Lesson Module Status

- Slides – draft
- Properties - done
- Flashcards - NA
- 1st minute quiz – done
- Web Calendar summary – done
- Web book pages – done
- Commands – done
- Howtos – NA
- Skills pacing - na
- Lab 4 published - done
- Extra credit lab published - done
- Practice test publish - done
- Depot (VMs) – done
- New quiz ?'s for next week - NA
- Add sniffer module for internal wireshark sniffing
- Add routerboard/MikoTik – done
- Add email option for all lesson quizzes
- Add opus answers directory/weekly cycle to housekeeping – done
- Bring MikroTik router - done

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.

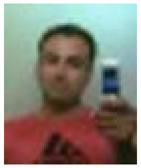
Chuck

Rich

Ryan

John

Robert


Josh

Patrick

Casady

Chris H.

Lieven

Jack

Julio

Drew

Edgar

Kay

Edwin

Aaron

Joe B.

Junious

Brynden

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

First Minute Quiz

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

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

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

Routing Continued and Transport Protocols

Objectives	Agenda
• Configure appropriate IP addresses, network and subnet masks, and broadcast addresses based on the size and number of network segments required. • Connect multiple network segments together using Linux servers as routers and configuring the appropriate routing tables. • Use a network sniffer to analyze network traffic between two hosts. • Identify, isolate, and correct malfunctions in a computer network. • Define the term 'socket' and describe its importance to the transport layer of the protocol stack.	• Quiz • Questions on previous material • Housekeeping • Virtual/Physical corner • Dynamic Routing • Quagga routing suite for Linux • Skills for doing Lab 4 • Transport Layer • TDP and UDP protocols • Service ports and sockets • Prepping for the test next week • Wrap

Questions on previous material

Questions?

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

Housekeeping

- Lab 3 due midnight
- Five posts due midnight

- Test 1 next week
- Lab 4 due in two weeks

- Extra credit lab on permanent NIC configuration available

- The real nosmo was rebooted this week, network any better?

- 3/6 Saturday workshop: 1 till whenever
- Lab assistants – Robert and Mark

- PE observation and survey tonight

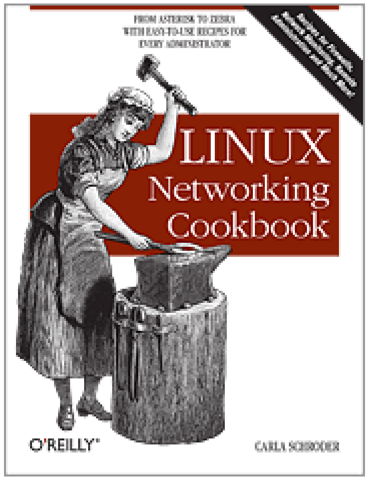
```
[rsimms@opus answers]$ head -30 /home/cis192/answers/lab2.simmsben
CIS 192 Lab 2
Benji Simms
Date: 02/25/2010

TBA hours: 5.5 hours
Station number: CIS-Lab-01
CPU: Intel Core2 Duo E7200 @ 2.53 GHz
RAM: 3.23 GB

FRODO TROUBLESHOOTING (Step 4)

Ping 172.30.4.1 error when eth0 is down:
Network is unreachable

Ping 172.30.4.1 error after releasing IP address:
Network is unreachable

Ping 207.62.186.9 error after deleting default gateway:
Network is unreachable

Ping opus.cabrillo.edu error with no DNS server:
unknown host opus.cabrillo.edu

Ping 172.30.4.1 error after disconnecting from network:
From 172.30.4.150 icmp_seq=10 Destination host Unreachable


CONFIGURATION AND CONNECTIVITY TESTS (Step 8)

*** Frodo ***
[rsimms@opus answers]$
```

After labs are graded I put up an example lab report showing the “answers” on Opus in:

/home/cis192/answers

Practicing skills at home

SBCs



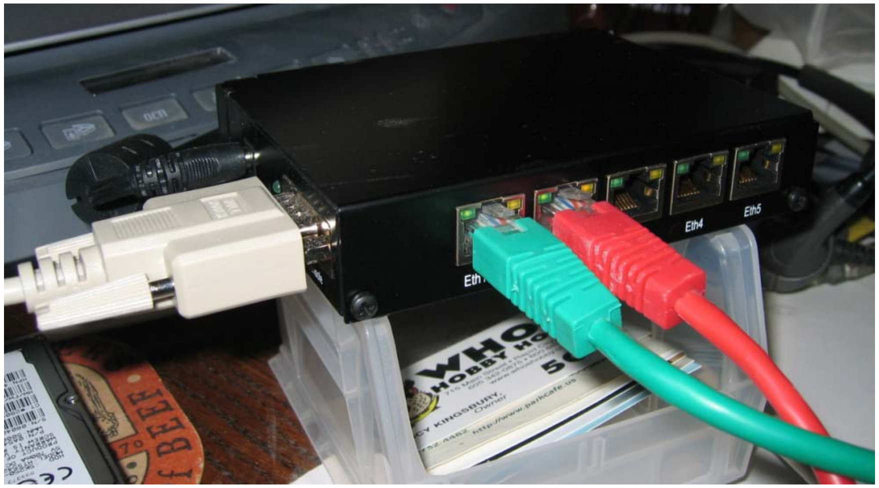
The **Linux Networking Cookbook** by Carla Schroeder has a section on SBCs (Single Board Computers):

