] via 192.168.2.1, eth0, 01:29:20
C>* 192.168.2.0/30 is directly connected, eth0
C>* 192.168.2.4/30 is directly connected, eth1
R>* 192.168.2.8/30 [120/2] via 192.168.2.1, eth0, 01:29:20
legolas.localdomain#
```

*Without a **default-information originate** command added on Elrond ….*

On Legolas …
RIP updates from Elrond are limited to connected networks.

Legolas no longer has a default gateway

Housekeeping

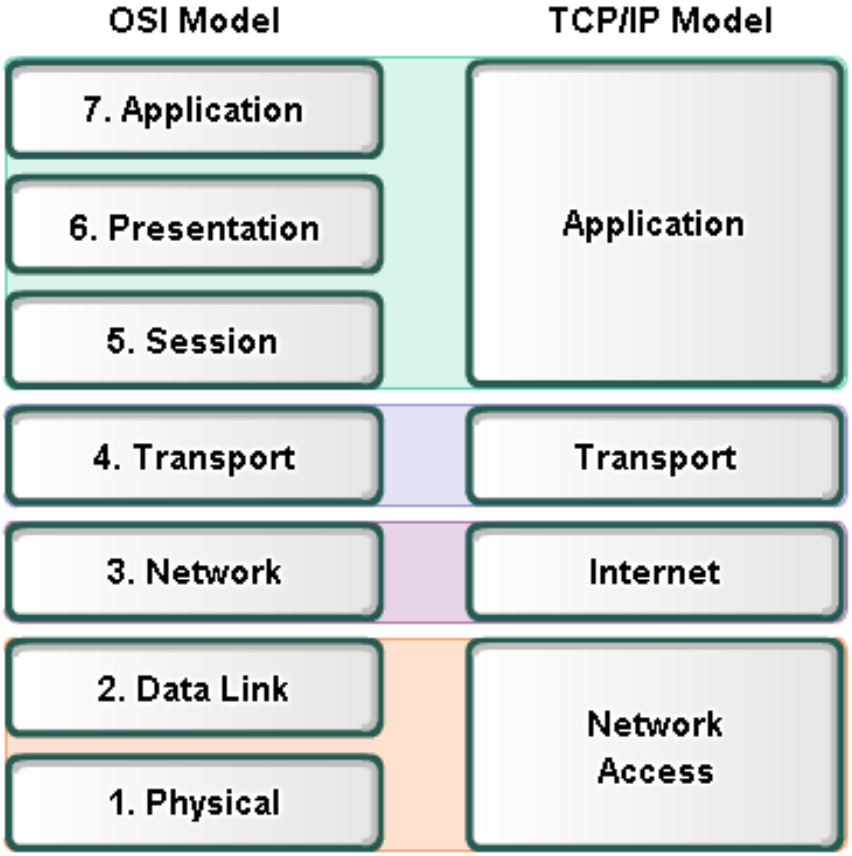
- Nothing due today!
- MSDN AA accounts
- CCC Confer
  - Group chat window test
  - 2nd microphone solution ??
- VTEA/Perkins Survey from Rock
  - Allows Cabrillo to qualify for grant funds
  - Confidential and paper surveys are shredded
  - Not necessary if completed in another class this Spring 2009 semester
  - Must have correct Name, SSN or Student ID and birthday
- Grades page check

Random Review

Layers

Protocol and Reference Models

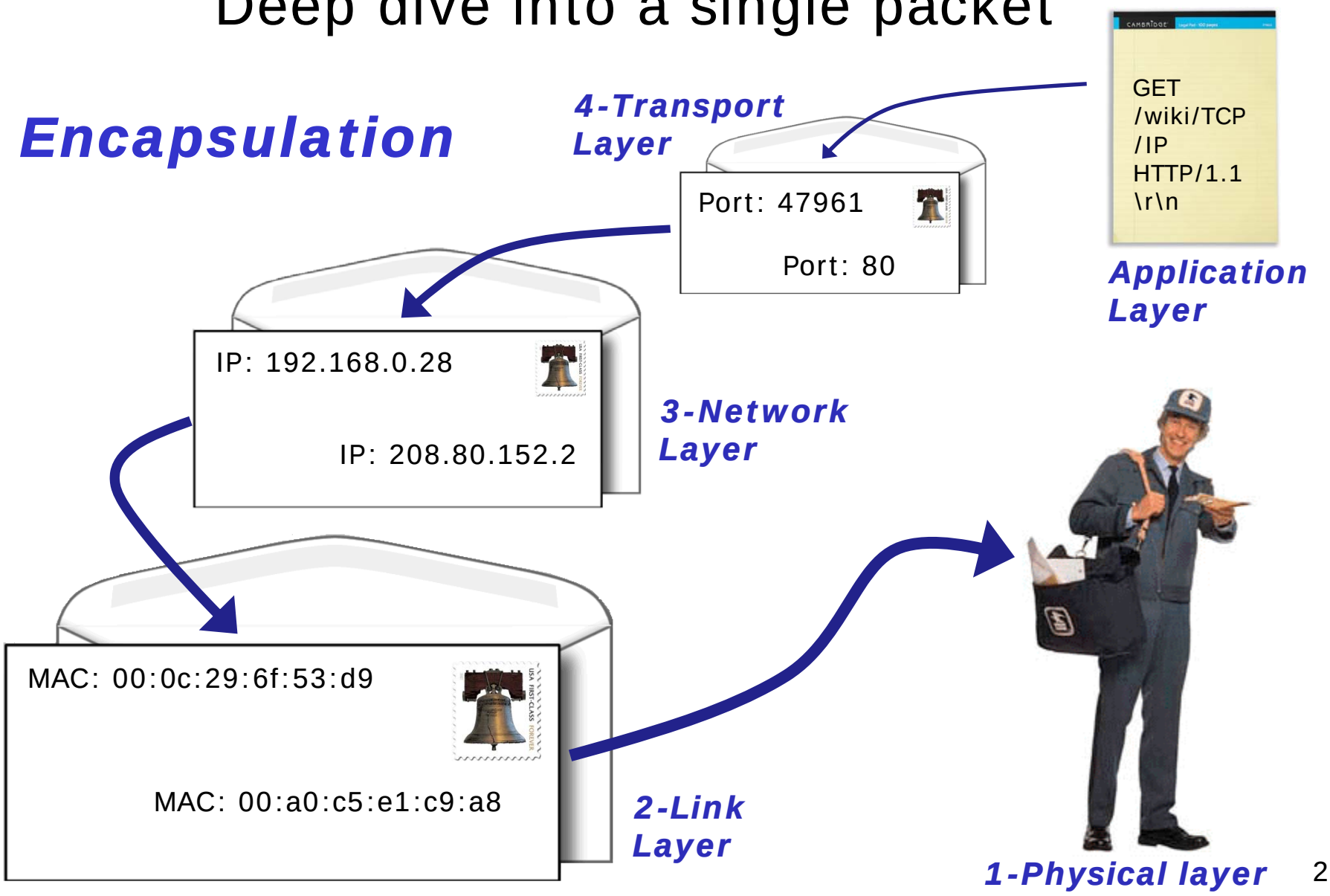
Late 70's, International Organization for Standardization (ISO)



Early 70's, Defense Advanced Research Projects Agency (DARPA)

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

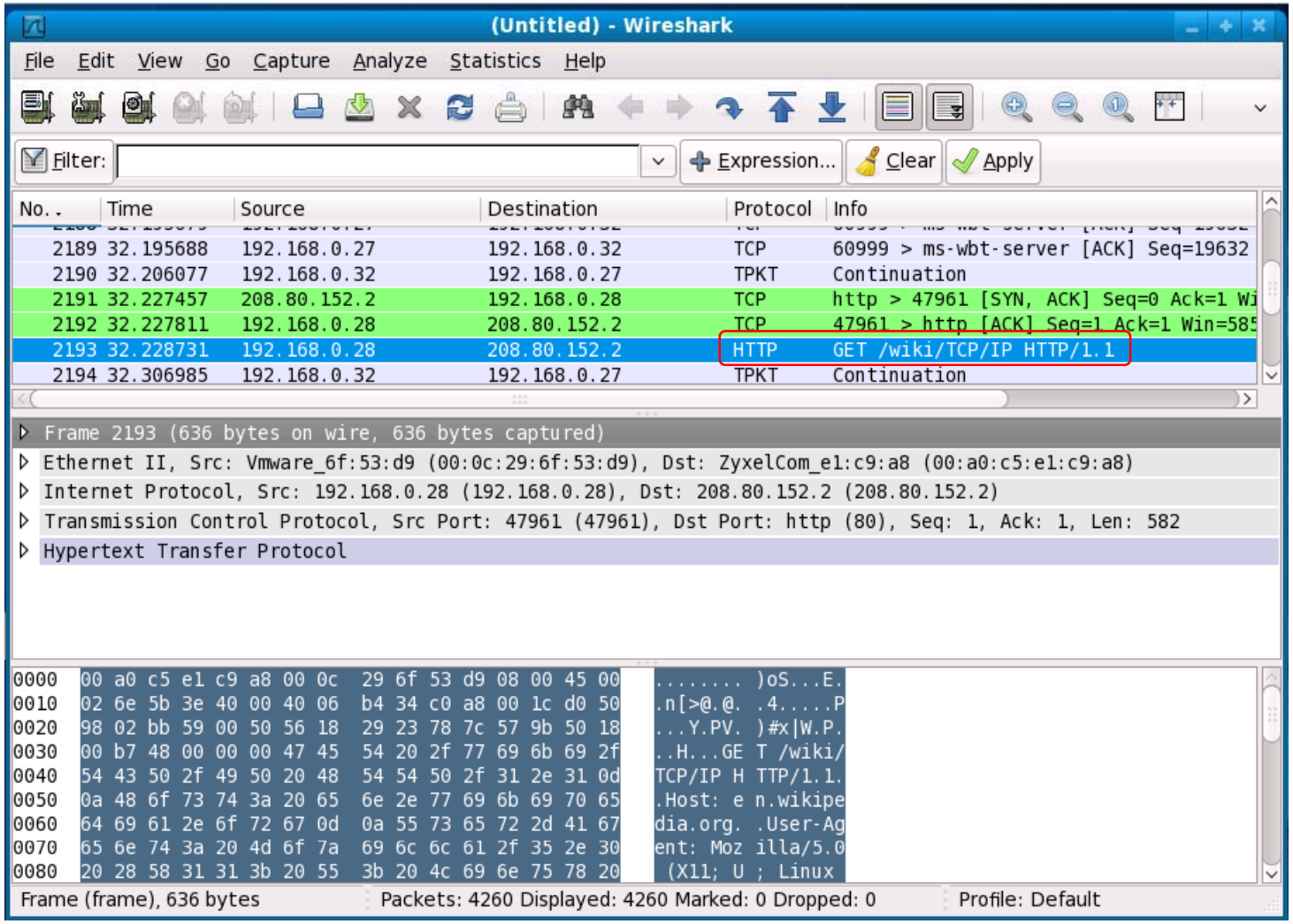
Deep dive into a single packet



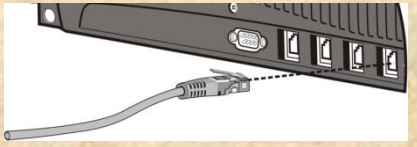
Deep dive into a single packet

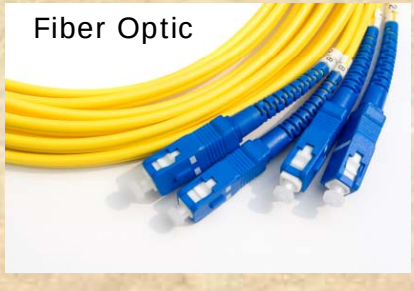
Note how Wireshark shows each layer for the selected HTTP GET packet

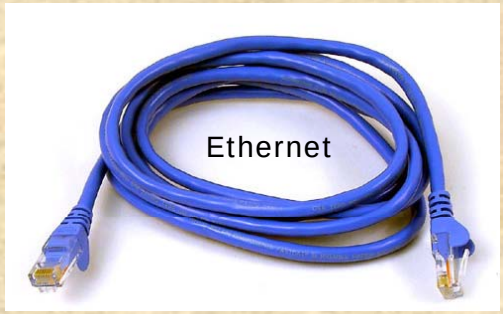
1-Physical
2-Link
3-Network
4-Transport
Application

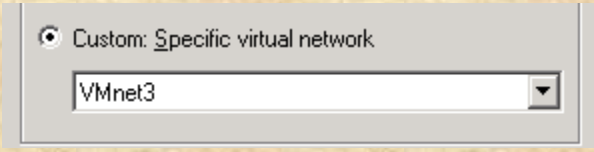


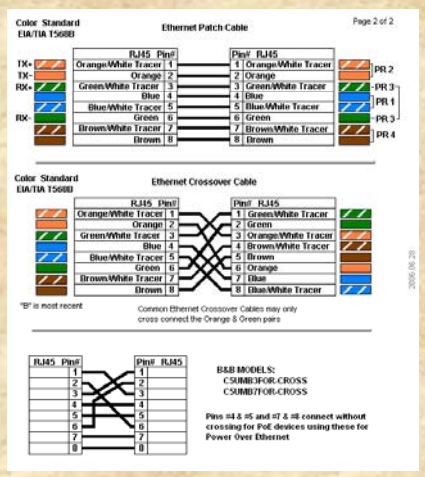
What layer is this?





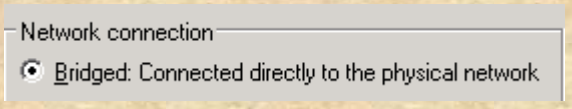






Hub

Fiber to Ethernet adapter



What layer is this?

```
ifconfig eth0 172.30.4.125 netmask 255.255.255.0
```

Header

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)				

```
ping 172.30.1.1
```

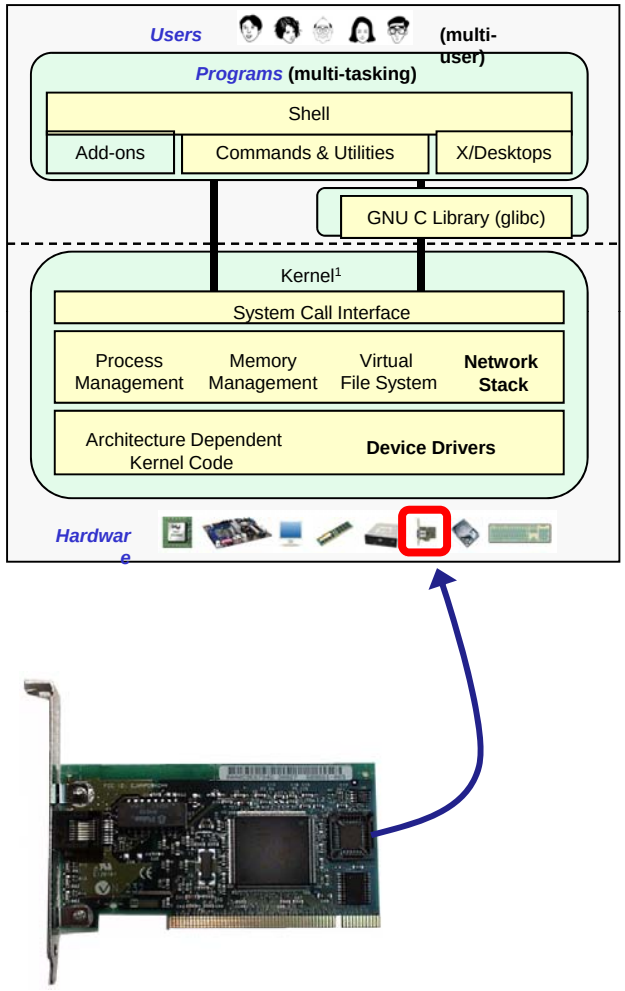
```
route add default gw 172.30.1.1
```

```
Internet Protocol, Src: 172.30.1.150 (172.30.1.150), Dst: 207.211.65.20 (207.211.65.20)
  Version: 4
  Header length: 20 bytes
⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  Total Length: 752
  Identification: 0x6d71 (28017)
⊞ Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 128
  Protocol: TCP (0x06)
⊞ Header checksum: 0xcbfa [correct]
  Source: 172.30.1.150 (172.30.1.150)
  Destination: 207.211.65.20 (207.211.65.20)
```

Wireshark view

NIC/Driver Inventory

NIC Hardware Inventory



- The Network Stack is built into the Linux kernel
- NIC drivers are dynamic kernel modules that can be loaded and unloaded while Linux is running
- Linux systems can have one or more NICs to connect with one or more networks

NIC Hardware Inventory

Use **lspci** to show Ethernet controllers

```
[root@celebrian ~]# lspci
00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)
00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01)
00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)
00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:0f.0 VGA compatible controller: VMware Inc Abstract SVGA II Adapter
00:10.0 SCSI storage controller: LSI Logic / Symbios Logic 53c1030 PCI-X Fusion-MPT Dual
  Ultra320 SCSI (rev 01)
00:11.0 Ethernet controller: Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE] (rev 10)
00:12.0 Ethernet controller: Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE] (rev 10)
[root@celebrian ~]#
```

This Celebrian VM has two AMD 79C970 NICs installed

NIC Hardware Inventory

Use **lspci** to show Ethernet controllers

```
cis192@hershey:~
[cis192@hershey cis192]$ /sbin/lspci
00:00.0 Host bridge: Intel Corp. 82845 845 (Brookdale) Chipset Host Bridge (rev
11)
00:01.0 PCI bridge: Intel Corp. 82845 845 (Brookdale) Chipset AGP Bridge (rev 11
)
00:1e.0 PCI bridge: Intel Corp. 82801BA/CA/DB PCI Bridge (rev 05)
00:1f.0 ISA bridge: Intel Corp. 82801BA ISA Bridge (LPC) (rev 05)
00:1f.1 IDE interface: Intel Corp. 82801BA IDE U100 (rev 05)
00:1f.2 USB Controller: Intel Corp. 82801BA/BAM USB (Hub #1) (rev 05)
00:1f.3 SMBus: Intel Corp. 82801BA/BAM SMBus (rev 05)
00:1f.4 USB Controller: Intel Corp. 82801BA/BAM USB (Hub #2) (rev 05)
02:0c.0 Ethernet controller: Intel Corp. 82557/8/9 [Ethernet Pro 100] (rev 0d)
02:0d.0 Ethernet controller: Intel Corp. 82557/8/9 [Ethernet Pro 100] (rev 0d)
02:0f.0 VGA compatible controller: ATI Technologies Inc Rage XL (rev 27)
[cis192@hershey cis192]$
```

/sbin/lspci

Hershey has two Intel 82557/8/9 Pro 100 NICs installed

NIC Hardware Inventory

Use **lspci** to show Ethernet controllers

Tip: Add /sbin to command if you are not logged in as root

```
[cis192@elrond ~]$ /sbin/lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:02.0 VGA compatible controller: InnoTek Systemberatung GmbH VirtualBox Graphics Adapter
00:03.0 Ethernet controller: Intel Corporation 82543GC Gigabit Ethernet Controller
  (Copper) (rev 02)
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service
00:05.0 Multimedia audio controller: Intel Corporation 82801AA AC'97 Audio Controller (rev
  01)
00:06.0 USB Controller: Apple Computer Inc. KeyLargo/Intrepid USB
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:08.0 Ethernet controller: Intel Corporation 82543GC Gigabit Ethernet Controller
  (Copper) (rev 02)
00:09.0 Ethernet controller: Intel Corporation 82543GC Gigabit Ethernet Controller
  (Copper) (rev 02)
00:0b.0 USB Controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB2 EHCI
  Controller
```

This Elrond VM has three Intel 82543GC NICs installed

NIC Hardware Inventory

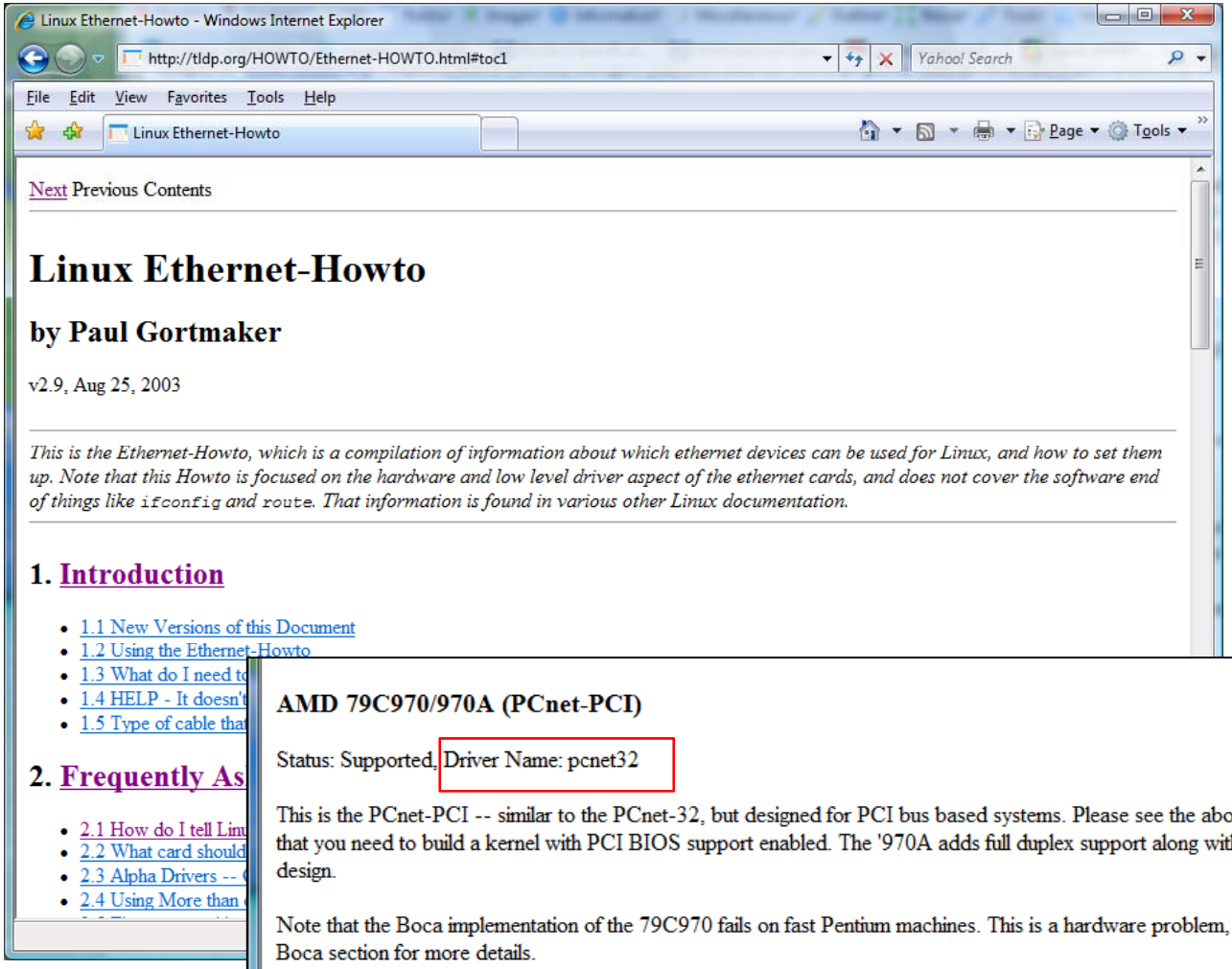
Use **lspci** to show Ethernet controllers