- Small
- Quiet
- Low power consumption
- Can run Linux OS

Examples:

- Soekris Engineering (Santa Cruz) - http://soekris.com/
- PC Engines (Switzerland) - http://www.pcengines.ch/
- MikroTik Routerboard (Latvia) - http://www.routerboard.com/
- Many more at http://www.linuxfordevices.com/

MikroTik/Routerboard – A Linux based router

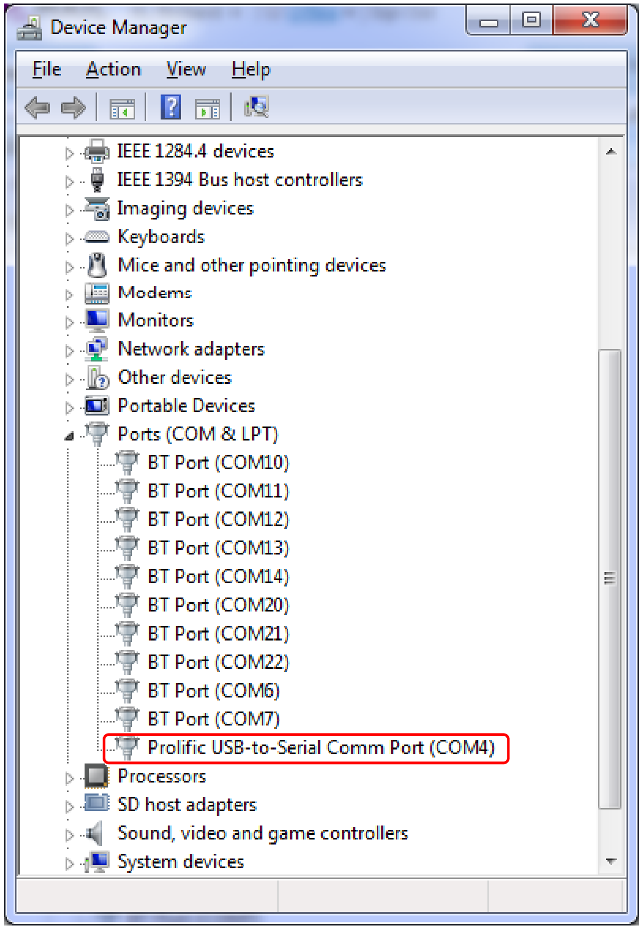


Assemble your own Linux based Router. This one has five Ethernet interfaces and uses 6.4 watts of power.

- *Eth1 is attached to the home LAN.*
- *Eth2 is attached to a 172.30.4.0/24 network.*
- *The serial cable (console) can be attached to a laptop.*

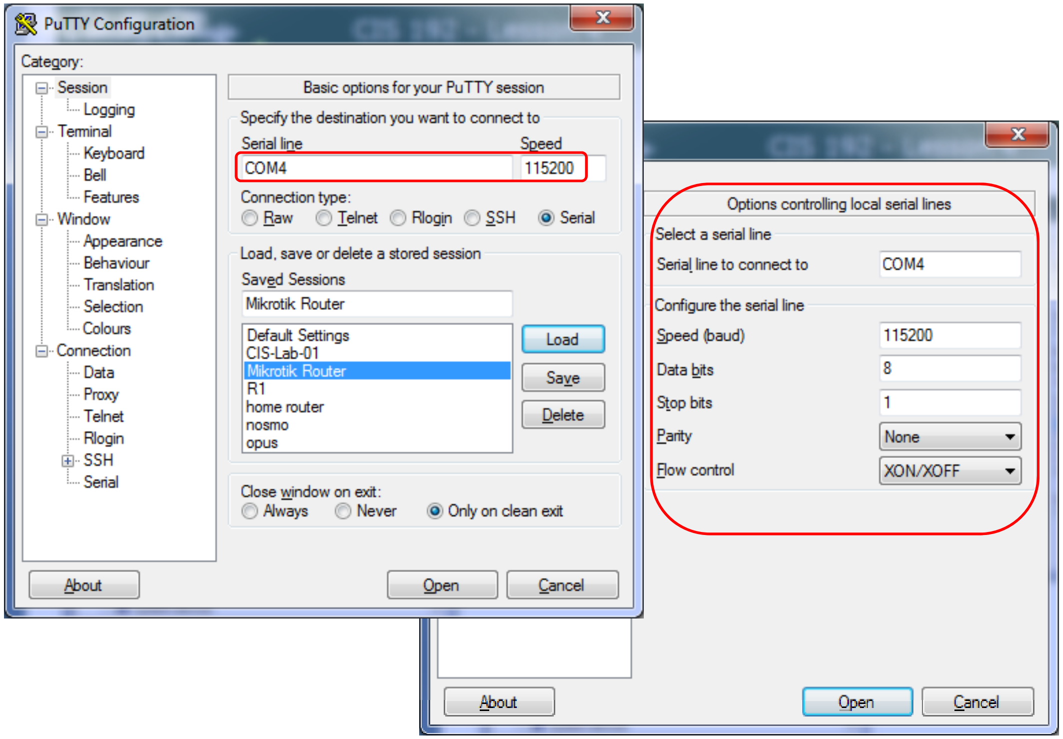
• RB/450 Routerboard	$69
• CA/150 indoor case	$19
• 24HPOW power supply	$18
• SW-1301 USB-to-serial adapter	$12

MikroTik/Routerboard – A Linux based router



- IEEE 1284.4 devices
- IEEE 1394 Bus host controllers
- Imaging devices
- Keyboards
- Mice and other pointing devices
- Modems
- Monitors
- Network adapters
- Other devices
- Portable Devices
- Ports (COM & LPT)
  - BT Port (COM10)
  - BT Port (COM11)
  - BT Port (COM12)
  - BT Port (COM13)
  - BT Port (COM14)
  - BT Port (COM20)
  - BT Port (COM21)
  - BT Port (COM22)
  - BT Port (COM6)
  - BT Port (COM7)
  - Prolific USB-to-Serial Comm Port (COM4)
- Processors
- SD host adapters
- Sound, video and game controllers
- System devices

With a USB-to-Serial adapter Putty can be used as the console



Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM4 Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Load, save or delete a stored session

Saved Sessions: Mikrotik Router

- Default Settings
- CIS-Lab-01
- Mikrotik Router
- R1
- home router
- nosmo
- opus

Close window on exit: ○ Always ○ Never ● Only on clean exit

Options controlling local serial lines

Select a serial line

Serial line to connect to: COM4

Configure the serial line

Speed (baud): 115200

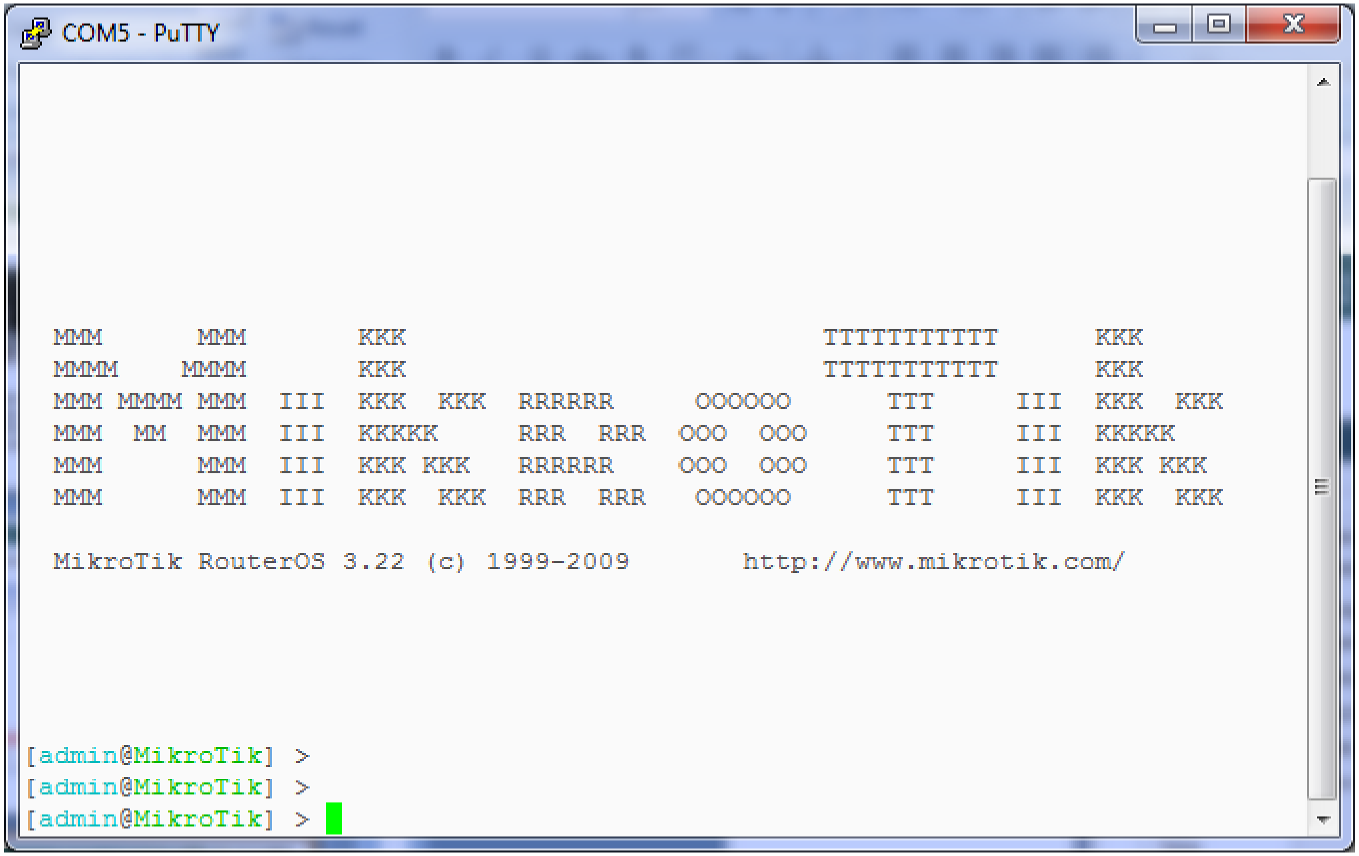
Data bits: 8

Stop bits: 1

Parity: None

Flow control: XON/XOFF

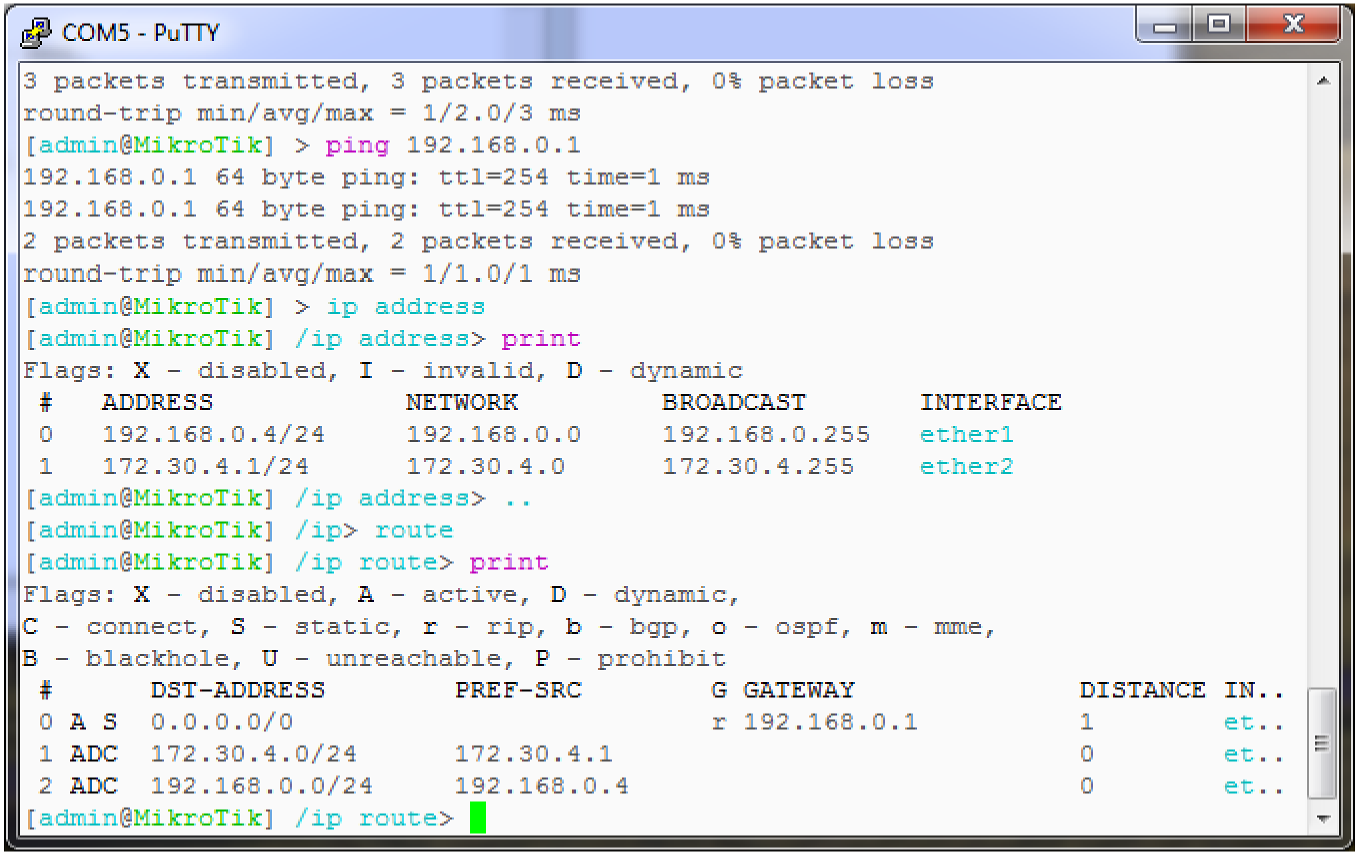
MikroTik/Routerboard – A Linux based router



```
  MMM      MMM       KKK                          TTTTTTTTTTT      KKK
  MMMM    MMMM       KKK                          TTTTTTTTTTT      KKK
  MMM MMMM MMM  III  KKK  KKK  RRRRRR     OOOOOO      TTT     III  KKK  KKK
  MMM  MM  MMM  III  KKKKK     RRR  RRR  OOO  OOO     TTT     III  KKKKK
  MMM      MMM  III  KKK KKK   RRRRRR    OOO  OOO     TTT     III  KKK KKK
  MMM      MMM  III  KKK  KKK  RRR  RRR   OOOOOO      TTT     III  KKK  KKK

  MikroTik RouterOS 3.22 (c) 1999-2009       http://www.mikrotik.com/




[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
```