```
[cis192@elrond ~]$ /sbin/lspci | grep Ethernet
00:03.0 Ethernet controller: Intel Corporation 82543GC Gigabit Ethernet Controller
  (Copper) (rev 02)
00:08.0 Ethernet controller: Intel Corporation 82543GC Gigabit Ethernet Controller
  (Copper) (rev 02)
00:09.0 Ethernet controller: Intel Corporation 82543GC Gigabit Ethernet Controller
  (Copper) (rev 02)
[cis192@elrond ~]$
```

*Tip: Pipe **lspci** output into **grep** to make sure you don't miss any controllers which may be buried visually in the output*

NIC Drivers

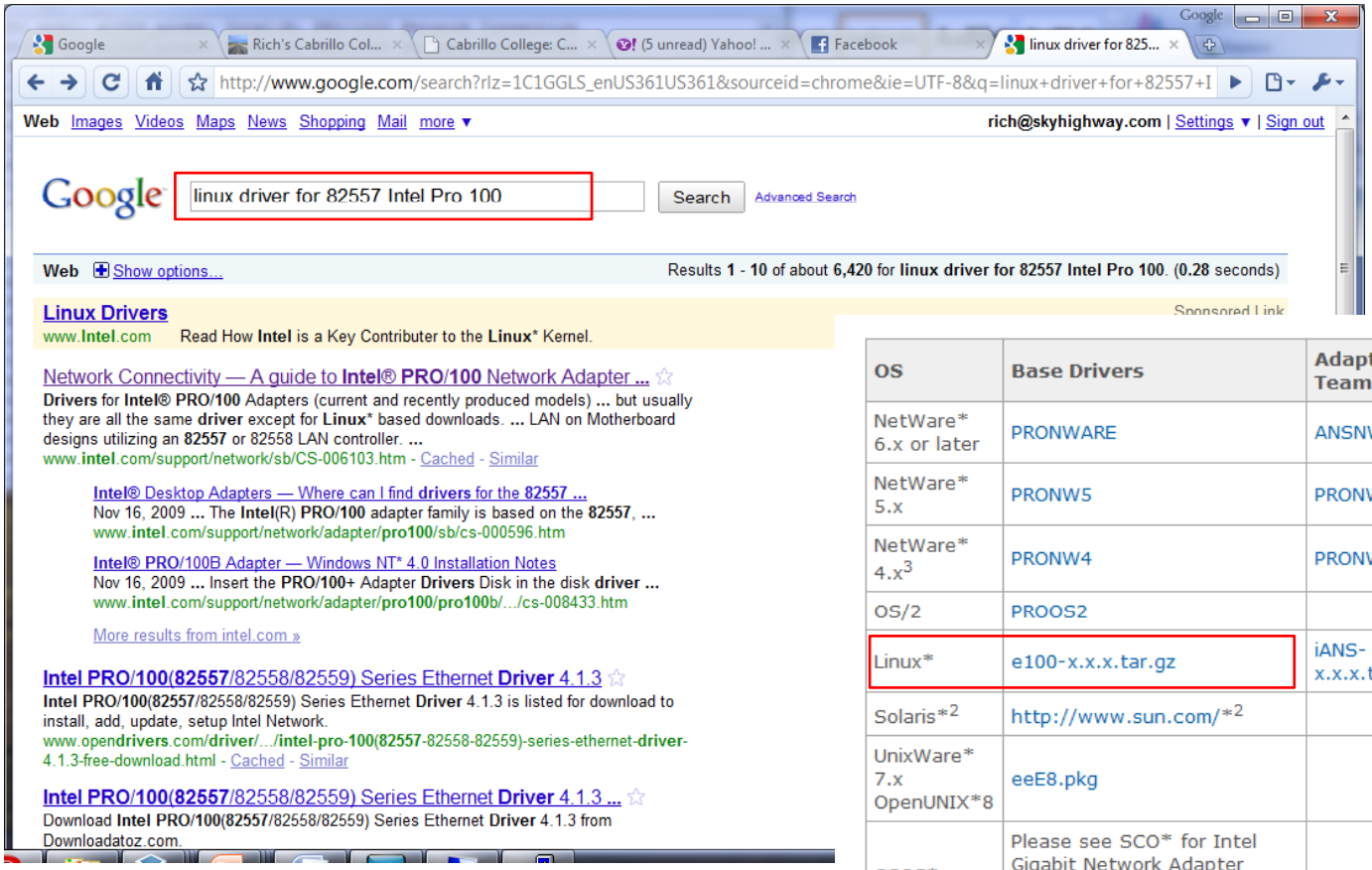
http://tldp.org/HOWTO/Ethernet-HOWTO.html



The AMD 79c970 NIC uses the pcnet32 driver

NIC Drivers

Googling for Linux drivers



This Intel NIC uses the e100 driver found on the Intel web site

OS	Base Drivers	Adapter Teaming	Instrumentation (DMI/SNMP)
NetWare* 6.x or later	PRONWARE	ANSNWARE	
NetWare* 5.x	PRONW5	PRONW5	
NetWare* 4.x[3]	PRONW4	PRONW4	
OS/2	PROOS2		
Linux*	e100-x.x.x.tar.gz	iANS-x.x.x.tar.gz	inic-snmp
Solaris*[2]	http://www.sun.com/*[2]		
UnixWare* 7.x OpenUNIX*8	eeE8.pkg		
SCO5*	Please see SCO* for Intel Gigabit Network Adapter drivers. http://www.sco.com/*		

NIC Hardware Inventory

Use **dmesg** to show NIC/Driver related information

```
[root@celebrian ~]# dmesg | grep eth0
eth0: registered as PCnet/PCI II 79C970A
eth0: link up
eth0: link up
eth0: no IPv6 routers present
[root@celebrian ~]# dmesg | grep net
audit: initializing netlink socket (disabled)
SELinux:  Registering netfilter hooks
Initializing IPsec netlink socket
pcnet32.c:v1.32 18.Mar.2006 tsbogend@alpha.franken.de
pcnet32: PCnet/PCI II 79C970A at 0x1400, 00 0c 29 12 50 1e assigned IRQ 177.
eth0: registered as PCnet/PCI II 79C970A
pcnet32: PCnet/PCI II 79C970A at 0x1480, 00 0c 29 12 50 28 assigned IRQ 185.
eth1: registered as PCnet/PCI II 79C970A
pcnet32: 2 cards_found.
VMware vmxnet virtual NIC driver release 1.0.8 build-126538
[root@celebrian ~]#
```

The Celebrian VM is using the pcnet32 driver for the two AMD 79C970 NICs

NIC Hardware Inventory

Use **dmesg** to show NIC/Driver related information

cis192@hershey:~

```
[cis192@hershey cis192]$ dmesg | grep eth
divert: allocating divert_blk for eth0
e100: eth0: Intel(R) 8255x-based Ethernet Adapter
divert: allocating divert_blk for eth1
e100: eth1: Intel(R) 8255x-based Ethernet Adapter
e100: eth0 NIC Link is Up 100 Mbps Full duplex
e100: eth1 NIC Link is Up 100 Mbps Full duplex
e100: eth0 NIC Link is Up 100 Mbps Full duplex
e100: eth0 NIC Link is Up 100 Mbps Full duplex
e100: eth0 NIC Link is Down
e100: eth0 NIC Link is Up 100 Mbps Full duplex
e100: eth0 NIC Link is Down
e100: eth0 NIC Link is Up 100 Mbps Full duplex
e100: eth0 NIC Link is Up 100 Mbps Full duplex
e100: eth1 NIC Link is Up 100 Mbps Full duplex
[cis192@hershey cis192]$
```

dmesg | grep eth

Hershey is using the e100 driver for the two Intel NICs

NIC Drivers

*NIC drivers are **kernel object modules** and are located in a specific directory so the kernel knows where to find them. Note, they were .o files in older distros.*

```
[root@celebrian ~]# uname -r
2.6.18-92.1.22.el5
[root@celebrian ~]# ls /lib/modules/$(uname -r)/kernel/drivers/net
3c59x.ko      dummy.ko       natsemi.ko      ppp_synctty.ko  sunhme.ko
8139cp.ko     e1000          ne2k-pci.ko     qla3xxx.ko      tg3.ko
8139too.ko    e1000e         netconsole.ko   r8169.ko        tlan.ko
8390.ko       e100.ko        netxen          s2io.ko         tokenring
acenic.ko     epic100.ko     ns83820.ko      sis190.ko       tulip
amd8111e.ko   fealnx.ko      pcmcia          sis900.ko       tun.ko
b44.ko        forcedeth.ko   pcnet32.ko      skge.ko         typhoon.ko
bnx2.ko       ifb.ko         phy             sky2.ko         via-rhine.ko
bnx2x.ko      igb            ppp_async.ko    slhc.ko         via-velocity.ko
bonding       ixgb           ppp_deflate.ko  slip.ko         wireless
cassini.ko    ixgbe          ppp_generic.ko  starfire.ko
chelsio       mii.ko         ppp_mppe.ko     sundance.ko
cxgb3         mlx4           pppoe.ko        sungem.ko
dl2k.ko       myri10ge       pppox.ko        sungem_phy.ko
[root@celebrian ~]#

[root@celebrian ~]# file /lib/modules/$(uname -r)/kernel/drivers/net/e100.ko
/lib/modules/2.6.18-92.1.22.el5/kernel/drivers/net/e100.ko: ELF 32-bit LSB relocatable,
  Intel 80386, version 1 (SYSV), not stripped
[root@celebrian ~]#
```

Commands for handling NIC drivers (kernel modules)

- To show available NIC drivers:
  ls /lib/modules/$(uname -r)/kernel/drivers/net

- To show loaded kernel modules including NIC drivers
  lsmod
  example: **lsmod | grep pcnet32** (show NIC drivers used on VMs)

- To remove (unload) a NIC driver
  rmmod *driver*
  example: **rmmod pcnet32** (removes pcnet32 VM NIC driver)

  Do not specify the path or suffix (.ko) for drivers

- To insert (load) a NIC driver
  modprobe *driver*
  example: **modprobe pcnet32** (installs pcnet32 VM NIC driver)

modprobe is more intelligent and recommended over insmod

1. What are the NICs on the system at 172.30.1.20?
2. What driver is used for those NICs?
3. Where is that driver located?
4. What command could be used to unload that driver?
5. What command could be used to reload that driver?

ifconfig and aliases

Create an Alias IP Address (more than one IP address per interface)

Set

- To set an alias IP address and subnet mask:
  ifconfig ethx:n xxx.xxx.xxx.xxx **netmask** xxx.xxx.xxx.xxx **broadcast** xxx.xxx.xxx.xxx

Verify

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

- To show a single alias interface:
  ifconfig ethx:n

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

Create an Alias IP Address (more than one IP address per interface)

Example setting a temporary IP alias address

```
[root@elrond ~]# ifconfig eth0:1 172.30.4.122 netmask 255.255.255.0 broadcast 172.30.4.255
[root@elrond ~]#
[root@elrond ~]# ifconfig eth0:1
eth0:1    Link encap:Ethernet  HWaddr 00:0C:29:82:68:7A
          inet addr:172.30.4.122  Bcast:172.30.4.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:177 Base address:0x1400

[root@elrond ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:82:68:7A
          inet addr:172.30.4.121  Bcast:172.30.4.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe82:687a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4863 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3442 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:566772 (553.4 KiB)  TX bytes:382355 (373.3 KiB)
          Interrupt:177 Base address:0x1400

[root@elrond ~]#
```

Frodo now can ping either of Elrond's two host IP addresses

```
root@frodo:~# ping -c 1  172.30.1.121
PING 172.30.1.121 (172.30.1.121) 56(84) bytes of data.
64 bytes from 172.30.1.121: icmp_seq=1 ttl=127 time=6.04 ms

--- 172.30.1.121 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 6.049/6.049/6.049/0.000 ms
root@frodo:~# ping -c 1  172.30.1.122
PING 172.30.1.122 (172.30.1.122) 56(84) bytes of data.
64 bytes from 172.30.1.122: icmp_seq=1 ttl=127 time=0.900 ms

--- 172.30.1.122 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.900/0.900/0.900/0.000 ms
root@frodo:~#
```

Create an Alias IP Address
(more than one IP address per interface)

Example setting a permanent IP alias address

Create and edit this file

```
[root@elrond ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0:1
# Intel Corporation 82543GC Gigabit Ethernet Controller (Copper)
DEVICE=eth0:1
ONBOOT=yes
BOOTPROTO=static
IPADDR=172.30.4.122
NETMASK=255.255.255.0

[root@elrond ~]# service network restart
```

For new settings to take effect

Shortcuts

Slash notation and broadcast addresses with ifconfig command

```
[rootelrond ~]# ifconfig eth1:4 10.205.163.203 netmask 255.255.248.0 broadcast
10.205.167.255
[root@elrond ~]# ifconfig eth1:4
eth1:4    Link encap:Ethernet  HWaddr 08:00:27:C1:6B:A5
          inet addr:10.205.163.203  Bcast:10.205.167.255  Mask:255.255.248.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:5 Base address:0xd260
```

```
[root@elrond ~]# ifconfig eth1:4 10.205.163.203 netmask 255.255.248.0
[root@elrond ~]# ifconfig eth1:4
eth1:4    Link encap:Ethernet  HWaddr 08:00:27:C1:6B:A5
          inet addr:10.205.163.203  Bcast:10.205.167.255  Mask:255.255.248.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:5 Base address:0xd260
```

*These three **ifconfig** commands are equivalent*

```
[root@elrond ~]# ifconfig eth1:4 10.205.163.203/21
[root@elrond ~]# ifconfig eth1:4
eth1:4    Link encap:Ethernet  HWaddr 08:00:27:C1:6B:A5
          inet addr:10.205.163.203  Bcast:10.205.167.255  Mask:255.255.248.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:5 Base address:0xd260
```

The slash notation can be used on ifconfig commands. The broadcast address is not needed on newer distros because they don't do classful calculations

Slash notation and broadcast addresses with ifconfig command

Watch out … older distros like Red Hat 9 do classful broadcast calculations!

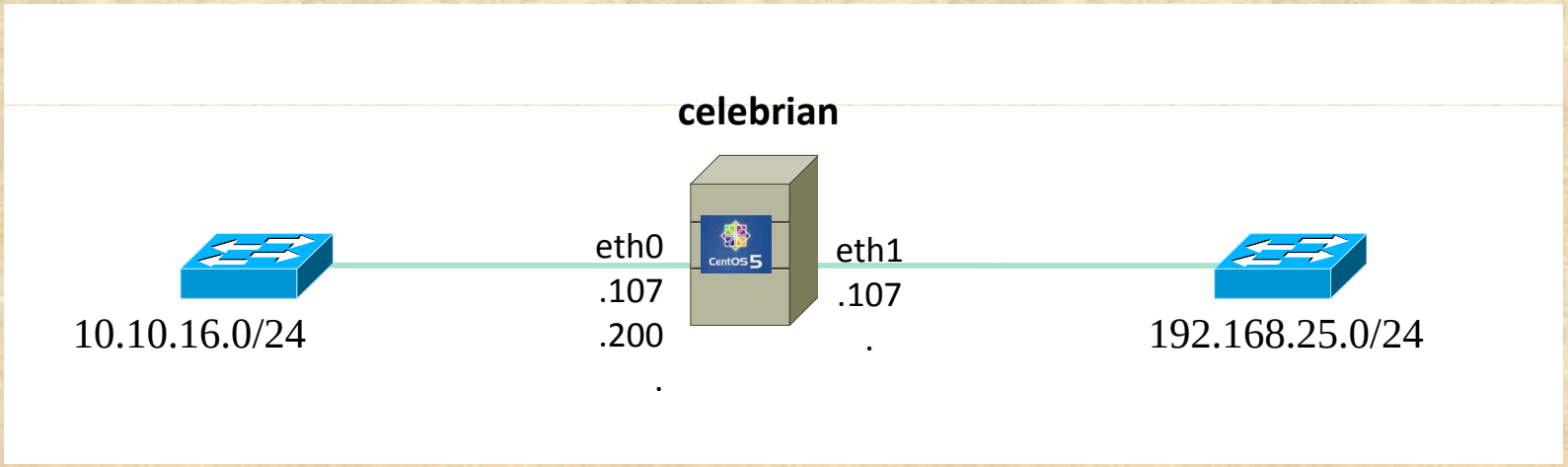
```
[root@nosmo root]# ifconfig eth0 172.30.1.221/24
[root@nosmo root]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:46:7B:2A
          inet addr:172.30.1.221  Bcast:172.30.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:1366 (1.3 Kb)  TX bytes:838 (838.0 b)
          Interrupt:10 Base address:0xd020
```

Using class based addressing, the 172 network is a class B network yielding a 172.30.255.255 broadcast address!

On CentOS 5.4, what broadcast addresses would be calculated using the following commands:

1. **ifconfig eth0 172.30.5.227/24**

2. **ifconfig eth0 172.30.5.227 netmask 255.255.255.0**

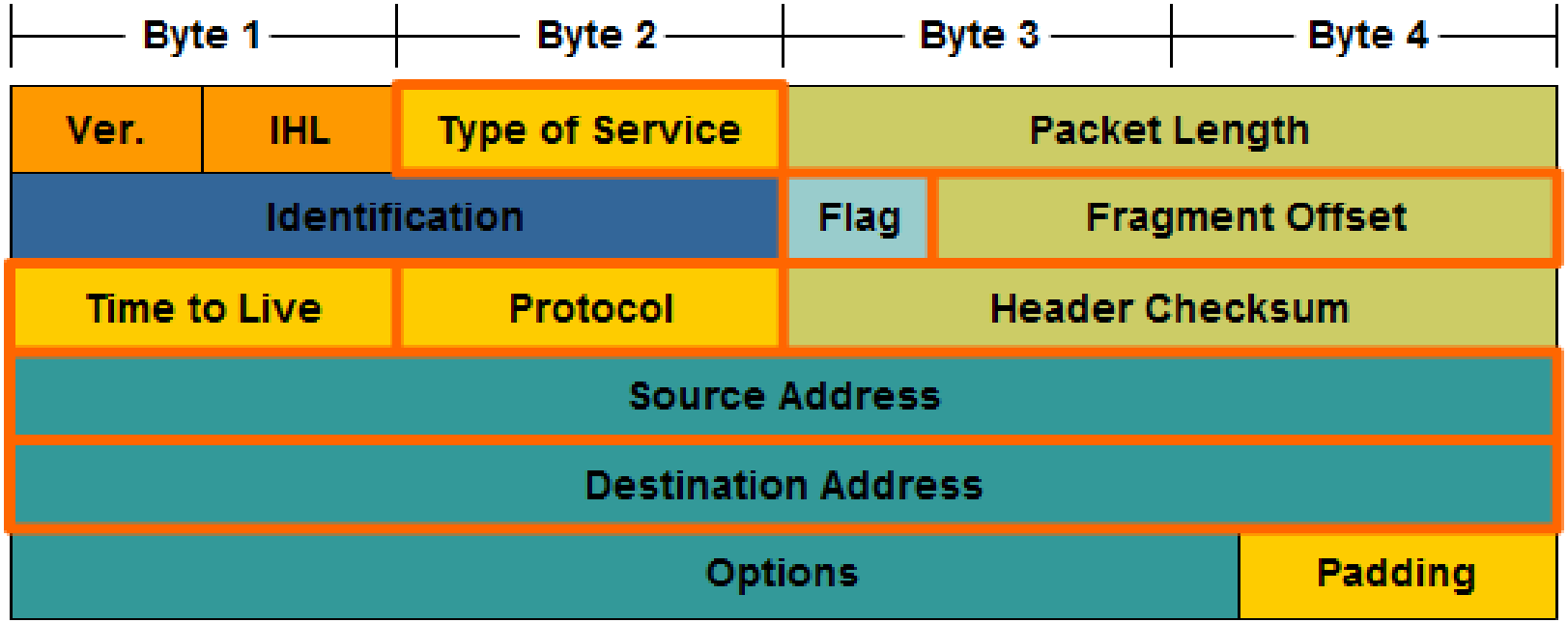


Configure Celebrian as show above then run **ifconfig**

Looking at the **ifconfig** output, what information is not displayed for eth0:1 but is displayed for eth0?

IPv4 Addressing

IPv4 Addresses



- IPv4 addresses are 32 bit addresses

The IPv4 packet header

IPv4 Addressing

An example network, 10.0.0.0/24

```
root@frodo:~# ipcalc 10.0.0.0/24
Address:   10.0.0.0             00001010.00000000.00000000. 00000000
Netmask:   255.255.255.0 = 24   11111111.11111111.11111111. 00000000
Wildcard:  0.0.0.255            00000000.00000000.00000000. 11111111
=>
Network:   10.0.0.0/24          00001010.00000000.00000000. 00000000
HostMin:   10.0.0.1             00001010.00000000.00000000. 00000001
HostMax:   10.0.0.254           00001010.00000000.00000000. 11111110
Broadcast: 10.0.0.255           00001010.00000000.00000000. 11111111
Hosts/Net: 254                   Class A, Private Internet

root@frodo:~#
```

10.0.0.0/24 network

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

Types of Addresses

Network Address

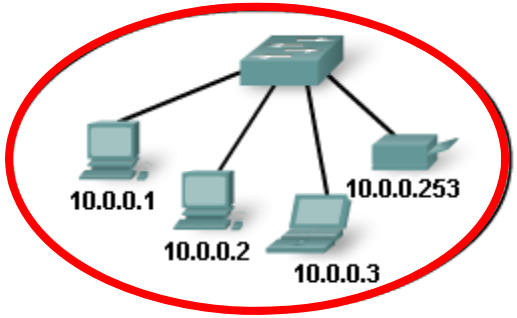
Broadcast Address

Host Address

Roll over to learn more.

	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

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

10.0.0.0/24 network

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

Network Address

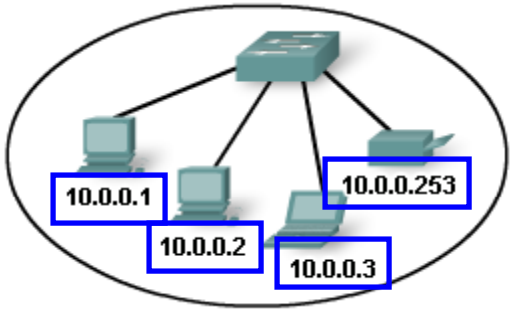
Broadcast Address

Host Address

Roll over to learn more.

	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

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

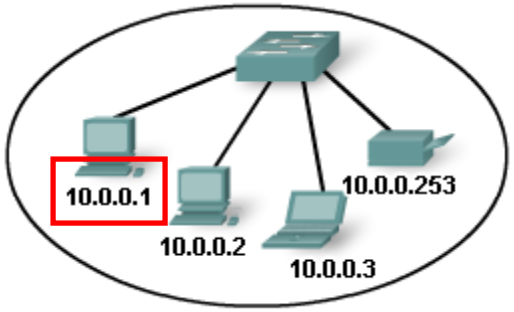
Roll over to learn more.

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

10.0.0.0/24 network

Host Addresses can not 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

ipcalc command

What is the network, broadcast address and netmask for the host address 192.168.2.45/24?

```
[root@bigserver ~]# ipcalc -nbpm 192.168.2.45/24
NETMASK=255.255.255.0
PREFIX=24
BROADCAST=192.168.2.255
NETWORK=192.168.2.0
[root@bigserver ~]#
```

What is the network, broadcast address and netmask for the host address 15.20.34.2/21?

```
[root@bigserver ~]# ipcalc -nbpm 15.20.34.2/21
NETMASK=255.255.248.0
PREFIX=21
BROADCAST=15.20.39.255
NETWORK=15.20.32.0
[root@bigserver ~]#
```

Class Exercise

1. Given the following ip address and network mask, what is the broadcast address?
   IP: 172.30.4.100
   Netmask: 255.255.0.0

2. Given the following ip address and network mask, what is the network address?
   IP: 192.168.30.100
   Netmask: 255.255.240.0

ARP

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.

ARP – Address Resolution Protocol

Overview

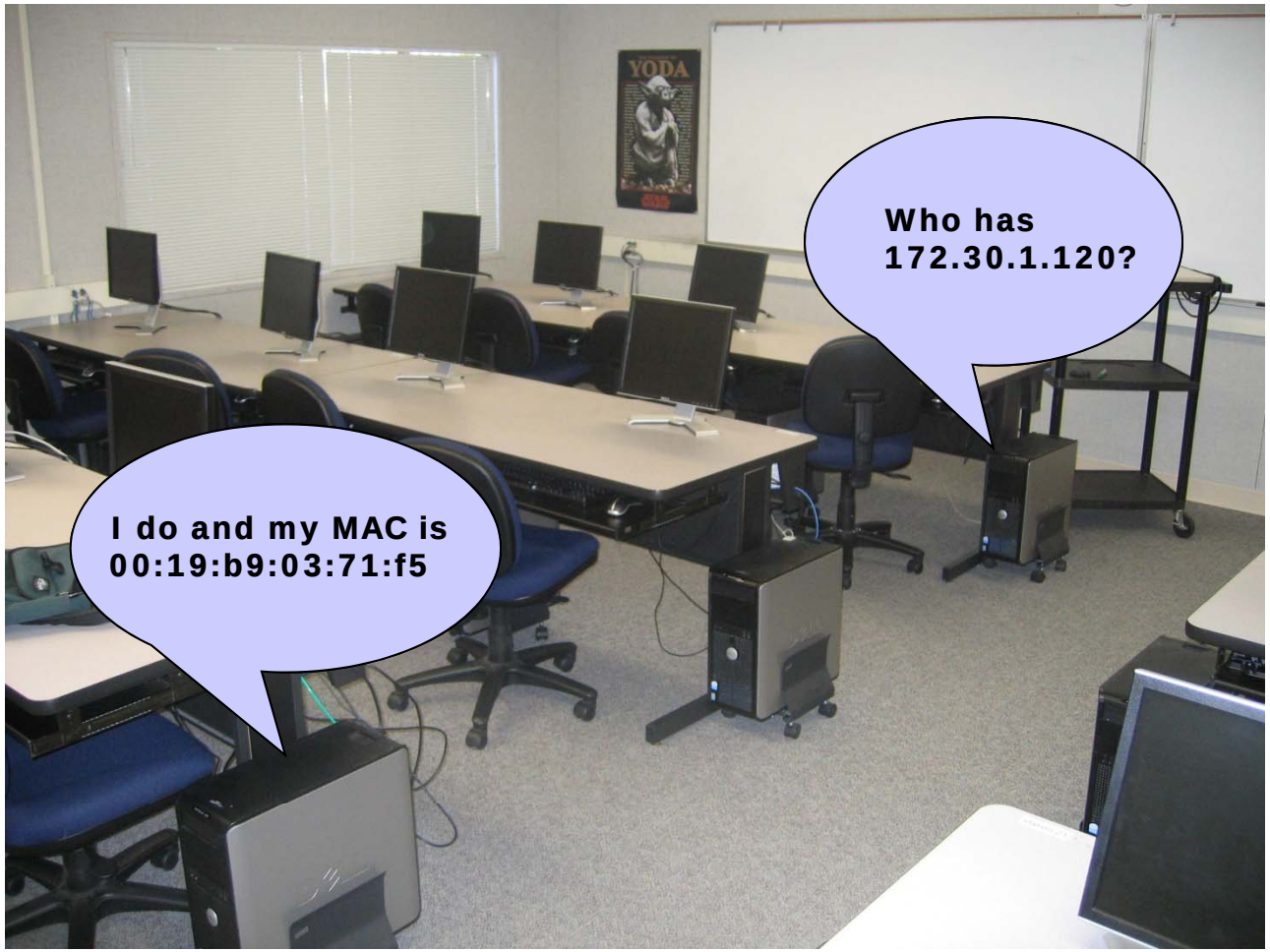
The purpose of ARP is to provide the correct destination physical address given the destination IP address.

- RFC 826 (http://tools.ietf.org/html/rfc826)
- Part of IPv4 (IPv6 uses Neighbor Discovery ND)
- The ARP request: broadcasts a "Who has this IP address?"
- The ARP replay: targeted to the requestor's address (unicast) – "I do and my MAC address is *xx:xx:xx:xx:xx:xx*"

ARP – Address Resolution Protocol

Overview

Station04 wants to ping Station20



ARP Example (1 of 3)

Address resolution commands

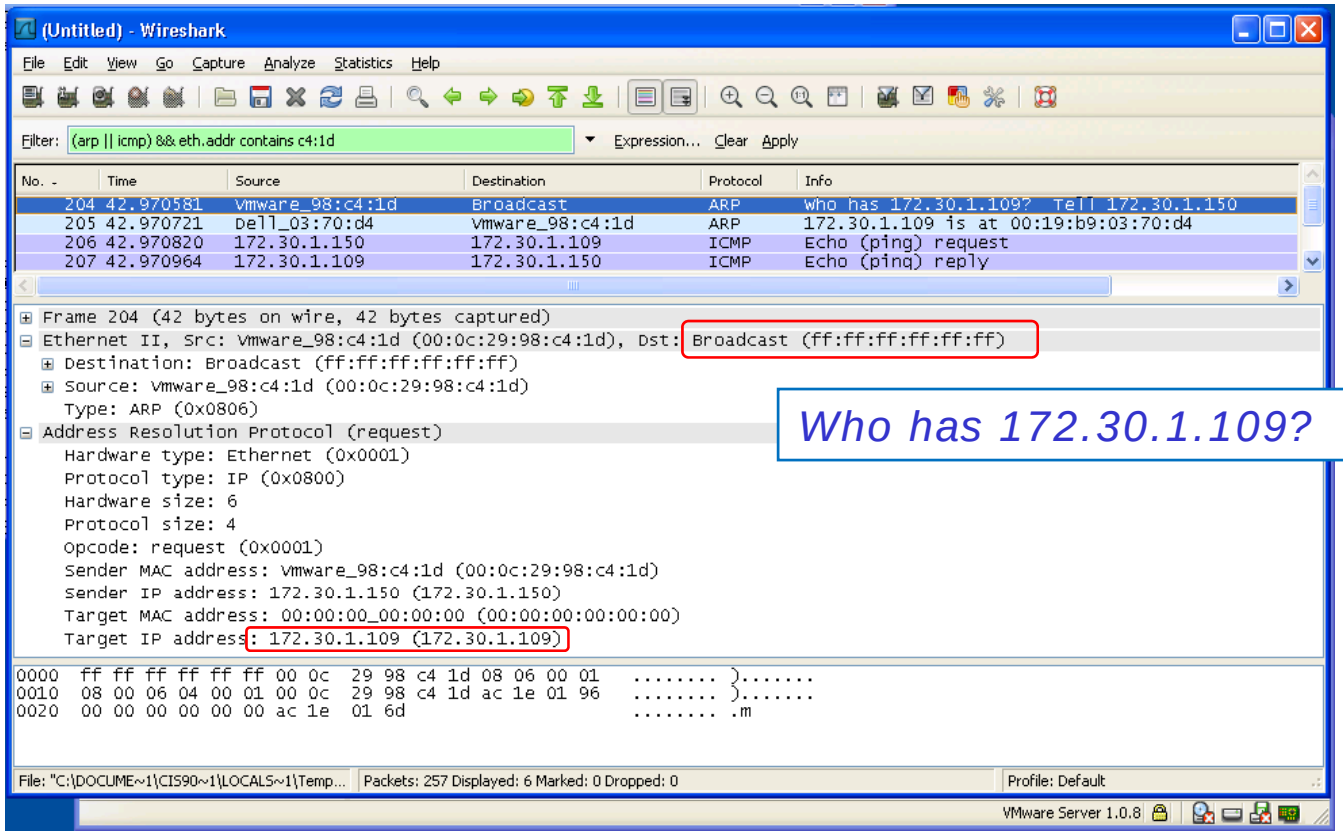
From Frodo, ping 172.30.1.109 which results in a ARP request and new arp cache entry

```
root@frodo:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0c:29:98:c4:1d
      inet addr:172.30.1.150  Bcast:172.30.1.255  Mask:255.255.255.0
< snipped>

root@frodo:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.30.1.106             ether   00:18:8b:28:ac:90   C                     eth0
172.30.1.101             ether   00:19:b9:03:71:07   C                     eth0
172.30.1.1               ether   00:b0:64:53:42:01   C                     eth0

root@frodo:~# ping -c 1 172.30.1.109
PING 172.30.1.109 (172.30.1.109) 56(84) bytes of data.
64 bytes from 172.30.1.109: icmp_seq=1 ttl=128 time=3.71 ms
< snipped >

root@frodo:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.30.1.109             ether   00:19:b9:03:70:d4   C                     eth0
172.30.1.106             ether   00:18:8b:28:ac:90   C                     eth0
172.30.1.101             ether   00:19:b9:03:71:07   C                     eth0
172.30.1.1               ether   00:b0:64:53:42:01   C                     eth0
```

ARP Example (2 of 3)

Address resolution example - request

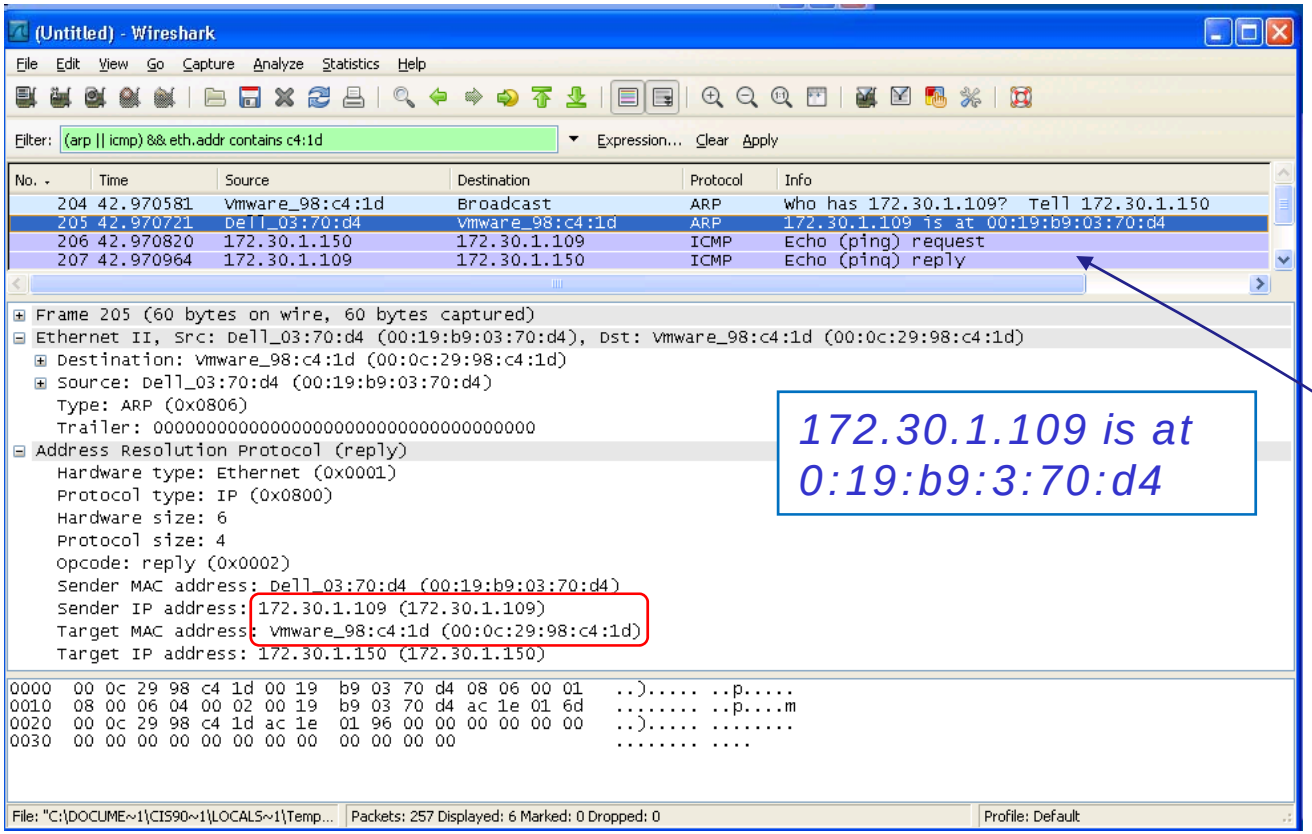
Before Frodo can ping 172.30.1.109, it needs the destination MAC address to finish making the Ethernet frame for the ping request.



Frodo's ARP request is a broadcast. Every NIC on the subnet will hear it and check to see if the requested IP address belongs to them.

ARP Example (3 of 3)

Address resolution example - reply



172.30.1.109 is at 0:19:b9:3:70:d4

Station09's ARP reply sent as a unicast directly back to Frodo.

Now Frodo knows the destination MAC address and can start pinging

ARP – Address Resolution Protocol

Overview

Managing the ARP cache

- Complete (C) 0x02

  Temporary arp cached entries are aged out after several minutes

- Permanent (M) 0x04

  Till next system restart

- Published (P) 0x08

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

ARP – Address Resolution Protocol

Overview

```
root@frodo:~# ping 172.30.1.222
PING 172.30.1.222 (172.30.1.222) 56(84) bytes of data.
From 172.30.1.151 icmp_seq=2 Destination Host Unreachable
From 172.30.1.151 icmp_seq=3 Destination Host Unreachable
From 172.30.1.151 icmp_seq=4 Destination Host Unreachable
^C
--- 172.30.1.222 ping statistics ---
6 packets transmitted, 0 received, +3 errors, 100% packet loss, time 5006ms
, pipe 3
root@frodo:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.30.1.222                     (incomplete)                              eth0
172.30.1.125             ether   08:00:27:a6:20:b6   C                     eth0
root@frodo:~#
```

Pinging a non-existing system or a system that is powered off results in a failed arp request and an incomplete arp cache entry

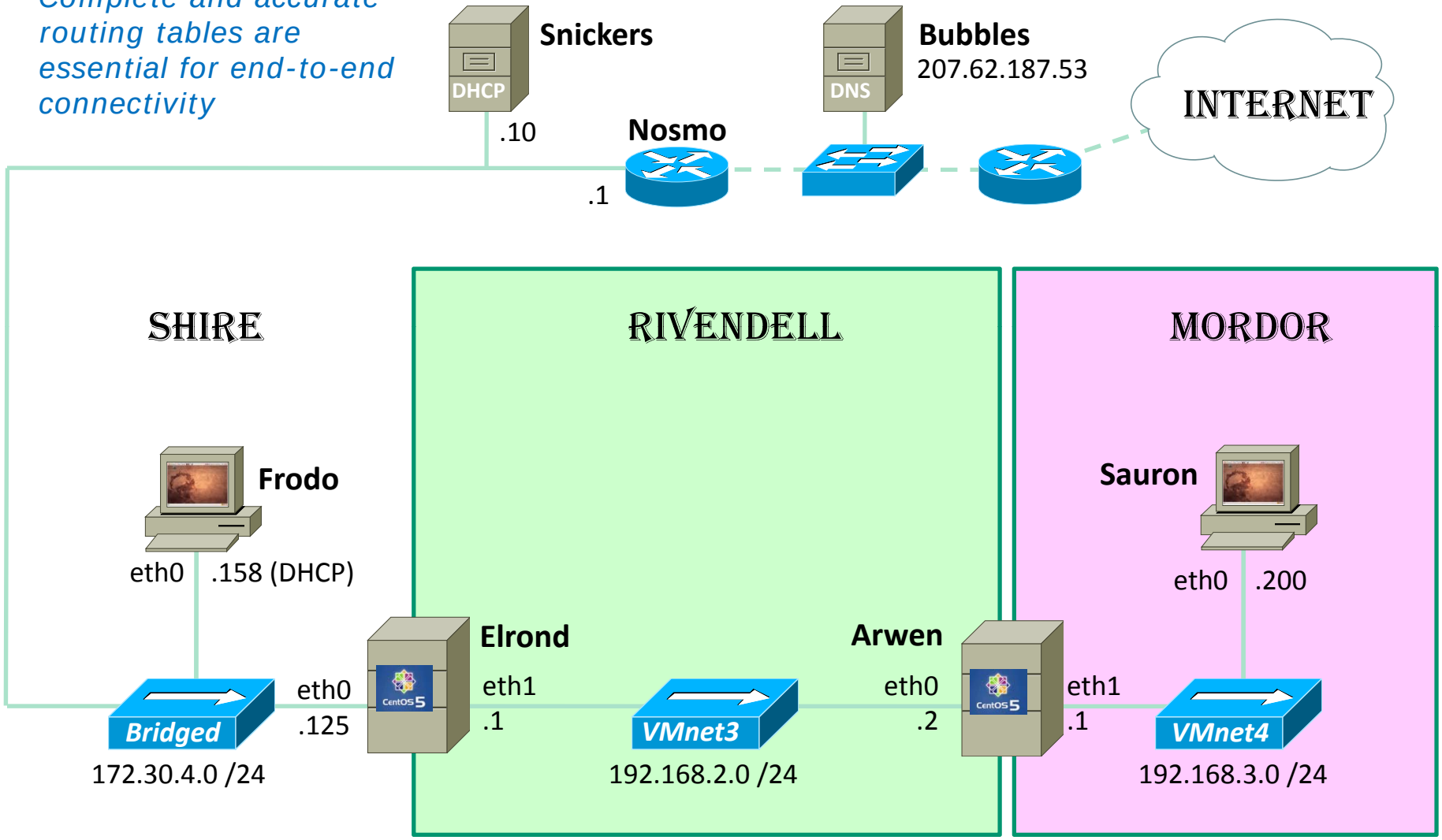
Class Exercise

1. What is the MAC address on the Nosmo router (a Cisco 2621) at 172.30.1.1?

2. Who wrote RFC 826?

Routing

Complete and accurate routing tables are essential for end-to-end connectivity



Frodo
Default GW > 172.30.4.1
192.168.2.0/24> 172.30.4.125
192.168.3.0/24 > 172.30.4.125

Elrond
Default GW > 172.30.4.1
192.168.3.0/24 > 192.168.2.2
forwarding on

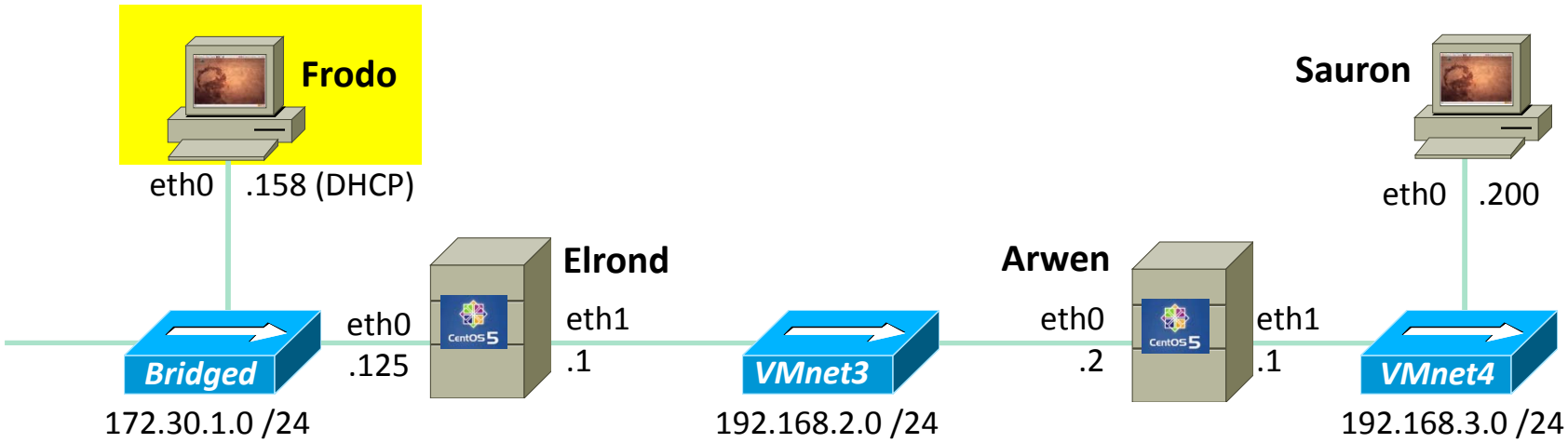
Arwen
Default GW > 192.168.2.1
forwarding on

Sauron
Default GW > 192.168.3.1

The Routing Table

This is the routing table on Frodo for Lab 3

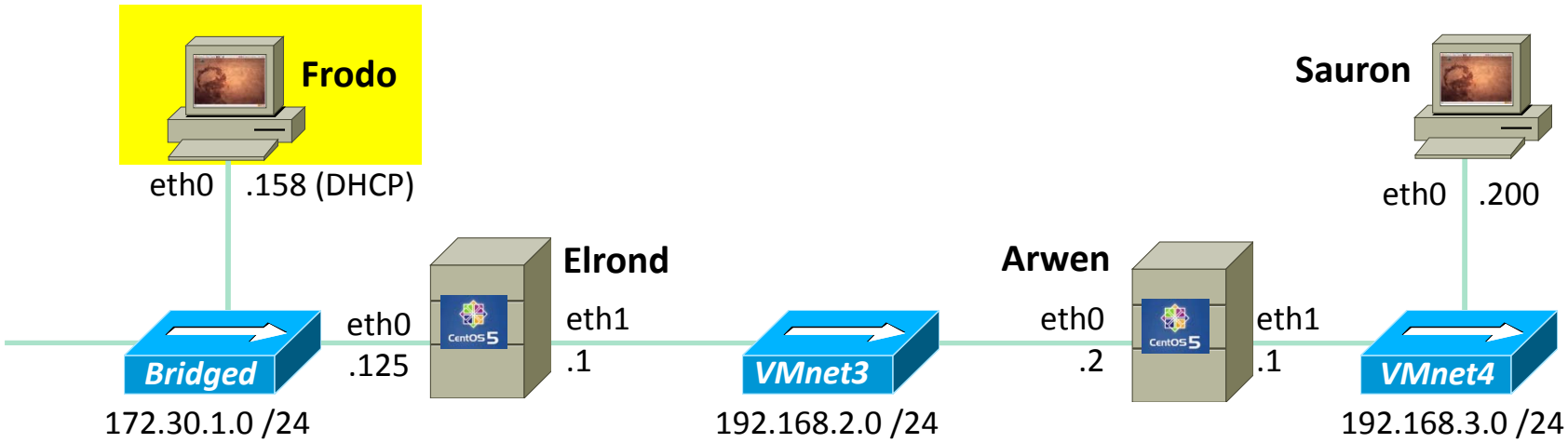
```
root@frodo:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.3.0     172.30.1.125    255.255.255.0   UG    0      0        0 eth0
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.2.0     172.30.1.125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