MikroTik RouterOS provides their own shell and software that runs on a Linux 2.6 kernel. The admin account is initially set with no password for first time login.

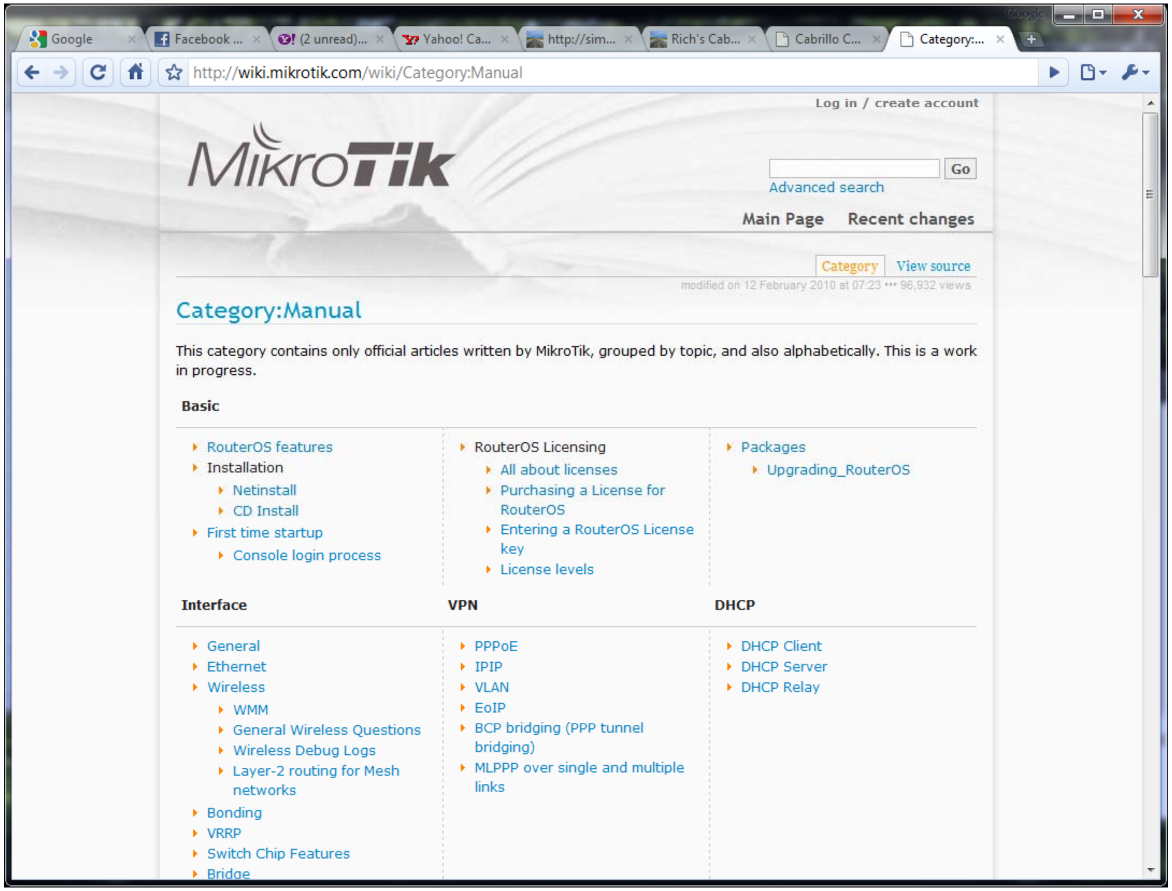
MikroTik/Routerboard – A Linux based router



```
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 1/2.0/3 ms
[admin@MikroTik] > ping 192.168.0.1
192.168.0.1 64 byte ping: ttl=254 time=1 ms
192.168.0.1 64 byte ping: ttl=254 time=1 ms
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 1/1.0/1 ms
[admin@MikroTik] > ip address
[admin@MikroTik] /ip address> print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         BROADCAST       INTERFACE
 0   192.168.0.4/24     192.168.0.0     192.168.0.255   ether1
 1   172.30.4.1/24      172.30.4.0      172.30.4.255    ether2
[admin@MikroTik] /ip address> ..
[admin@MikroTik] /ip> route
[admin@MikroTik] /ip route> print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        G GATEWAY             DISTANCE IN..
 0 A S  0.0.0.0/0                          r 192.168.0.1         1        et..
 1 ADC  172.30.4.0/24      172.30.4.1                            0        et..
 2 ADC  192.168.0.0/24     192.168.0.4                           0        et..
[admin@MikroTik] /ip route>
```

The shell lets you configure and show interfaces, routes, DHCP, etc.