The Routing Table

Directly connected networks

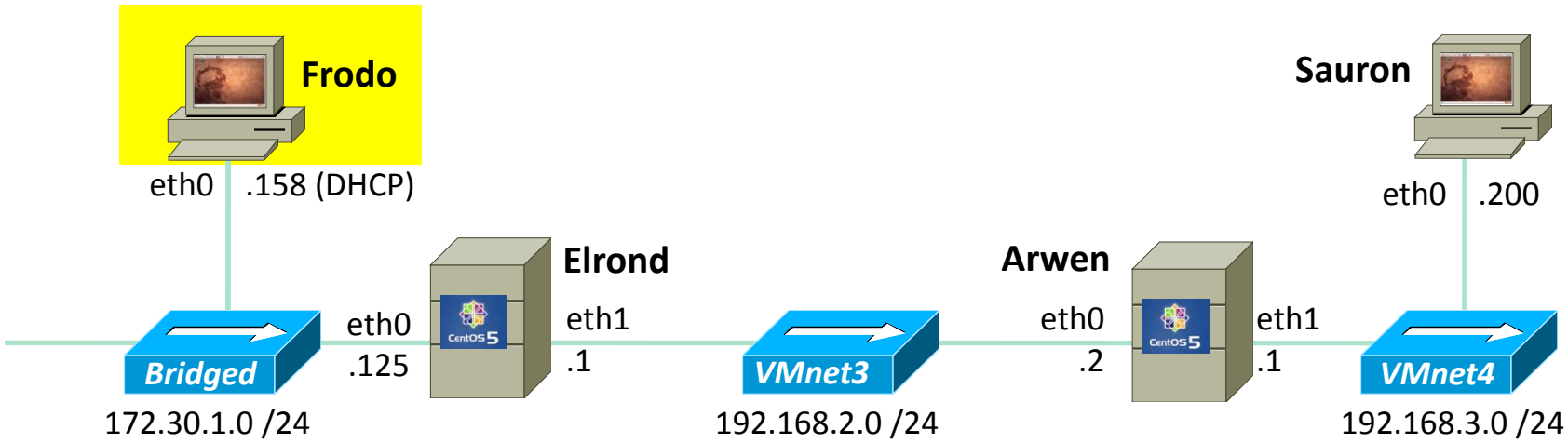
```
root@frodo:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.3.0     172.30.1.125    255.255.255.0   UG    0      0        0 eth0
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.2.0     172.30.1.125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



The Routing Table

Non-directly connected networks that are reachable via gateways (routers)

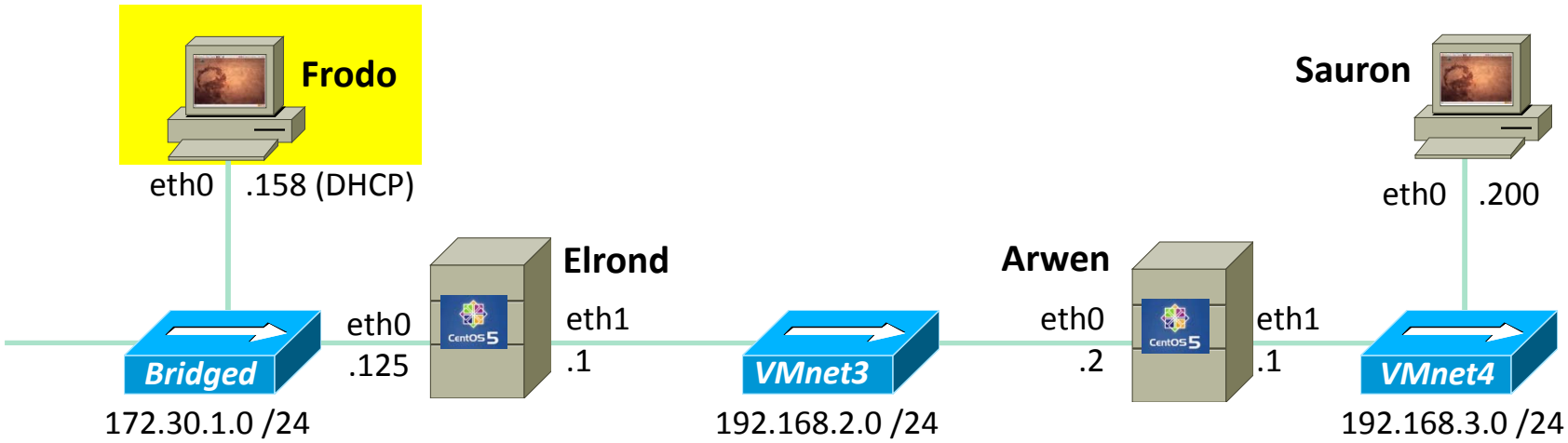
```
root@frodo:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.3.0     172.30.1.125    255.255.255.0   UG    0      0        0 eth0
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.2.0     172.30.1.125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



The Routing Table

The default gateway matches all networks

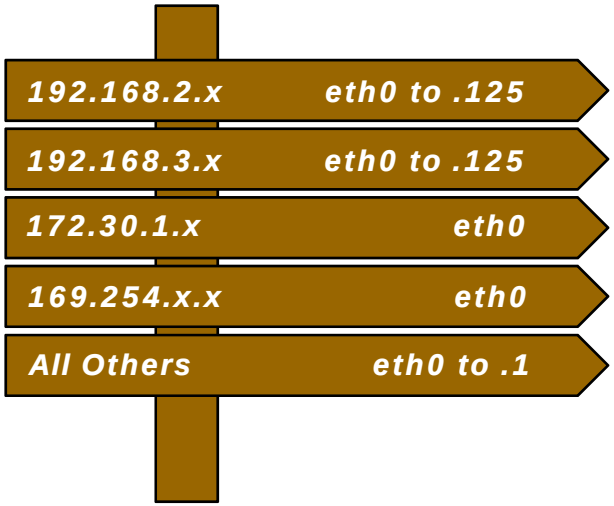
```
root@frodo:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.3.0     172.30.1.125    255.255.255.0   UG    0      0        0 eth0
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.2.0     172.30.1.125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



The Routing Table

The routing table is a sign post

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

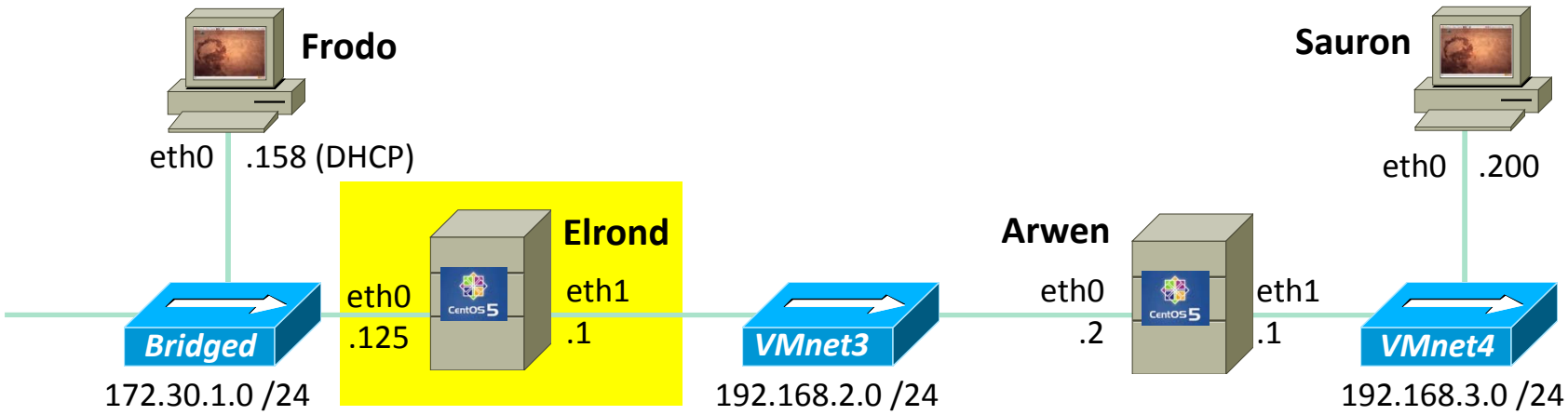


In this case, all packets are forwarded out eth0. Each IP packet is placed in an Ethernet frame. The destination MAC address for the frame will be either the MAC address of the destination device if it is on a connected network, or the MAC address of the appropriate gateway.

The Routing Table

This is the routing table on Elrond for Lab 3

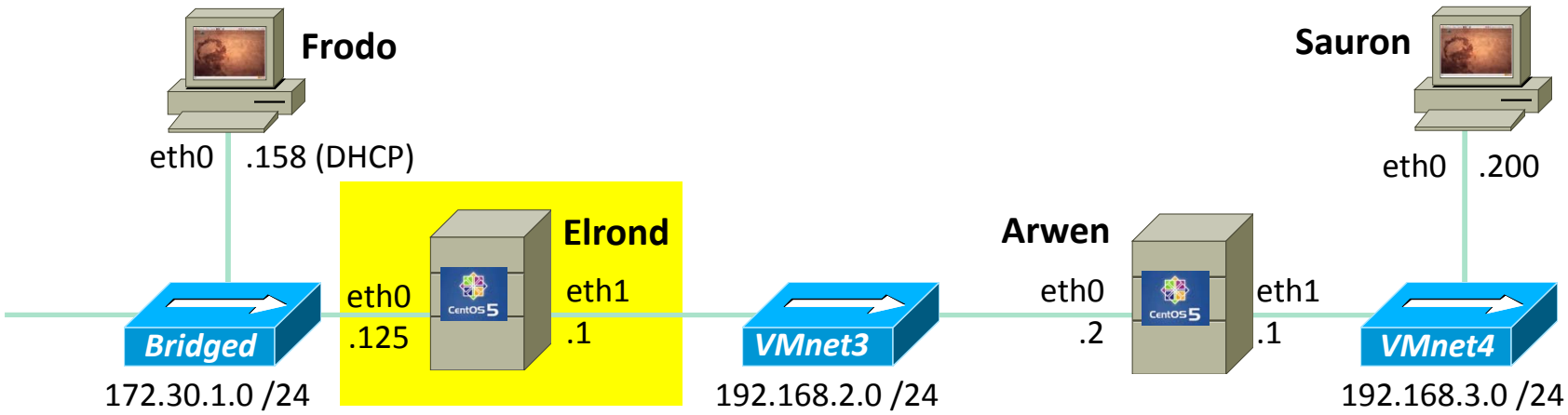
```
[root@elrond bin]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.2     255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
0.0.0.0         172.30.1.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond bin]#
```



The Routing Table

Directly connected networks

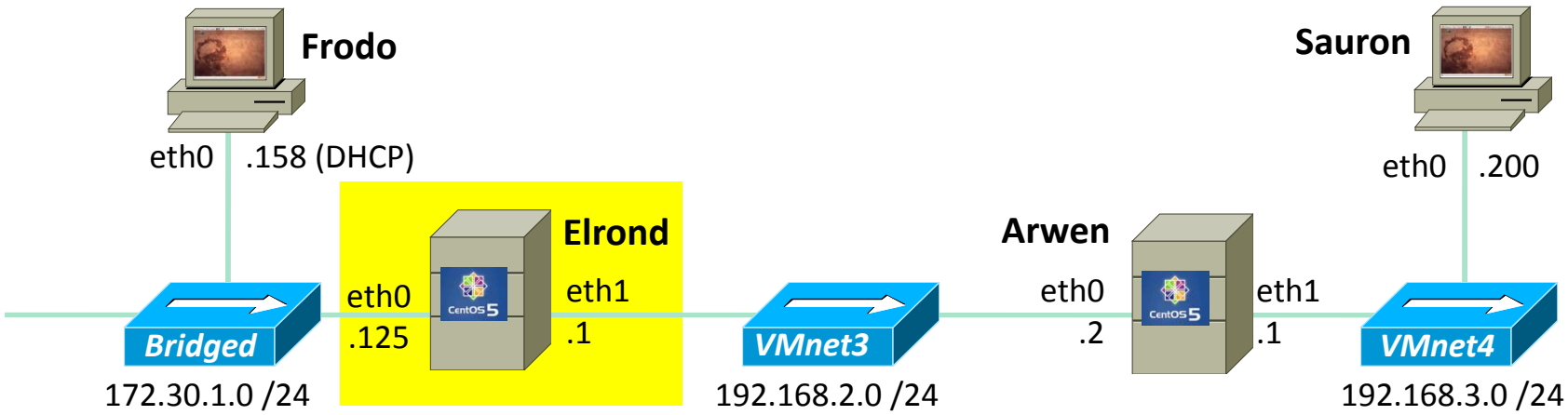
```
[root@elrond bin]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.2     255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
0.0.0.0         172.30.1.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond bin]#
```



The Routing Table

Non-directly connected networks that are reachable via gateways (routers)

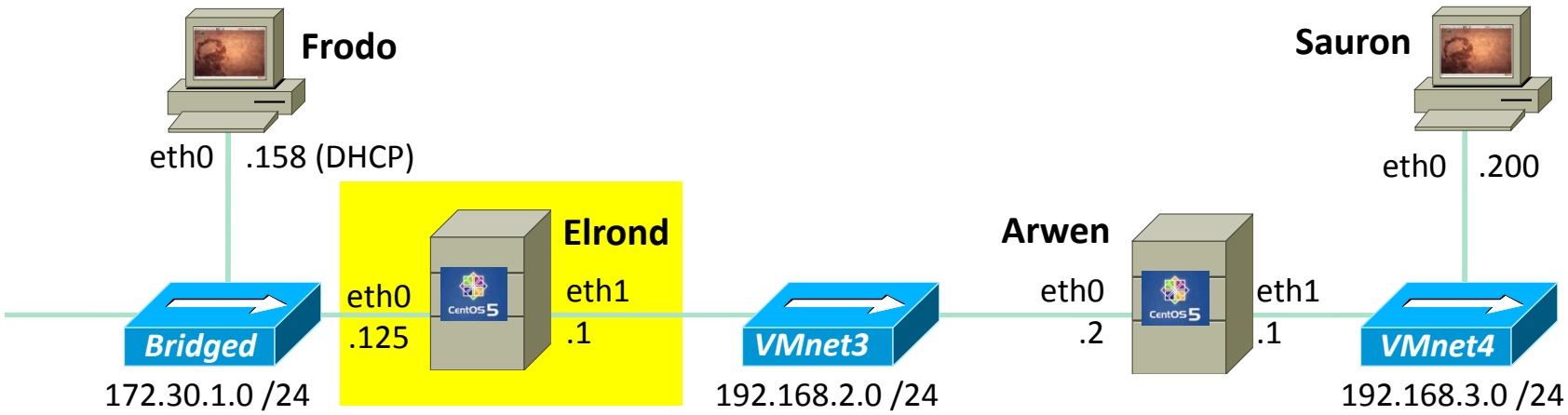
```
[root@elrond bin]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.2     255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
0.0.0.0         172.30.1.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond bin]#
```



The Routing Table

The default gateway matches all networks

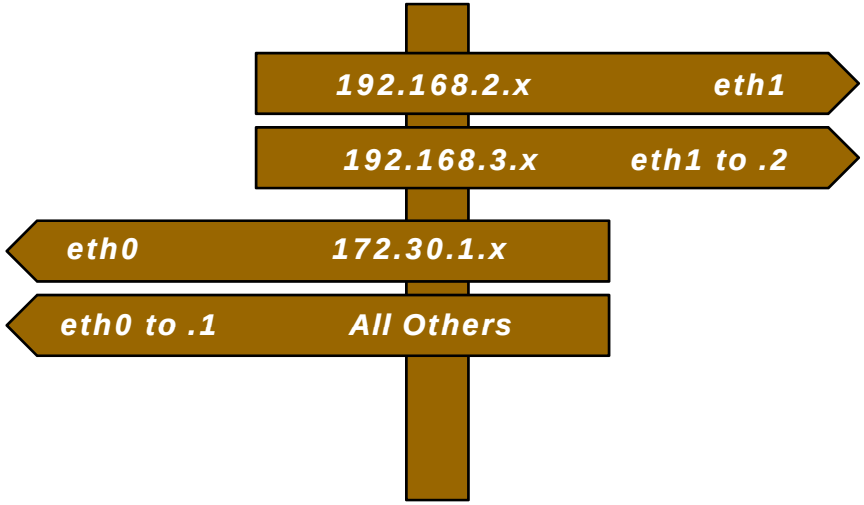
```
[root@elrond bin]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.2     255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
0.0.0.0         172.30.1.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond bin]#
```



The Routing Table

The routing table is a sign post

```
[root@elrond bin]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.2     255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
0.0.0.0         172.30.1.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond bin]#
```



In this case, all packets are forwarded out either eth0 or eth1. Each IP packet is placed in an Ethernet frame. The destination MAC address for the frame will be either the MAC address of the destination device if it is on a connected network, or the MAC address of the appropriate gateway.

The Routing Table
Supernetting

Routing Table

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

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

The Routing Algorithm

(How the decision is made)

Routing Algorithm

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

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

route command

show route table with names

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

show route table with IP addresses

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

route command
flushing the cache

Flush the route cache

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

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

```
[root@elrond bin]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.3.0     192.168.2.2     255.255.255.0   UG    0      0        0 eth1
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
0.0.0.0         172.30.1.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond bin]#
```

1. What happens with an IP packet having a destination address of 207.62.187.53?
2. What happens with an IP packet having a destination address of 192.168.2.15?
3. What happens with an IP packet having a destination address of 172.30.4.1?
4. What networks are directly connected to this Linux router?
5. Are gateway addresses on directly connected networks, non-directly connected networks or both?
6. When creating an Ethernet frame for an IP packet with destination address 192.168.3.99, what destination MAC address is used?
7. What is the default gateway?

```
[root@arwen ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.2.8     0.0.0.0         255.255.255.252 U     0      0        0 eth0
192.168.2.0     192.168.2.5     255.255.255.252 UG    2      0        0 eth1
192.168.2.4     0.0.0.0         255.255.255.252 U     0      0        0 eth1
172.30.1.0      192.168.2.10    255.255.255.0   UG    2      0        0 eth0
10.10.10.0      192.168.2.5     255.255.255.0   UG    2      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
[root@arwen ~]#
```

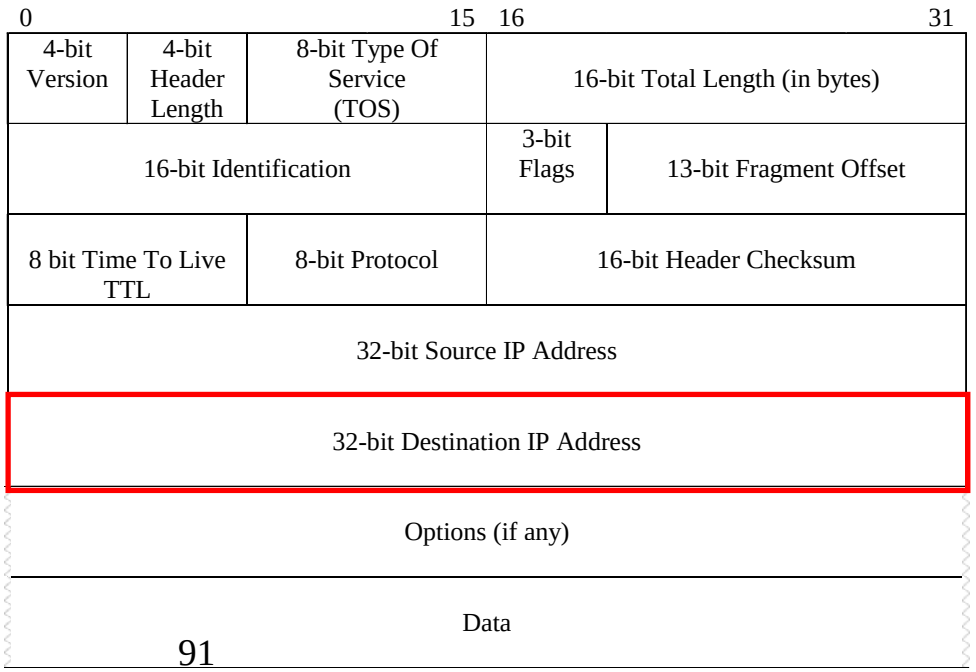
1. What does the “G” mean under flags?
2. What is the default gateway?
3. What happens with a packet with destination IP address 207.62.186.9?
4. What networks are not directly connected?
5. In what order are potential routes tested for a match?
6. When creating an Ethernet frame for an IP packet with destination address 10.10.10.200, what destination MAC address is used?
7. When configuring a static route , should the gateway be on a connected or non-connected network?

Dynamic Routing Protocols

Routed Protocol

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

IP Header



Rick Graziani
graziani@cabrillo.edu

Routing Protocols

*After doing lab 3 can you imagine **manually** setting up and maintaining static routes on dozens or evens hundreds of routers!*

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

These are major routing protocols you will learn to implement in the Cabrillo routing and advanced routing classes.

*These protocols allow routers to talk to each other and **automatically** configure the routing tables with remote network routes*

Routing Types

- **Directly connected networks** are networks that the router is connected to, has an IP address/mask.
- **Non-directly connected networks** are remote networks connected to other routers.

*A router must learn about non-directly connected networks either **statically** or **dynamically**.*

Static

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

Dynamic

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

BGP
ARP
OSPF
RIP
IPv4
IPv6
Ethernet
DHCP

1. Which of the protocols above are routable?

2. Which are routing protocols?

3. Which are neither routing protocols or routable?

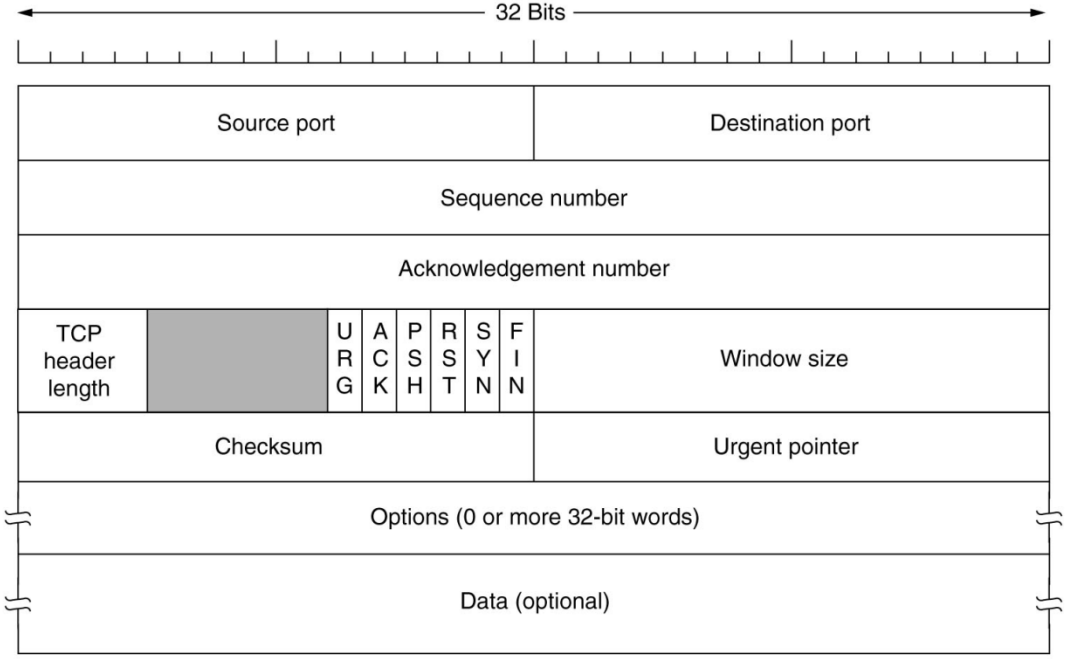
End of Review

Transmission Control Protocol

Transport Layer

The Transmission Control Protocol

TCP Header



Sequence and acknowledgement numbers are used for flow control.

ACK, SYN and FIN flags are used for initiating connections, acknowledging data received and terminating connections

Window size is use to communicate buffer size of recipient.

Options like SACK permit selective acknowledgement

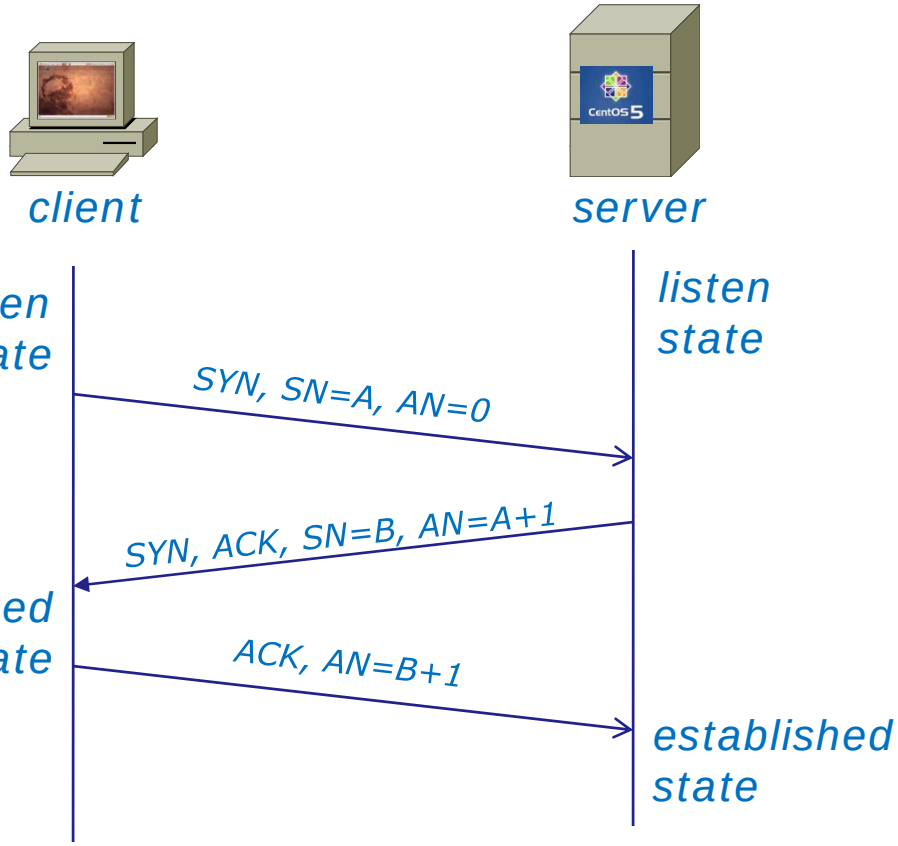
Transport Layer

The Transmission Control Protocol

Initial Connection

Three-Way Handshake

1. SYN
2. SYN-ACK
3. ACK



AN=Acknowledgment Number
SN=Sequence Number
ACK=ACK flag set

Transport Layer

Sockets

Sockets are communication endpoints which define a network connection between two computers (RFC 793).

- Source IP address
- Source port number
- Destination IP address
- Destination port number

The socket is associated to a port number so that the TCP layer can identify the application to send data to.

Application programs can read and write to a socket just like they do with files.

Transport Layer

The Transmission Control Protocol

Continuing Communications

- o The Sliding Window

  Used for flow control - allows sending additional segments before an acknowledgement is received based on recipients buffer size

- o Flow Control (cumulative acknowledgment)

  Recipient tells sender the size of its input buffer and sends acknowledgements when data has been received. Sequence numbers are used to detect missing segments.

- o The SACK ption

  Selective acknowledgement so only the dropped segments need to be retransmitted.

- o The RST Flag

  Used to terminate a connection when an abnormal situation happens

Transport Layer

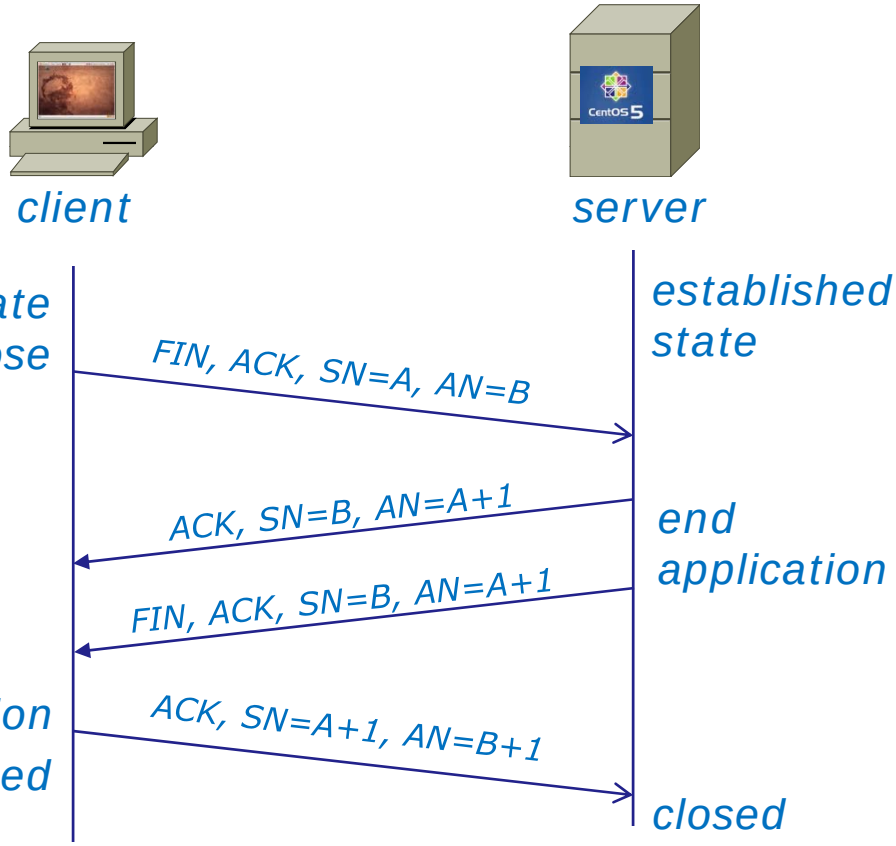
The Transmission Control Protocol

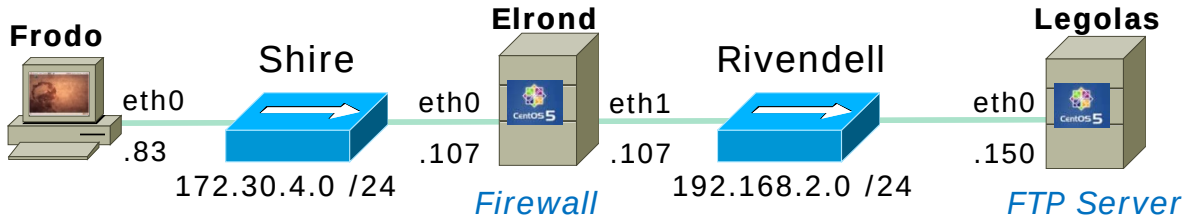
Closing a Connection

Four-Way Handshake

1. FIN, ACK
2. ACK
3. FIN, ACK
4. ACK

AN=Acknowledgment Number
SN=Sequence Number
ACK=ACK flag set
FIN=FIN flag set





Active Mode *is when server initiates new connection for data transfer*

Socket for data transfer

Client	Server
172.30.4.83	192.168.2.150
42571	20

```
ftp> get legolas
local: legolas remote: legolas
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for legolas (18 bytes).
226 File send OK.
18 bytes received in 0.04 secs (0.5 kB/s)
```

SIP	SP	DIP	DP	Protocol	Info
172.30.4.83	42855	192.168.2.150	21	FTP	Request: PORT 172,30,4,83,166,75
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 200 PORT command successful. Consider using PAS
172.30.4.83	42855	192.168.2.150	21	FTP	Request: RETR legolas
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [SYN] Seq=0 Wi
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [SYN, ACK] Seq
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [ACK] Seq=1 Ack=1
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 150 Opening BINARY mode data connection for leg
192.168.2.150	20	172.30.4.83	42571	FTP-DATA	FTP Data: 18 bytes
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [FIN, ACK] Seq=19 Ack=1 Win=5888 Len=0
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [ACK] Se
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [FIN, AC
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [ACK] Seq=20 Ack=2 Win=5888 Len=0
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 226 File send OK.
172.30.4.83	42855	192.168.2.150	21	TCP	42855 > ftp [ACK] Seq=82 Ack=263 Win=5856 Len=0

Retrieve legolas file

3 way handshake initiated by server

File transfer

4 way handshake to close connection

Tunable Kernel Parameters

Transport Layer

TCP Tunable Kernel Parameters

tcp_fin_timeout	*how long to keep in FIN-WAIT-2 state*
tcp_keepalive_time	*how long to keep an unused connection alive*
tcp_sack	*enable/disable selective acknowledgments*
tcp_timestamps	*enable RFC 1323 definition for round-trip measurement*
tcp_window_scaling	*enable RFC 1323 window scaling*
tcp_retries1	*how many times to retry before reporting an error*
tcp_retries2	*how many times to retry before killing connection*
tcp_syn_retries	*how many times to retransmit the SYN, ACK reply*

In the same directory:

ip_forward	*enable/disable selective acknowledgments*

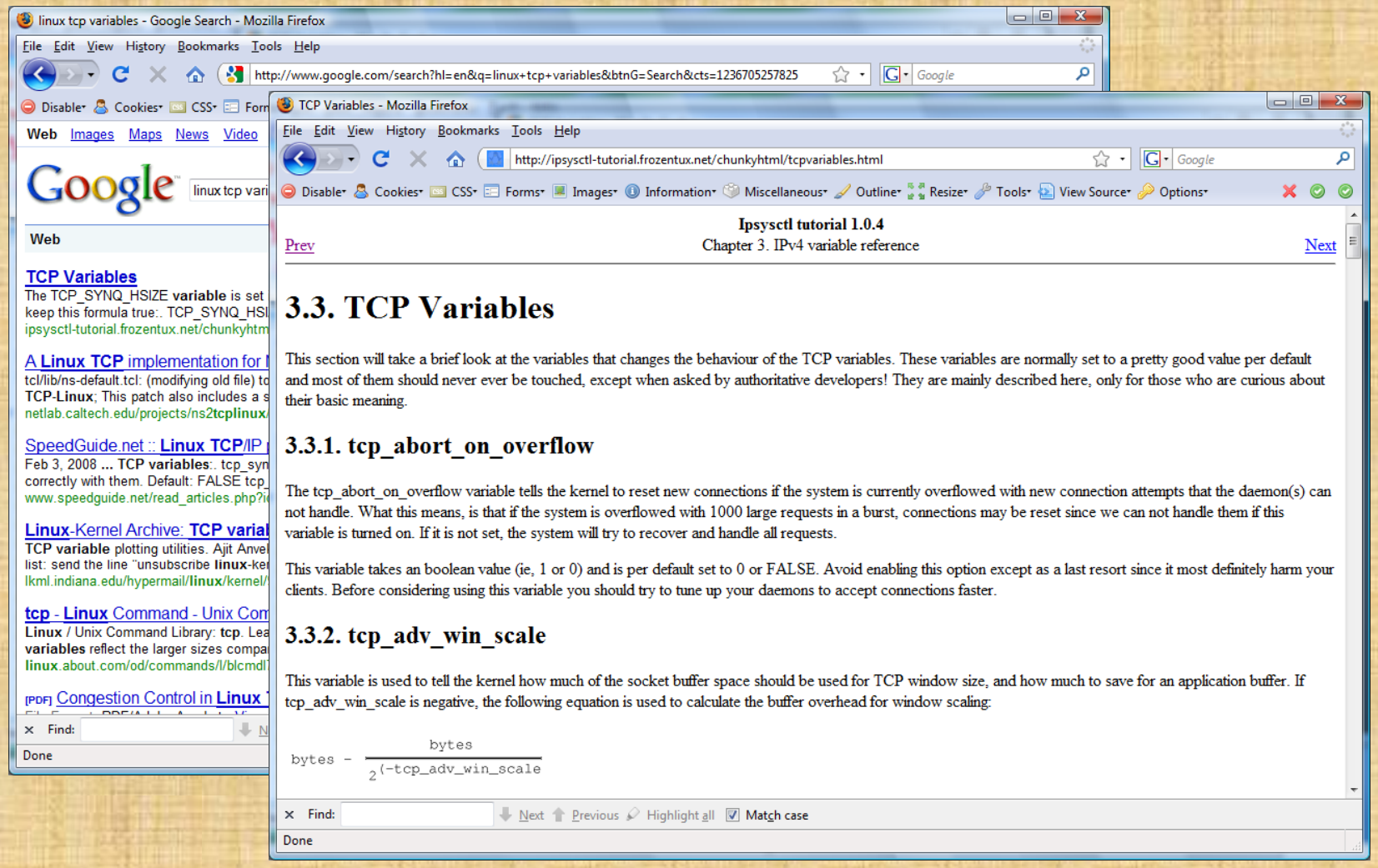
Exercise

Explore the TCP, UDP and other variables in the **/proc/sys/net/ipv4** directory

```
[root@bigserver ~]# ls /proc/sys/net/ipv4
cipso_cache_bucket_size           ip_dynaddr              tcp_dsack             tcp_retries2
cipso_cache_enable                ip_forward              tcp_ecn               tcp_rfc1337
cipso_rbm_optfmt                  ipfrag_high_thresh      tcp_fack              tcp_rmem
cipso_rbm_strictvalid             ipfrag_low_thresh       tcp_fin_timeout       tcp_sack
conf                              ipfrag_max_dist         tcp_frto              tcp_slow_start_after_idle
icmp_echo_ignore_all              ipfrag_secret_interval  tcp_keepalive_intvl   tcp_stdurg
icmp_echo_ignore_broadcasts       ipfrag_time             tcp_keepalive_probes  tcp_synack_retries
icmp_errors_use_inbound_ifaddr    ip_local_port_range     tcp_keepalive_time    tcp_syncookies
icmp_ignore_bogus_error_responses ip_nonlocal_bind        tcp_low_latency       tcp_syn_retries
icmp_ratelimit                    ip_no_pmtu_disc         tcp_max_orphans       tcp_timestamps
icmp_ratemask                     neigh                   tcp_max_syn_backlog   tcp_tso_win_divisor
igmp max memberships              netfilter               tcp max tw buckets    tcp tw recycle
igmp_max_msf                      route                   tcp_mem               tcp_tw_reuse
inet_peer_gc_maxtime              tcp_abc                 tcp_moderate_rcvbuf   tcp_window_scaling
inet_peer_gc_mintime              tcp_abort_on_overflow   tcp_mtu_probing       tcp_wmem
inet_peer_maxttl                  tcp_adv_win_scale       tcp_no_metrics_save
tcp_workaround_signed_windows
inet_peer_minttl                  tcp_app_win             tcp_orphan_retries    udp_mem
inet_peer_threshold               tcp_base_mss            tcp_reordering        udp_rmem_min
ip_conntrack_max                  tcp_congestion_control  tcp_retrans_collapse  udp_wmem_min
ip_default_ttl                    tcp_dma_copybreak       tcp_retries1

[root@bigserver ~]# cat /proc/sys/net/ipv4/tcp_sack
1
[root@bigserver ~]# cat /proc/sys/net/ipv4/tcp_syn_retries
5
[root@bigserver ~]#
```

Exercise

Google linux tcp variables



Security Issues

Transport Layer

Security Issues

Resource: *www.securityfocus.org*

- SYN Flooding

  " ... Bombarding a system with, say, dozens of falsified connection requests a minute can seriously degrade its ability to give service to legitimate connection requests. This is why the attack is said to "deny service" to the system's users. ..."

  Source: http://www.securityfocus.com/advisories/141

- Falsifying TCP Communications

  "... In IP spoofing, an attacker gains unauthorized access to a computer or a network by making it appear that a malicious message has come from a trusted machine by "spoofing" the IP address of that machine. ..."

  Source: http://www.securityfocus.com/infocus/1674

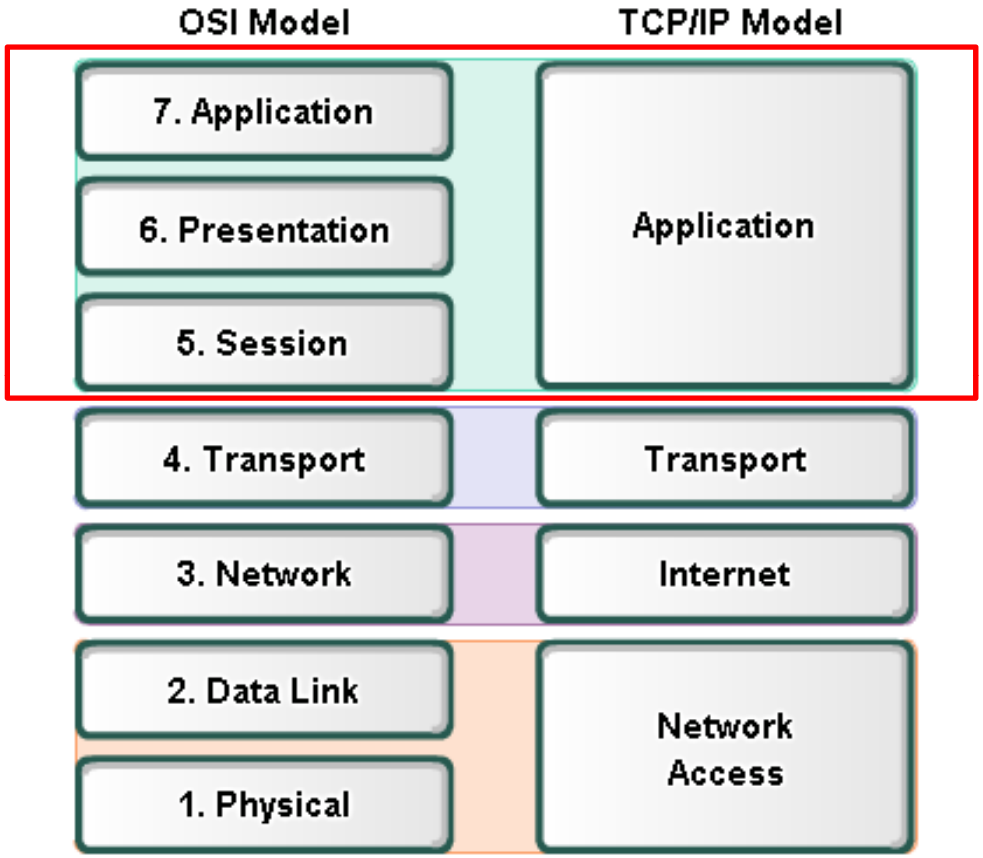
- Hijacking connections

  "... Another consequence, specific to TCP, is sequence number prediction, which can lead to session hijacking or host impersonating. This method builds on IP spoofing, since a session, albeit a false one, is built. ...

  source: http://www.securityfocus.com/infocus/1674

Application Layer

Protocol and Reference Models



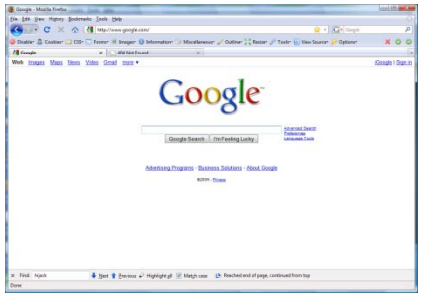
- The **Open Systems Interconnection (OSI)** model is the *most widely known internetwork reference model*.

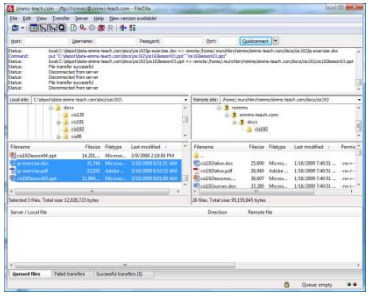
Application Layer

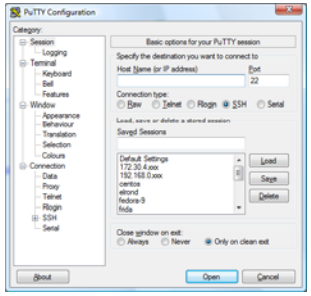
Applications

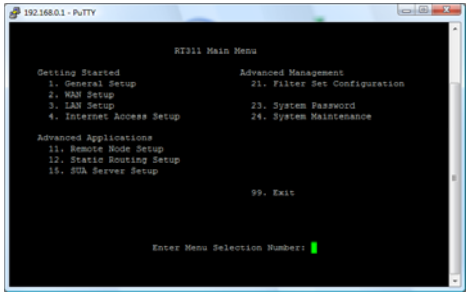
Examples:

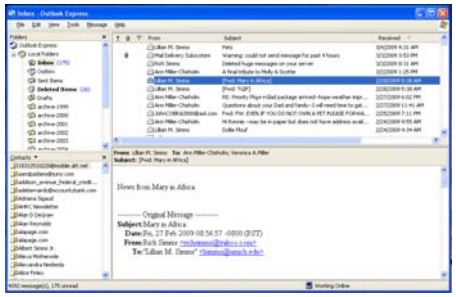
- Web servers
- FTP servers
- SSH daemon
- Telnet server
- email











Application Layer

Responsibilities of Applications

Network connections, routing, and transfer of data are all taken care of by the lower layers of the protocol stack. What must applications do?

- Authenticate users
- Control access
- Log important information
- Format data (compress/encrypt)
- Provide whatever functionality is desired.

Service Ports

Last week we talked about Layer 4 ports . Ports are used to direct requests to the appropriate service/application