MikroTik/Routerboard – A Linux based router



modified on 12 February 2010 at 07:23 ••• 96,932 views

Category:Manual

This category contains only official articles written by MikroTik, grouped by topic, and also alphabetically. This is a work in progress.

Basic

- RouterOS features
- Installation
  - Netinstall
  - CD Install
- First time startup
  - Console login process
- RouterOS Licensing
  - All about licenses
  - Purchasing a License for RouterOS
  - Entering a RouterOS License key
  - License levels
- Packages
  - Upgrading_RouterOS

Interface

- General
- Ethernet
- Wireless
  - WMM
  - General Wireless Questions
  - Wireless Debug Logs
  - Layer-2 routing for Mesh networks
- Bonding
- VRRP
- Switch Chip Features
- Bridge

VPN

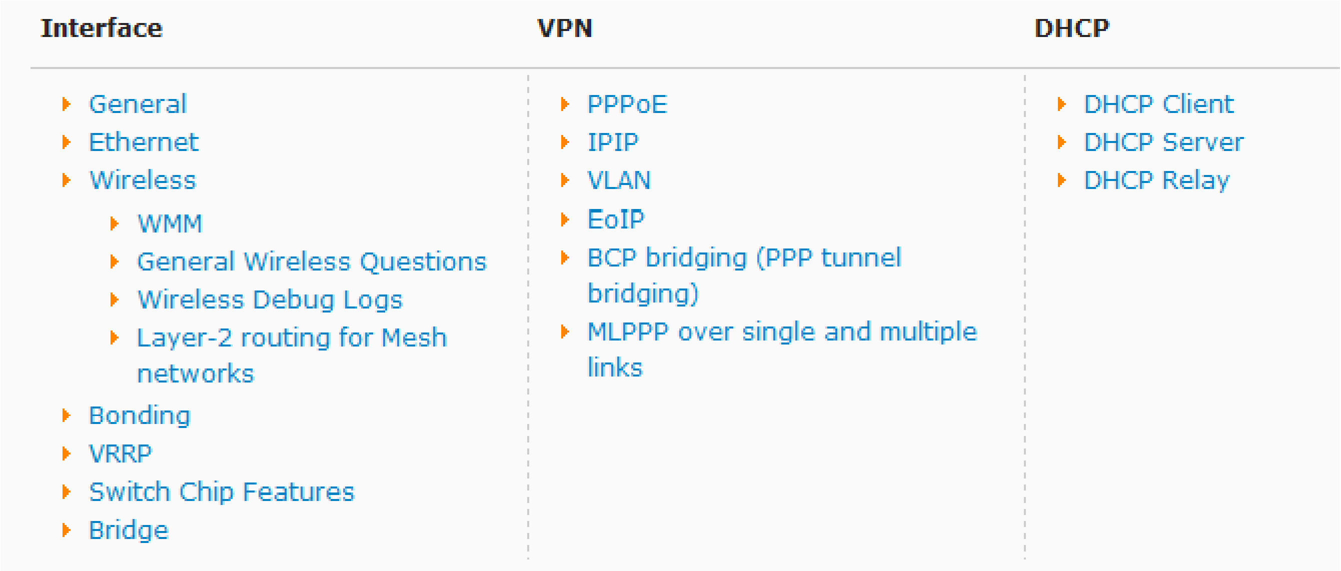
- PPPoE
- IPIP
- VLAN
- EoIP
- BCP bridging (PPP tunnel bridging)
- MLPPP over single and multiple links

DHCP

- DHCP Client
- DHCP Server
- DHCP Relay

Online wiki documentation

MikroTik/Routerboard – A Linux based router



Interface

- General
- Ethernet
- Wireless
  - WMM
  - General Wireless Questions
  - Wireless Debug Logs
  - Layer-2 routing for Mesh networks
- Bonding
- VRRP
- Switch Chip Features
- Bridge