```
< snipped >
# 21 is registered to ftp, but also used by fsp
ftp             21/tcp
ftp             21/udp          fsp fspd
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp                          # SSH Remote Login Protocol
telnet          23/tcp
telnet          23/udp
# 24 - private mail system
lmtp            24/tcp                          # LMTP Mail Delivery
lmtp            24/udp                          # LMTP Mail Delivery
smtp            25/tcp          mail
smtp            25/udp          mail
< snipped >
domain          53/tcp                          # name-domain server
domain          53/udp
whois++         63/tcp
whois++         63/udp
bootps          67/tcp                          # BOOTP server
bootps          67/udp
bootpc          68/tcp          dhcpc           # BOOTP client
bootpc          68/udp          dhcpc
tftp            69/tcp
tftp            69/udp
finger          79/tcp
finger          79/udp
http            80/tcp          www www-http    # WorldWideWeb HTTP
http            80/udp          www www-http    # HyperText Transfer Protocol
kerberos        88/tcp          kerberos5 krb5  # Kerberos v5
< snipped >
```

Application Layer

The Client-Server Model

Clients

Programs that are generally run on demand, and initiate the network connection to the server.
Examples: telnet, ftp, ssh, browsers, email clients.

Servers

Programs (services/daemons) that are constantly running in the background waiting for client connections.

- Services and Ports: */etc/services*
- Architecture:
  - Direct or iterative servers – listens to a particular port and directly responds to requests
  - Indirect or concurrent servers (e.g. super daemons) – listens to a particular port and then starts up another server program to process the request

Application Layer

The Super Daemons

- There are three primary super-daemons controlling server services.
- Super daemons spawn other daemons to handle specific client requests.

1. inetd - From early UNIX days, this was the primary daemon for handling tcp application services. It is being replaced by xinetd.

2. portmap - portmapper operates with Remote Procedure Call (RCP) applications.

3. xinetd - Extended Internet Services Daemon: used by modern distributions of Linux.

telnet
(via port 23)

Installing and Configuring Telnet (Red Hat Family)

Is it installed?

```
[root@bigserver ~]# rpm -qa | grep telnet
telnet-0.17-39.el5
telnet-server-0.17-39.el5
[root@bigserver ~]#
```

client

server

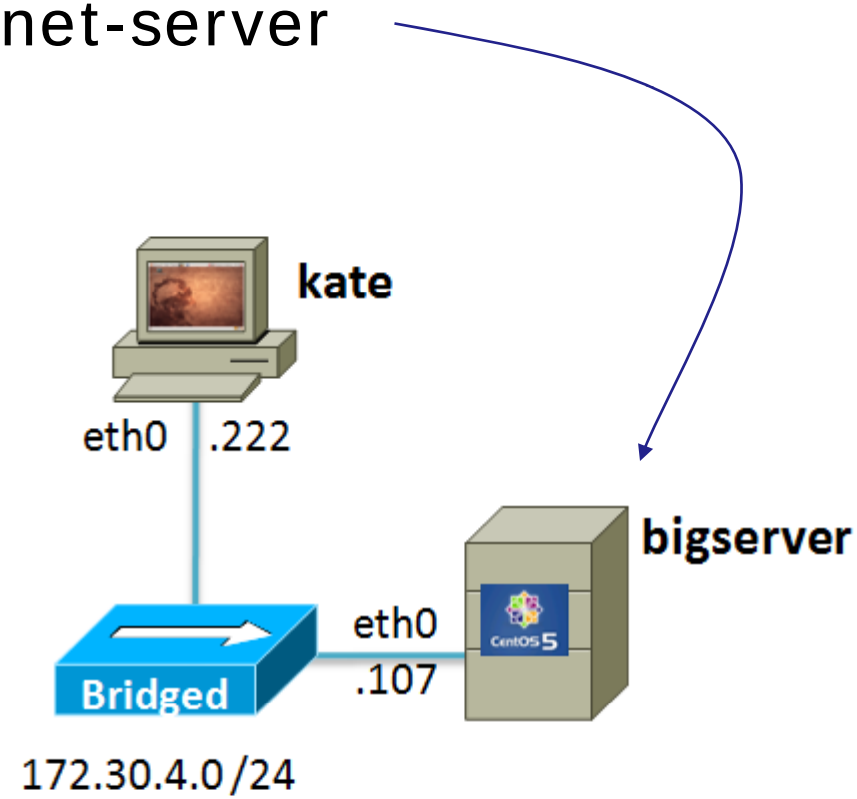
No response means it is not installed

*Use **dpkg –l | grep telnet** on the Debian family*

Installing and Configuring Telnet

Installing Telnet

yum install telnet-server



Installing and Configuring Telnet

```
[root@bigserver ~]# yum install telnet-server
Loading "fastestmirror" plugin
Loading mirror speeds from cached hostfile
 * base: centos.mirrors.redwire.net
 * updates: centos.mirrors.redwire.net
 * addons: centos.mirrors.redwire.net
 * extras: mirrors.usc.edu
Setting up Install Process
Parsing package install arguments
Resolving Dependencies
--> Running transaction check
---> Package telnet-server.i386 1:0.17-39.el5 set to be updated
--> Processing Dependency: xinetd for package: telnet-server
--> Running transaction check
---> Package xinetd.i386 2:2.3.14-10.el5 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved
```

Note that the telnet server uses xinetd

Installing and Configuring Telnet

```
Dependencies Resolved

================================================================================
 Package                 Arch       Version           Repository        Size
================================================================================
Installing:
 telnet-server           i386       1:0.17-39.el5     base               35 k
Installing for dependencies:
 xinetd                  i386       2:2.3.14-10.el5   base              124 k

Transaction Summary
================================================================================
Install      2 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 159 k
Is this ok [y/N]: y
Downloading Packages:
(1/2): xinetd-2.3.14-10.e 100% |=========================| 124 kB    00:00
(2/2): telnet-server-0.17 100% |=========================|  35 kB    00:00
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: xinetd                      ########################### [1/2]
  Installing: telnet-server               ########################### [2/2]

Installed: telnet-server.i386 1:0.17-39.el5
Dependency Installed: xinetd.i386 2:2.3.14-10.el5
Complete!
[root@bigserver ~]#
```

Installing and Configuring Telnet

Edit the configuration file

```
[root@arwen ~]# cat /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
#       unencrypted username/password pairs for authentication.
service telnet
{
        flags           = REUSE     Sets the TCP/IP socket to be reusable (after restarts)
        socket_type     = stream
        wait            = no        Starts a daemon for each request
        user            = root      Sets UID for daemon
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID   Use HOST for IP address and RECORD for terminal type
        disable         = no        This enables Telnet service
}
```

Great reference is "LINUX TCP/IP Network Administration" by Scott Mann

Installing and Configuring Telnet

Note: there may be multiple telnetd daemons

```
[root@arwen ~]# ls /etc/xinetd.d/*tel*
/etc/xinetd.d/ekrb5-telnet  /etc/xinetd.d/telnet
/etc/xinetd.d/krb5-telnet   /etc/xinetd.d/telnet.rpmsave

[root@arwen ~]# cat /etc/xinetd.d/krb5-telnet
# default: off
# description: The kerberized telnet server accepts normal telnet sessions, \
#              but can also use Kerberos 5 authentication.
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/kerberos/sbin/telnetd
        log_on_failure  += USERID
        disable         = yes
}
[root@arwen ~]#
```

Disable this telnet daemon

Installing and Configuring Telnet

Note: there may be multiple telnetd daemons

```
[root@arwen ~]# ls /etc/xinetd.d/*tel*
/etc/xinetd.d/ekrb5-telnet  /etc/xinetd.d/telnet
/etc/xinetd.d/krb5-telnet   /etc/xinetd.d/telnet.rpmsave

[root@arwen ~]# cat /etc/xinetd.d/ekrb5-telnet
# default: off
# description: The kerberized telnet server accepts only telnet sessions, \
#              which use Kerberos 5 authentication and encryption.
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/kerberos/sbin/telnetd
        server_args     = -e
        log_on_failure  += USERID
        disable         = yes
}
[root@arwen ~]#
```

Disable this telnet daemon

Installing and Configuring Telnet (Red Hat Family)

Start or restart service

```
[root@arwen ~]# service xinetd restart
Stopping xinetd:                                           [  OK  ]
Starting xinetd:                                           [  OK  ]
```

Or have xinetd reread it's configuration file

```
[root@arwen ~]# killall -1 xinetd
```

Automatically start at system boot

```
[root@arwen ~]# chkconfig xinetd on
[root@arwen ~]# chkconfig --list xinetd
xinetd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@arwen ~]#
```

Installing and Configuring Telnet

```
[root@arwen ~]# chkconfig –list

< snipped >

xinetd based services:
        chargen-dgram:  off
        chargen-stream: off
        daytime-dgram:  off
        daytime-stream: off
        discard-dgram:  off
        discard-stream: off
        echo-dgram:     off
        echo-stream:    off
        eklogin:        off
        ekrb5-telnet:   off
        gssftp:         off
        klogin:         off
        krb5-telnet:    off
        kshell:         off
        rsync:          off
        tcpmux-server:  off
        telnet:         on
        time-dgram:     off
        time-stream:    off
```

xinetd is a super daemon which acts as an umbrella for many other services

Multiple telnet daemons

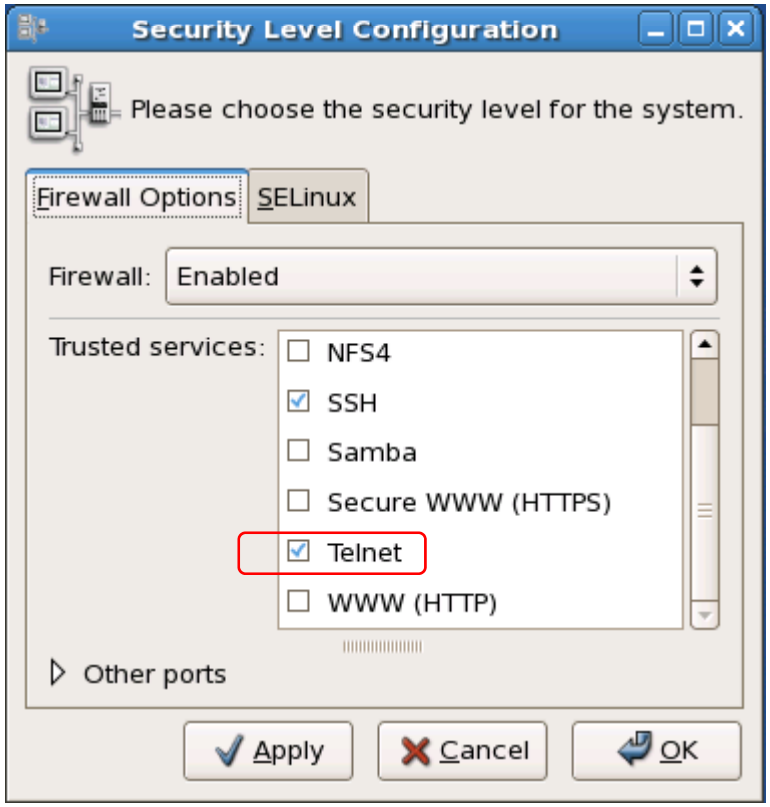
Firewall for Telnet

CentOS Default

```
[root@bigserver ~]# cat /etc/sysconfig/iptables
# Firewall configuration written by system-config-securitylevel
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d 224.0.0.251 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
[root@bigserver ~]#
```

Only the SSH port (22) is open for new incoming connections

Firewall for Telnet (CentOS)



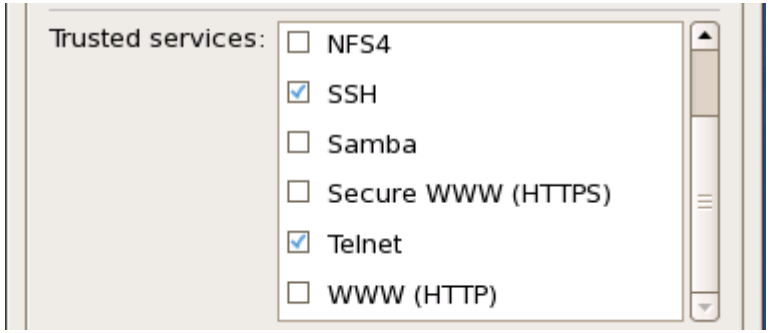
Open the Telnet port by placing check next to Telnet in the Trusted service list

*Later we will learn how to make these changes using the **iptables** command*

Firewall for Telnet

CentOS Modified

```
root@bigserver ~]# cat /etc/sysconfig/iptables
# Firewall configuration written by system-config-securitylevel
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d 224.0.0.251 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 23 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
[root@bigserver ~]#
```



Telnet and SSH port are open for incoming connections

Installing and Configuring Telnet

netstat

```
[root@bigserver ~]# netstat -tln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 127.0.0.1:2208              0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:111                 0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:6000                0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:21                  0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:23                  0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:631               0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:792                 0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:25                0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:2207              0.0.0.0:*                   LISTEN
tcp        0      0 :::6000                     :::*                        LISTEN
tcp        0      0 :::22                       :::*                        LISTEN
[root@bigserver ~]#
```

*Use **netstat -tln** command to see what port numbers your system is listening for requests on*

Installing and Configuring Telnet

netstat

```
[root@bigserver ~]# netstat -tl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 r1.localdomain:2208        *:*                         LISTEN
tcp        0      0 *:sunrpc                   *:*                         LISTEN
tcp        0      0 *:x11                      *:*                         LISTEN
tcp        0      0 *:ftp                      *:*                         LISTEN
tcp        0      0 *:telnet                   *:*                         LISTEN
tcp        0      0 r1.localdomain:ipp         *:*                         LISTEN
tcp        0      0 *:792                      *:*                         LISTEN
tcp        0      0 r1.localdomain:smtp        *:*                         LISTEN
tcp        0      0 r1.localdomain:2207        *:*                         LISTEN
tcp        0      0 *:x11                      *:*                         LISTEN
tcp        0      0 *:ssh                      *:*                         LISTEN
[root@bigserver ~]#
```

*Use **netstat –tl** command to see what port names your system is listening for requests on*

Installing and Configuring Telnet

telnetd processes

```
[cis192@bigserver ~]$ ps -ef | grep telnet
root      6156  6118  0 07:52 ?        00:00:00 in.telnetd: kate
root      6268  6118  0 07:53 ?        00:00:00 in.telnetd: 192.168.0.27
root      6299  6118  0 07:56 ?        00:00:00 in.telnetd: 192.168.0.23
cis192    6325  6270  0 07:56 pts/2    00:00:00 grep telnet
[cis192@bigserver ~]$
```

Individual telnetd daemons are run for each session

Installing and Configuring Telnet

Edit the configuration file

```
[root@arwen ~]# cat /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
#       unencrypted username/password pairs for authentication.
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        only_from       = 192.168.0.23
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
        disable         = no
}
[root@arwen ~]#
```

Use only_from to restrict clients that can access the Telnet service

Installing and Configuring Telnet

Only_ from examples

```
only_from = arwen
```
hostname

```
only_from = arwen legolas
```
multiple hostnames

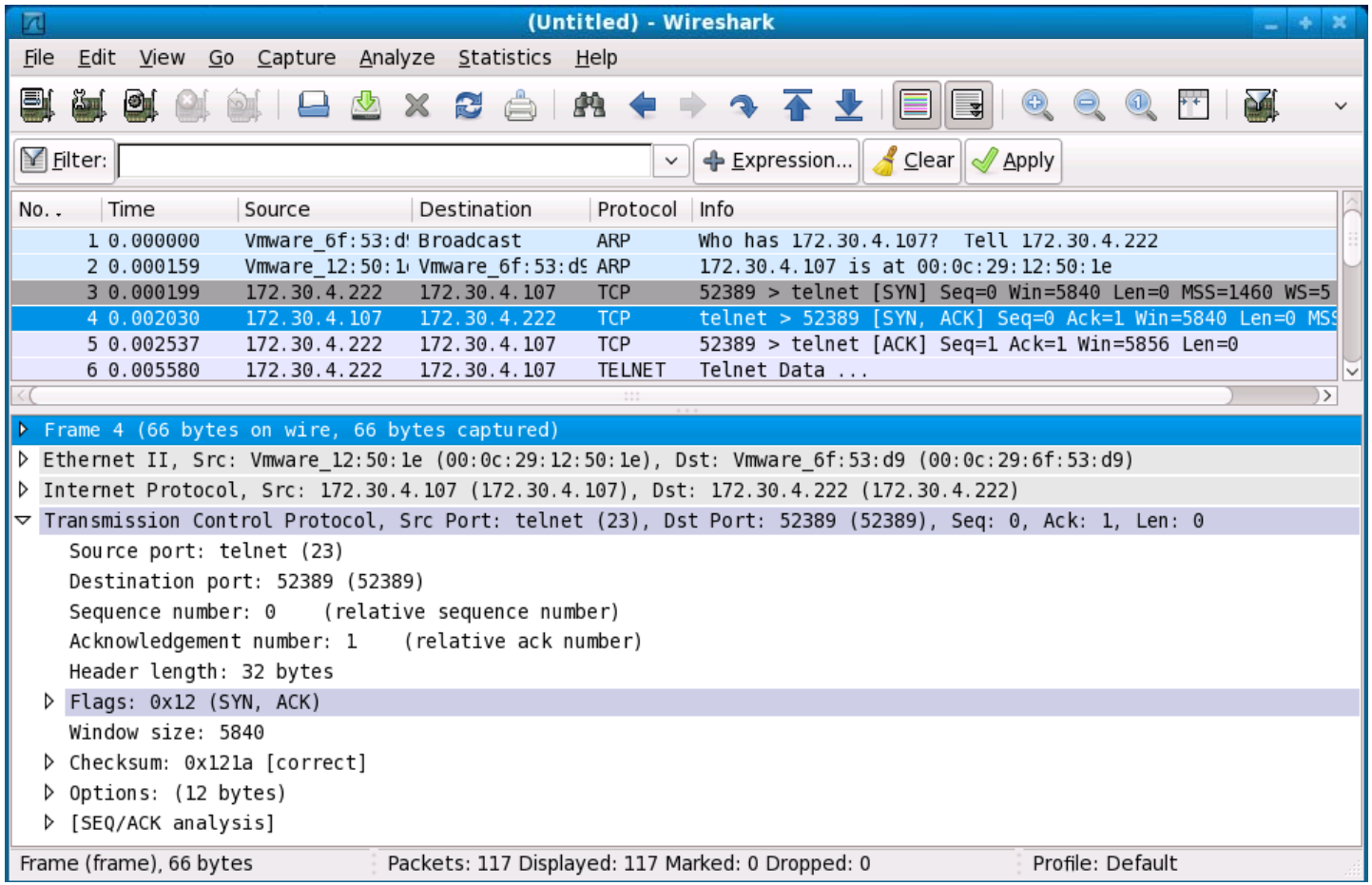
```
only_from = 192.168.3.12 192.168.3.14
```
or IP addresses

```
only_from = 192.168.3.{12, 14}
```
same as above

```
only_from = 192.168.0.0
```
0's are wildcards

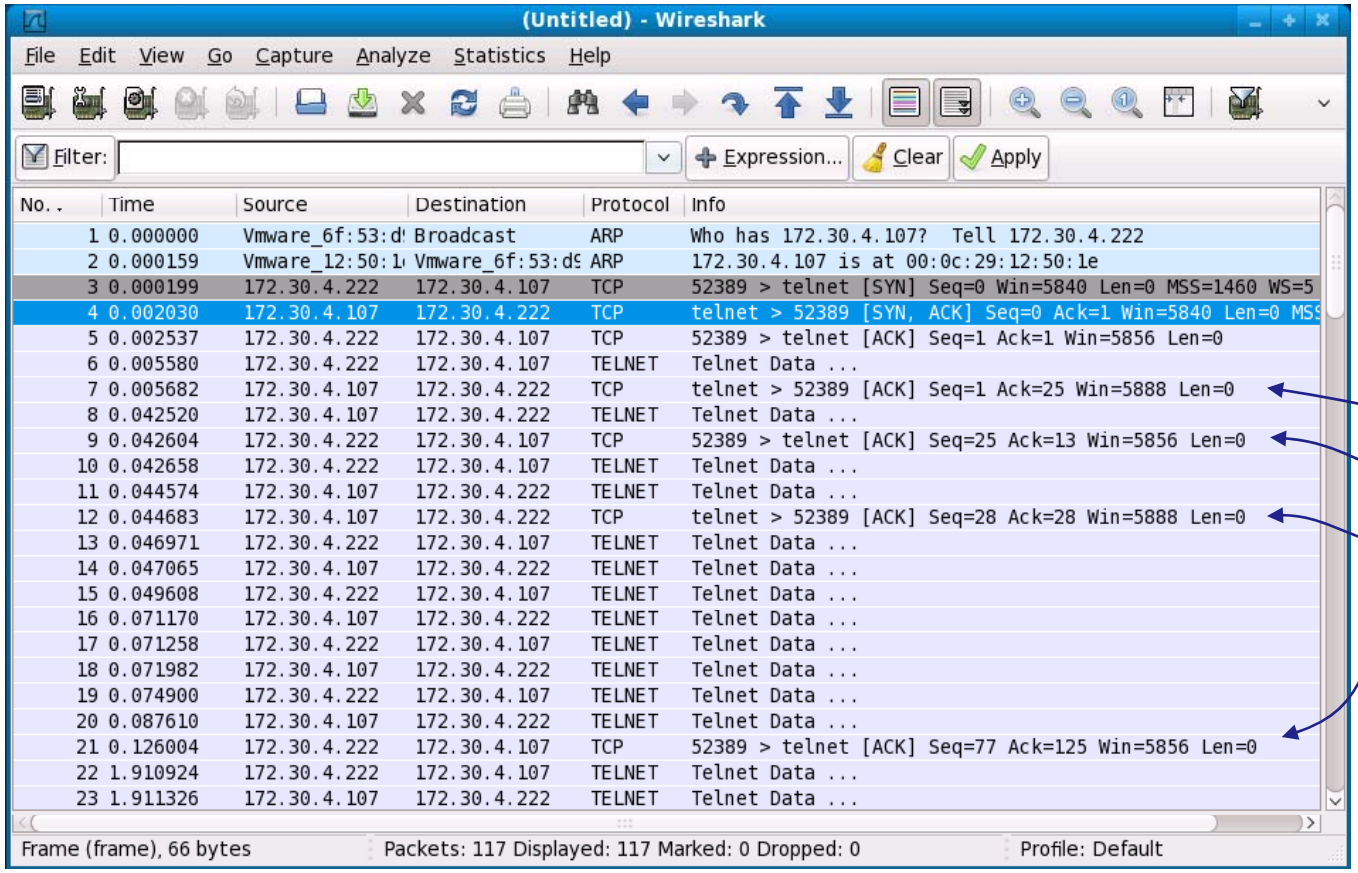
```
only_from = sauron 172.30.4.0 10.10.10.{1, 200}
```
mixes

Installing and Configuring Telnet



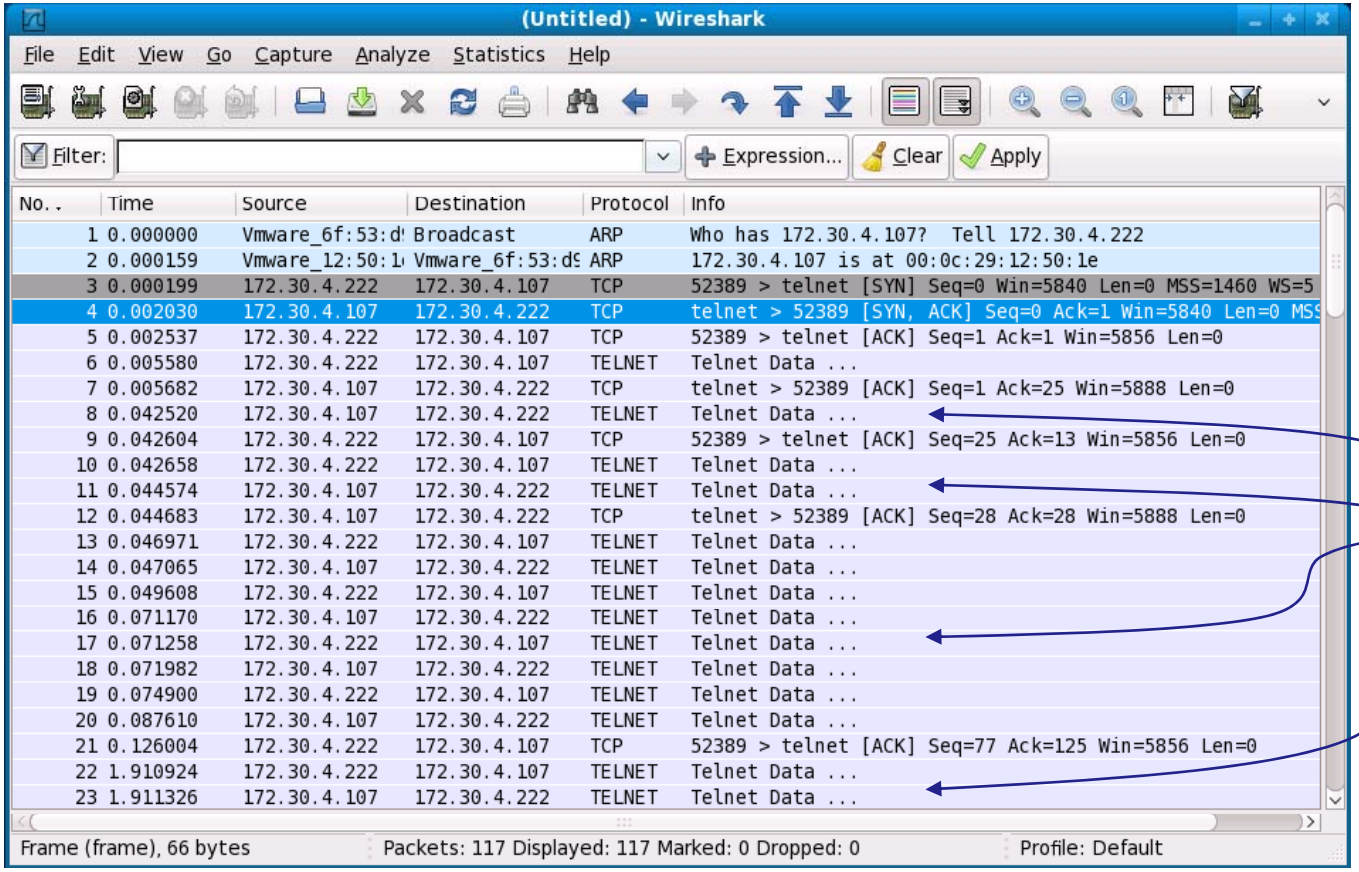
3-way handshake that initiates TCP connection

Installing and Configuring Telnet



TCP acknowledgments (ACKS) of data received

Installing and Configuring Telnet



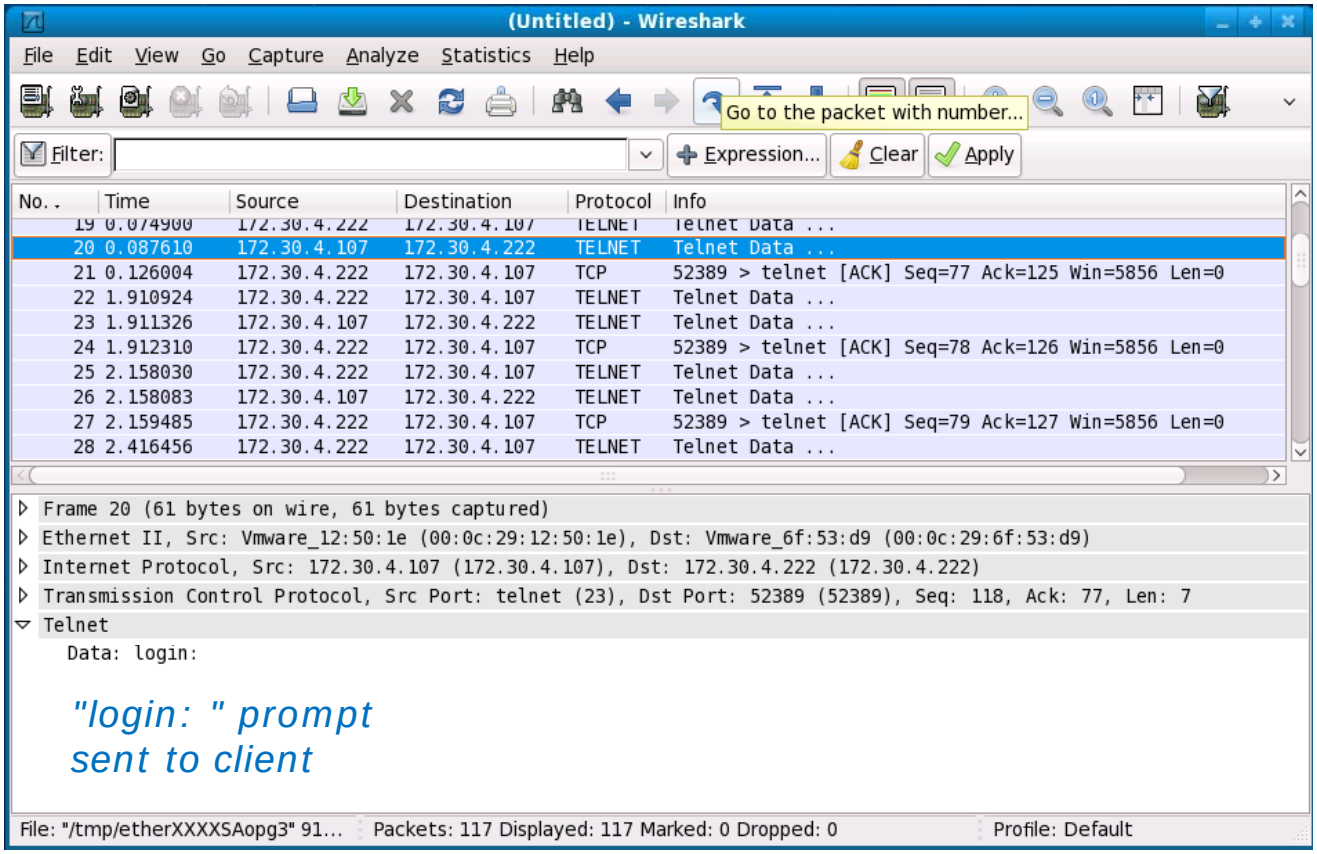
Installing and Configuring Telnet

Layer 2 Link frame with MAC addresses

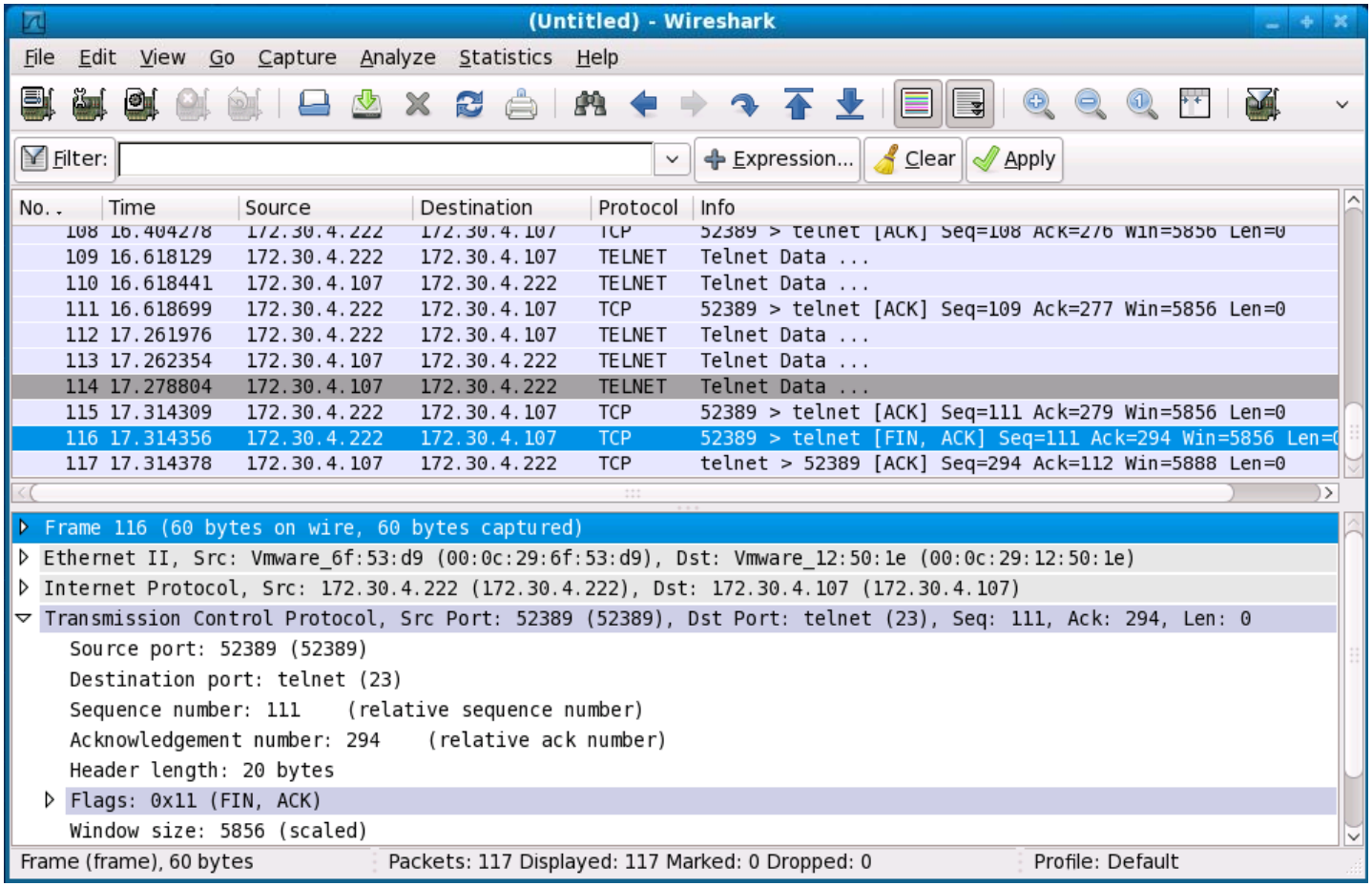
Layer 3 Network packet with IP addresses

Layer 4 Transport segment with port numbers

Layer 5 Application data



Installing and Configuring Telnet



Connection terminated

Installing and Configuring Telnet

Troubleshooting

```
root@sun:~# telnet 172.30.4.107
Trying 172.30.4.107...
telnet: Unable to connect to remote host: Connection refused
```

Open the firewall on the Telnet sever to accept incoming Telnet connections

```
root@sun:~# telnet 172.30.4.107
Trying 172.30.4.107...
Connected to 172.30.4.107.
Escape character is '^]'.
getaddrinfo: localhost Name or service not known
Connection closed by foreign host.
```

On the Telnet server, insure that the server's own hostname is configured in /etc/hosts

Installing and Configuring Telnet

Troubleshooting (continued)

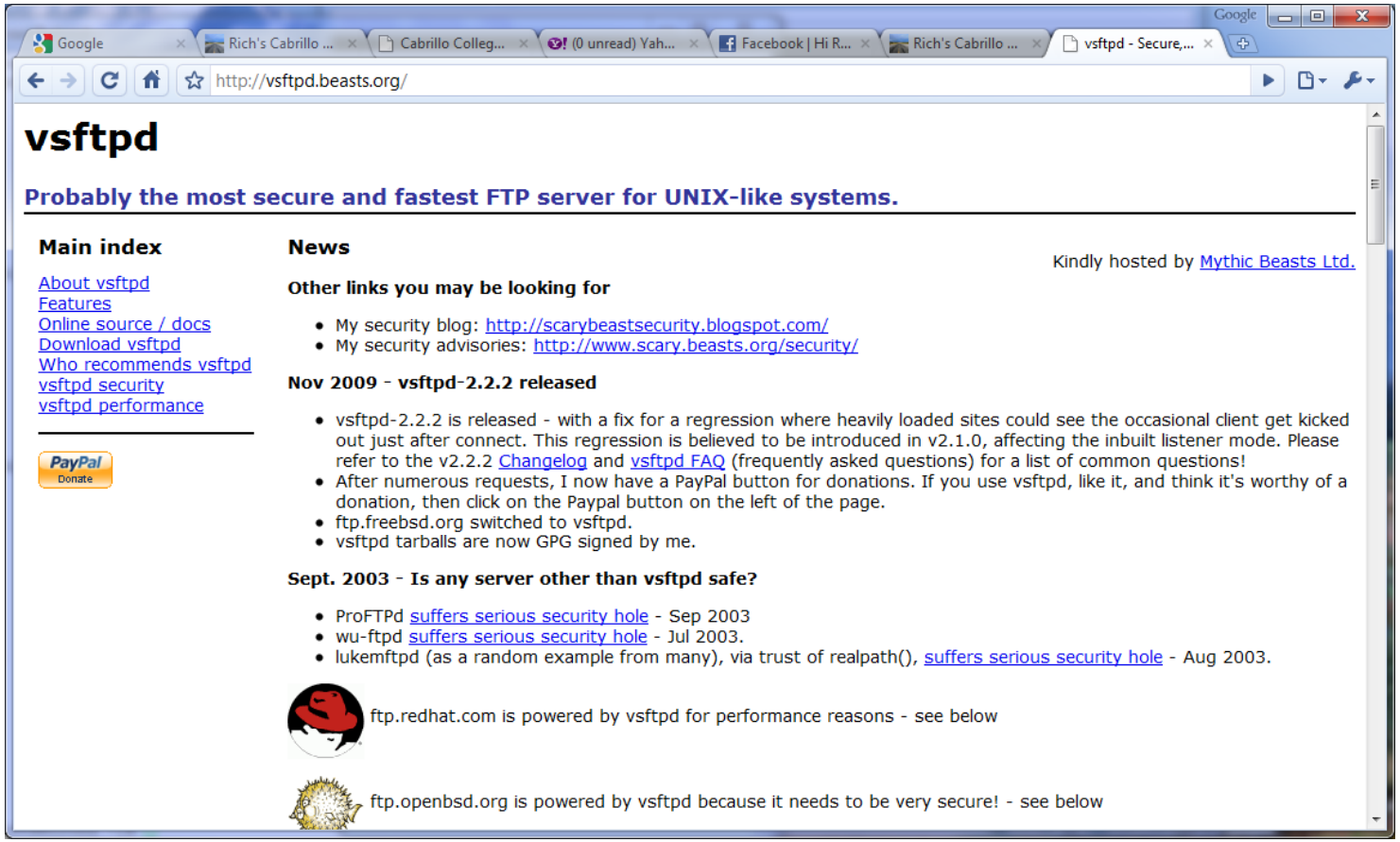
```
root@kate:~# telnet 172.30.4.107
Trying 172.30.4.107...
Connected to 172.30.4.107.
Escape character is '^]'.
Connection closed by foreign host.
root@kate:~# telnet 172.30.4.107
```

Edit /etc/xinetd.d/telnet file to allow connections from this client

vsftpd

vsftpd

- vsftpd = Very Secure FTP Daemon
- Licensed under the GNU General Public License
- http://vsftpd.beasts.org/



Installing and Configuring vsftpd (Red Hat Family)

Is it installed?

```
[root@bigserver ~]# rpm -qa | grep vsftpd
vsftpd-2.0.5-12.el5
```

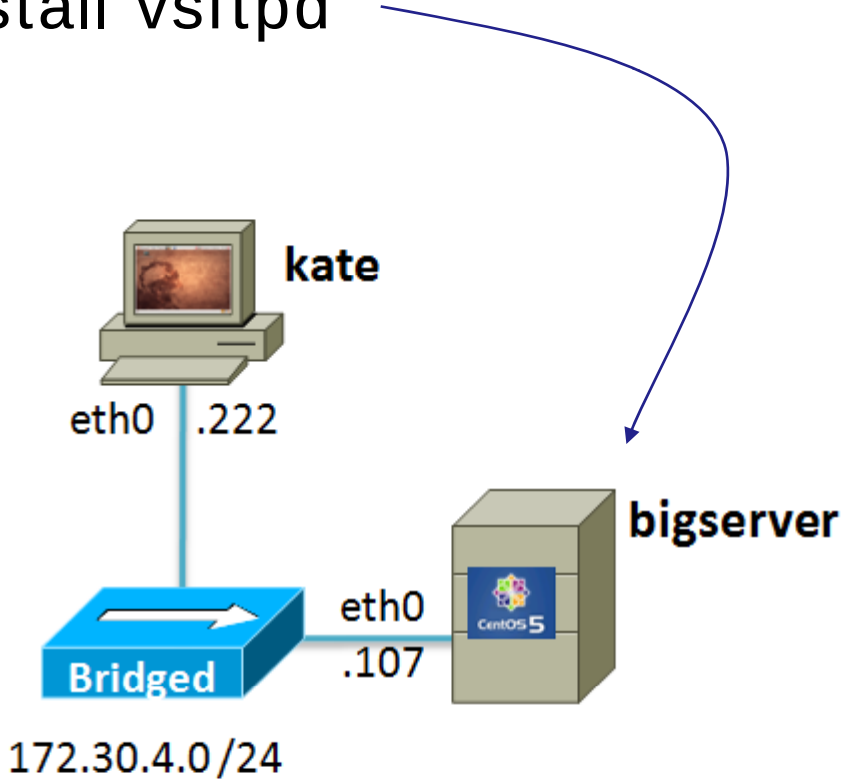
No response means it is not installed

*Use **dpkg –l | grep vsftpd** on the Debian family*

vsftpd

Installing vsftpd

yum install vsftpd



vsftpd

```
[root@bigserver ~]# yum install vsftpd
Loading "fastestmirror" plugin
Loading mirror speeds from cached hostfile
 * base: mirror.hmc.edu
 * updates: mirrors.easynews.com
 * addons: mirrors.cat.pdx.edu
 * extras: centos.cogentcloud.com
Setting up Install Process
Parsing package install arguments
Resolving Dependencies
--> Running transaction check
---> Package vsftpd.i386 0:2.0.5-12.el5 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved
```

vsftpd

```
Dependencies Resolved

================================================================================
 Package                 Arch       Version          Repository        Size
================================================================================
Installing:
 vsftpd                  i386       2.0.5-12.el5     base              137 k

Transaction Summary
================================================================================
Install      1 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 137 k
Is this ok [y/N]: y
Downloading Packages:
(1/1): vsftpd-2.0.5-12.el 100% |=========================| 137 kB    00:00
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: vsftpd                       ########################### [1/1]

Installed: vsftpd.i386 0:2.0.5-12.el5
Complete!
[root@bigserver ~]#
```

Installing and Configuring vsftpd

Does it use TCP Wrappers?