VPN

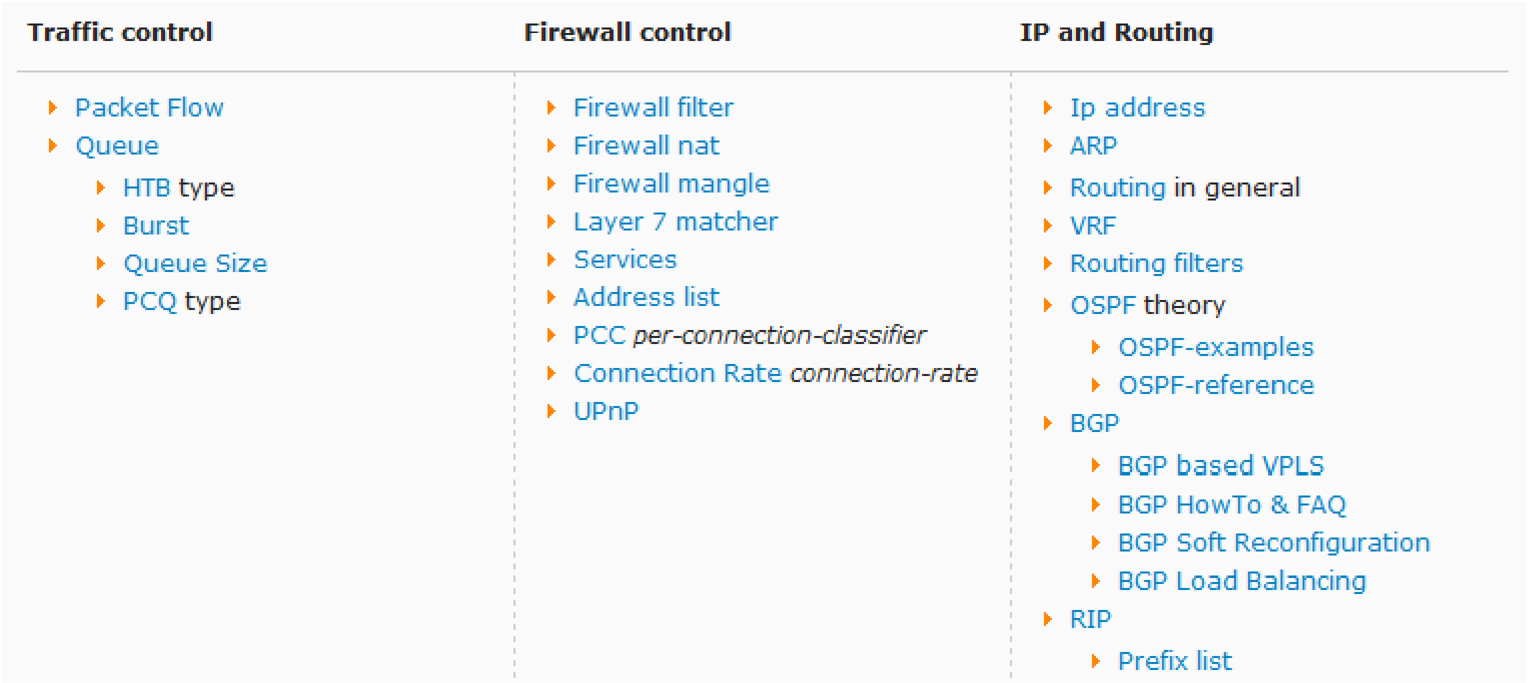
- PPPoE
- IPIP
- VLAN
- EoIP
- BCP bridging (PPP tunnel bridging)
- MLPPP over single and multiple links

DHCP

- DHCP Client
- DHCP Server
- DHCP Relay

Online wiki documentation areas

MikroTik/Routerboard – A Linux based router



Traffic control

- Packet Flow
- Queue
  - HTB type
  - Burst
  - Queue Size
  - PCQ type

Firewall control

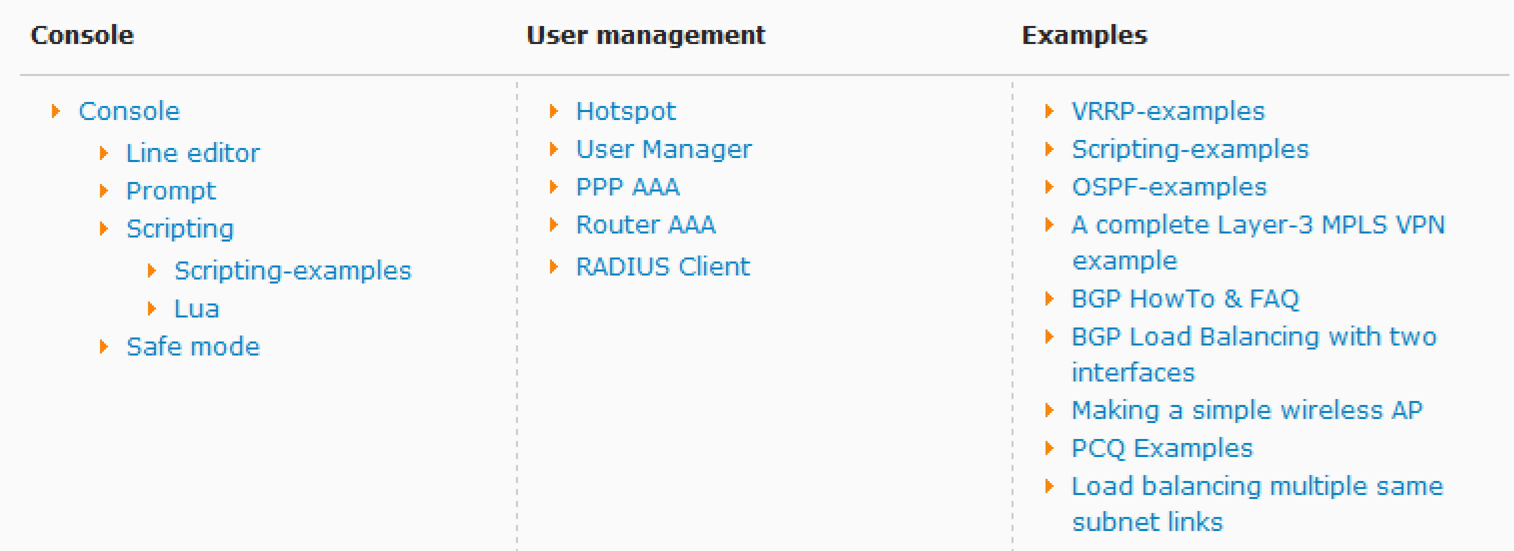
- Firewall filter
- Firewall nat
- Firewall mangle
- Layer 7 matcher
- Services
- Address list
- PCC *per-connection-classifier*
- Connection Rate *connection-rate*
- UPnP