```
[root@bigserver ~]# type vsftpd
vsftpd is /usr/sbin/vsftpd
[root@bigserver ~]# ldd /usr/sbin/vsftpd
        linux-gate.so.1 =>  (0x0074c000)
        libssl.so.6 => /lib/libssl.so.6 (0x0012a000)
        libwrap.so.0 => /usr/lib/libwrap.so.0 (0x005cb000)
        libnsl.so.1 => /lib/libnsl.so.1 (0x00913000)
        libpam.so.0 => /lib/libpam.so.0 (0x00b11000)
        libcap.so.1 => /lib/libcap.so.1 (0x0084a000)
        libdl.so.2 => /lib/libdl.so.2 (0x00110000)
        libc.so.6 => /lib/libc.so.6 (0x0016f000)
        libcrypto.so.6 => /lib/libcrypto.so.6 (0x002b2000)
        libgssapi_krb5.so.2 => /usr/lib/libgssapi_krb5.so.2 (0x00bb4000)
        libkrb5.so.3 => /usr/lib/libkrb5.so.3 (0x003e5000)
        libcom_err.so.2 => /lib/libcom_err.so.2 (0x0092c000)
        libk5crypto.so.3 => /usr/lib/libk5crypto.so.3 (0x0054c000)
        libresolv.so.2 => /lib/libresolv.so.2 (0x00114000)
        libz.so.1 => /usr/lib/libz.so.1 (0x00478000)
        libaudit.so.0 => /lib/libaudit.so.0 (0x004c5000)
        /lib/ld-linux.so.2 (0x0085a000)
        libkrb5support.so.0 => /usr/lib/libkrb5support.so.0 (0x00fb5000)
        libkeyutils.so.1 => /lib/libkeyutils.so.1 (0x00961000)
        libselinux.so.1 => /lib/libselinux.so.1 (0x0048b000)
        libsepol.so.1 => /lib/libsepol.so.1 (0x004da000)
[root@bigserver ~]#
```

yes it does

Installing and Configuring vsftpd

Edit the configuration file

```
[root@arwen ~]# cat /etc/vsftpd/vsftpd.conf
[root@bigserver ~]# cat /etc/vsftpd/vsftpd.conf
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.

< snipped >

# You may fully customise the login banner string:
ftpd_banner=Welcome to the Simms FTP service.

< snipped >

tcp_wrappers=YES
[root@bigserver ~]#
```

Installing and Configuring vsftpd (Red Hat Family)

Start or restart service

```
[root@bigserver ~]# service vsftpd start
Starting vsftpd for vsftpd:                                [  OK  ]
[root@bigserver ~]#
```

Automatically start at system boot

```
[root@bigserver ~]# chkconfig vsftpd on
[root@bigserver ~]# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@bigserver ~]#
```

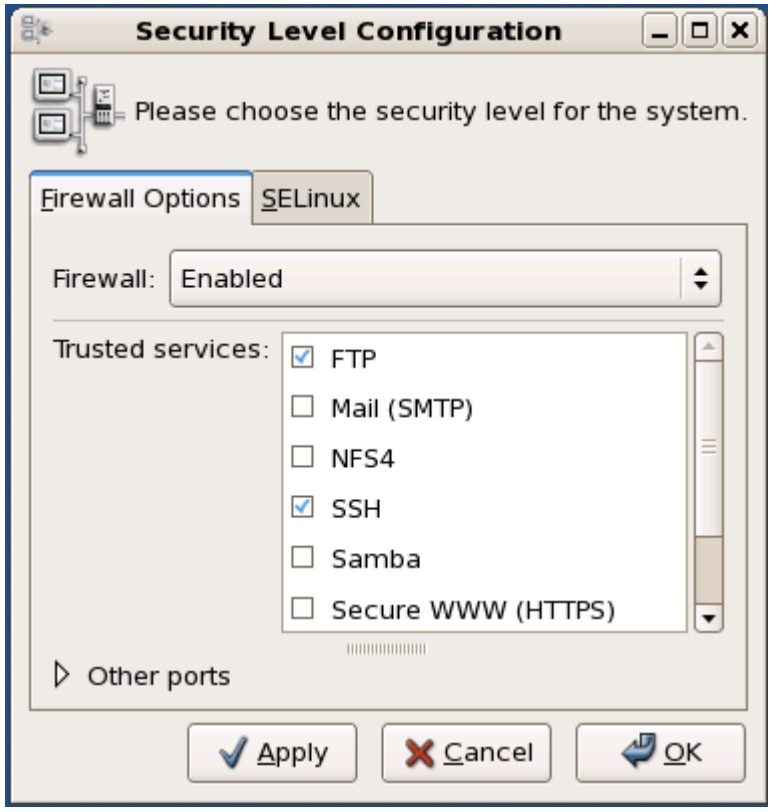
Firewall for FTP

CentOS Default

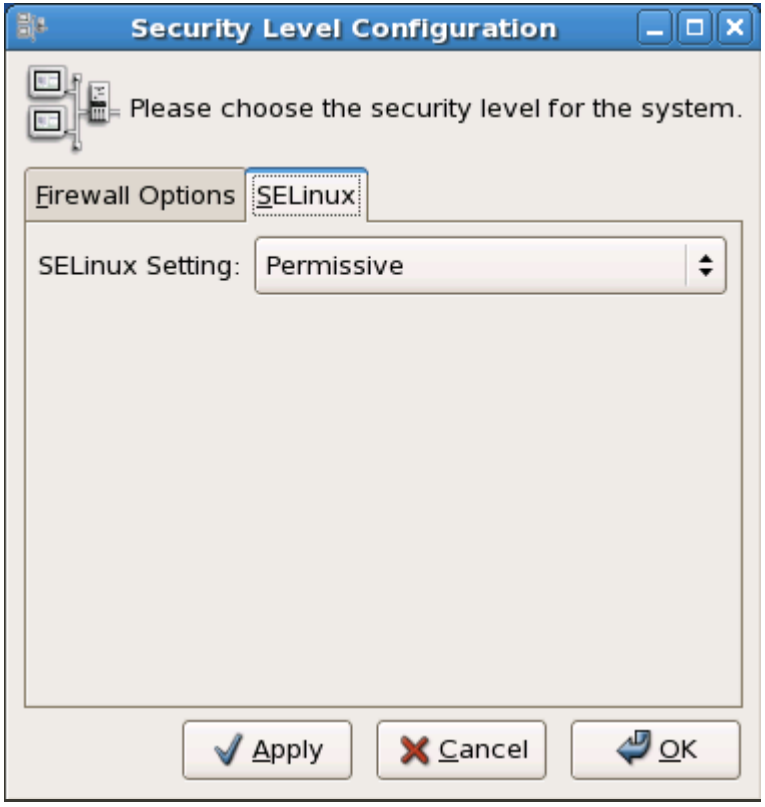
```
[root@bigserver ~]# cat /etc/sysconfig/iptables
# Firewall configuration written by system-config-securitylevel
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d 224.0.0.251 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
[root@bigserver ~]#
```

FTP port is not open, FTP clients will not be able to connect

Firewall for FTP (CentOS)



Open FTP port



SELinux set to Permissive

Firewall for FTP

CentOS Modified

```
root@bigserver ~]# cat /etc/sysconfig/iptables
# Firewall configuration written by system-config-securitylevel
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d 224.0.0.251 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 21 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 23 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
[root@bigserver ~]#
```

FTP port is now open

Installing and Configuring vsftpd

netstat

```
[root@bigserver ~]# netstat -tln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 127.0.0.1:2208              0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:111                 0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:6000                0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:21                  0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:23                  0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:631               0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:792                 0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:25                0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:2207              0.0.0.0:*                   LISTEN
tcp        0      0 :::6000                     :::*                        LISTEN
tcp        0      0 :::22                       :::*                        LISTEN
[root@bigserver ~]#
```

Use netstat command to see what ports your system is listening for requests on

Installing and Configuring vsftpd

netstat

```
[root@bigserver ~]# netstat -tl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 r1.localdomain:2208        *:*                         LISTEN
tcp        0      0 *:sunrpc                   *:*                         LISTEN
tcp        0      0 *:x11                      *:*                         LISTEN
tcp        0      0 *:ftp                      *:*                         LISTEN
tcp        0      0 *:telnet                   *:*                         LISTEN
tcp        0      0 r1.localdomain:ipp         *:*                         LISTEN
tcp        0      0 *:792                      *:*                         LISTEN
tcp        0      0 r1.localdomain:smtp        *:*                         LISTEN
tcp        0      0 r1.localdomain:2207        *:*                         LISTEN
tcp        0      0 *:x11                      *:*                         LISTEN
tcp        0      0 *:ssh                      *:*                         LISTEN
[root@bigserver ~]#
```

Use netstat command to see what ports your system is listening for requests on

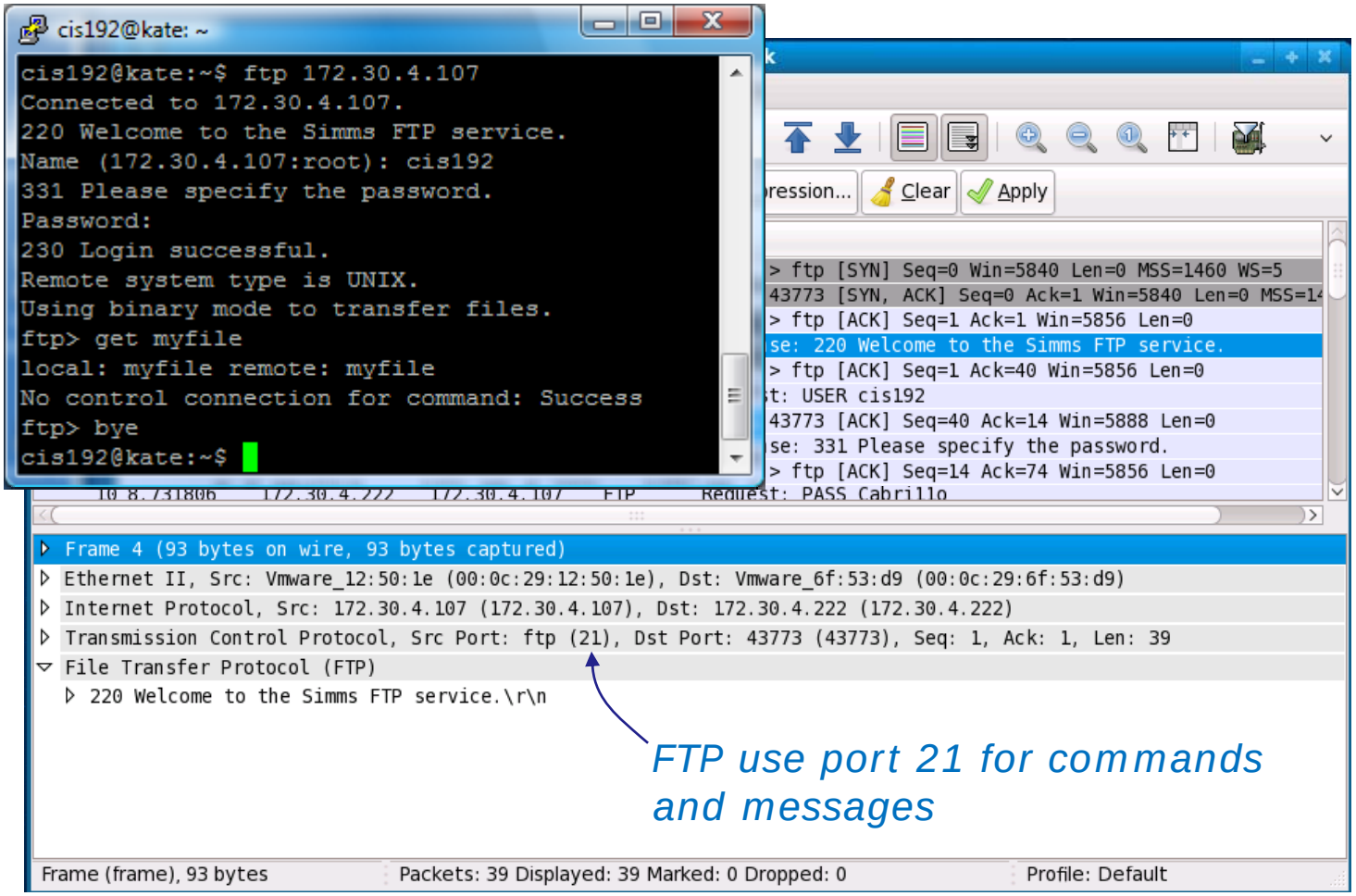
Installing and Configuring vsftpd

vsftpd processes

```
[root@bigserver ~]# ps -ef | grep vsftp
root     10541     1  0 00:29 ?        00:00:00 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
nobody   10806 10541  0 00:55 ?        00:00:00 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
cis192   10808 10806  0 00:55 ?        00:00:00 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
root     10812  9988  0 00:57 pts/0    00:00:00 grep vsftp
[root@bigserver ~]#
```

Individual vsftpd daemons are run for each session

Installing and Configuring vsftpd

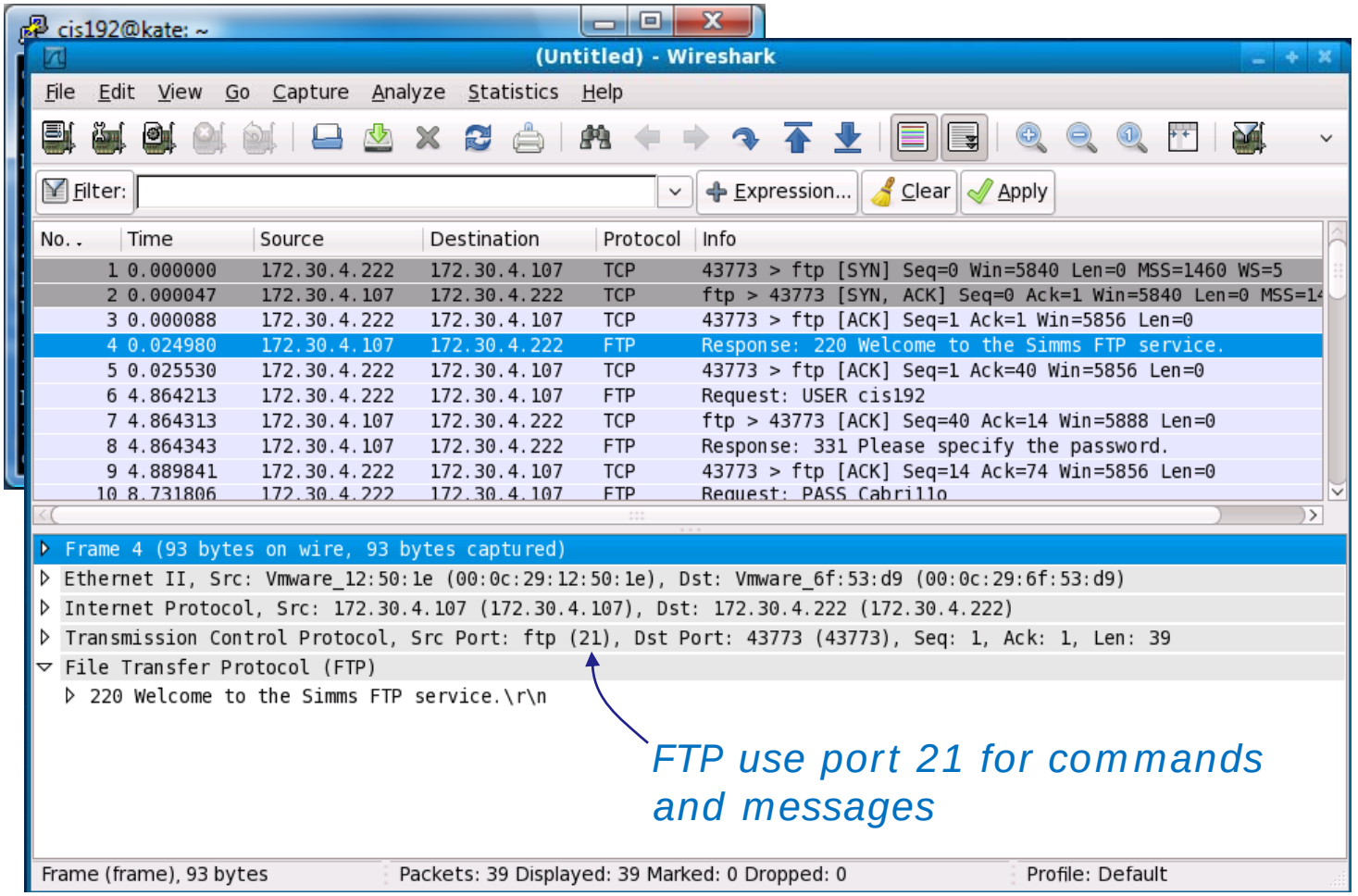


3-way handshake

Login is transmitted in clear text

Data (L5) encapsulated in TCP segment (L4), which is encapsulated in IP packet (L3), which is encapsulated in Ethernet frame (L2) which is encapsulated in L1 frame on the wire

Installing and Configuring vsftpd

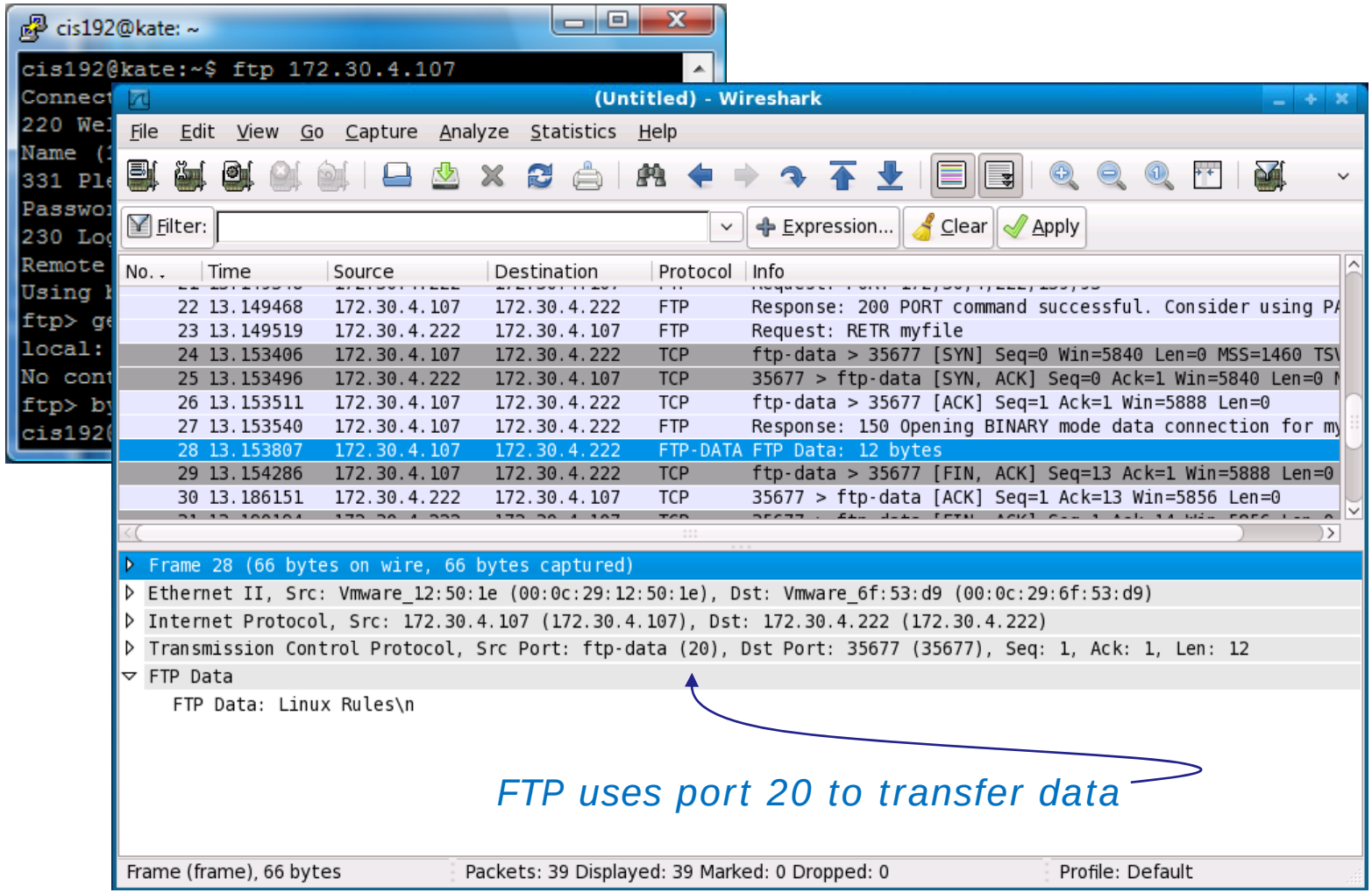


3-way handshake

Login is transmitted in clear text

Data (L5) encapsulated in TCP segment (L4), which is encapsulated in IP packet (L3), which is encapsulated in Ethernet frame (L2) which is encapsulated in L1 frame on the wire

Installing and Configuring vsftpd



Wrap

New commands, daemons and files:

- service
- chconfig
- killal
- netstat

Daemons and related configuraton files

inetd	/etc/inetd.conf
portmap	/etc/etc/rpc
xinetd	/etc/etc/xinetd.d
service	/etc//etc/init.d
chconfig	/etc/rc.d/rc*.d
tcpd	/etc/hosts.allow,hosts.deny
killall	

Next Class

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

Lab 4 due

Quiz questions for next class:

- How do you find out if vsftpd is installed?
- What two ports does FTP use?
- What command shows the ports on your system that are open and listening for requests?


Test 1

Backup

IP addresses for VM's in the classroom

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

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

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

SSH

super daemons

Application Layer

inet Daemon

- */etc/inetd.conf*
- */etc/services*
- */etc/protocols*

Application Layer

xinetd Daemon

Advantages

1. provides access control for TCP, UDP, and RPC services
2. Access limitations based on time
3. Extensive logging capabilities
4. Implements RFC 1413 username retrievals
5. Provides for hard reconfiguration
6. Provides numerous mechanisms to prevent denial of service attacks
7. Allows compiled in TCP_Wrappers through libwrap
8. Services may be bound to specific interfaces
9. Services may be forwarded (proxied) to another system
10. Supports ipv6

Application Layer

xinetd Daemon

Syntax:

service *service_name*
{
    attribute operator value value ...
}

Application Layer

xinetd Daemon

Required Attributes

1. socket_type
2. wait
3. user
4. server
5. port
6. protocol
7. rpc_version - only for RPC services
8. rpc_number - only for RPC services

Application Layer

xinetd Daemon

- Access Attributes
  1. only_from
  2. no_access
- The bind Attribute
- The redirect Attribute
- Incorporating TCP_Wrappers

Application Layer

xinetd Daemon

The xinetd Daemon command line options

1. -d
2. -syslog
3. -loop rate
4. -reuse
5. -limit
6. -logproc