IP and Routing

- Ip address
- ARP
- Routing in general
- VRF
- Routing filters
- OSPF theory
  - OSPF-examples
  - OSPF-reference
- BGP
  - BGP based VPLS
  - BGP HowTo & FAQ
  - BGP Soft Reconfiguration
  - BGP Load Balancing
- RIP
  - Prefix list

Online wiki documentation areas

MikroTik/Routerboard – A Linux based router



Console

- Console
  - Line editor
  - Prompt
  - Scripting
    - Scripting-examples
    - Lua
  - Safe mode

User management

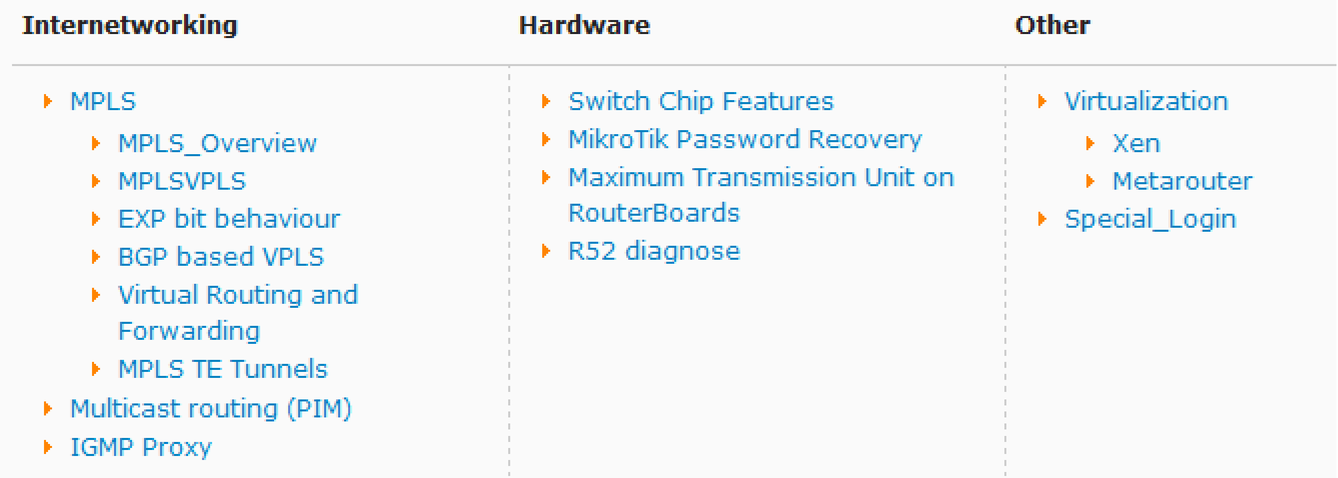
- Hotspot
- User Manager
- PPP AAA
- Router AAA
- RADIUS Client

Examples

- VRRP-examples
- Scripting-examples
- OSPF-examples
- A complete Layer-3 MPLS VPN example
- BGP HowTo & FAQ
- BGP Load Balancing with two interfaces
- Making a simple wireless AP
- PCQ Examples
- Load balancing multiple same subnet links

Online wiki documentation areas

MikroTik/Routerboard – A Linux based router



Internetworking

- MPLS
  - MPLS_Overview
  - MPLSVPLS
  - EXP bit behaviour
  - BGP based VPLS
  - Virtual Routing and Forwarding
  - MPLS TE Tunnels
- Multicast routing (PIM)
- IGMP Proxy

Hardware

- Switch Chip Features
- MikroTik Password Recovery
- Maximum Transmission Unit on RouterBoards
- R52 diagnose

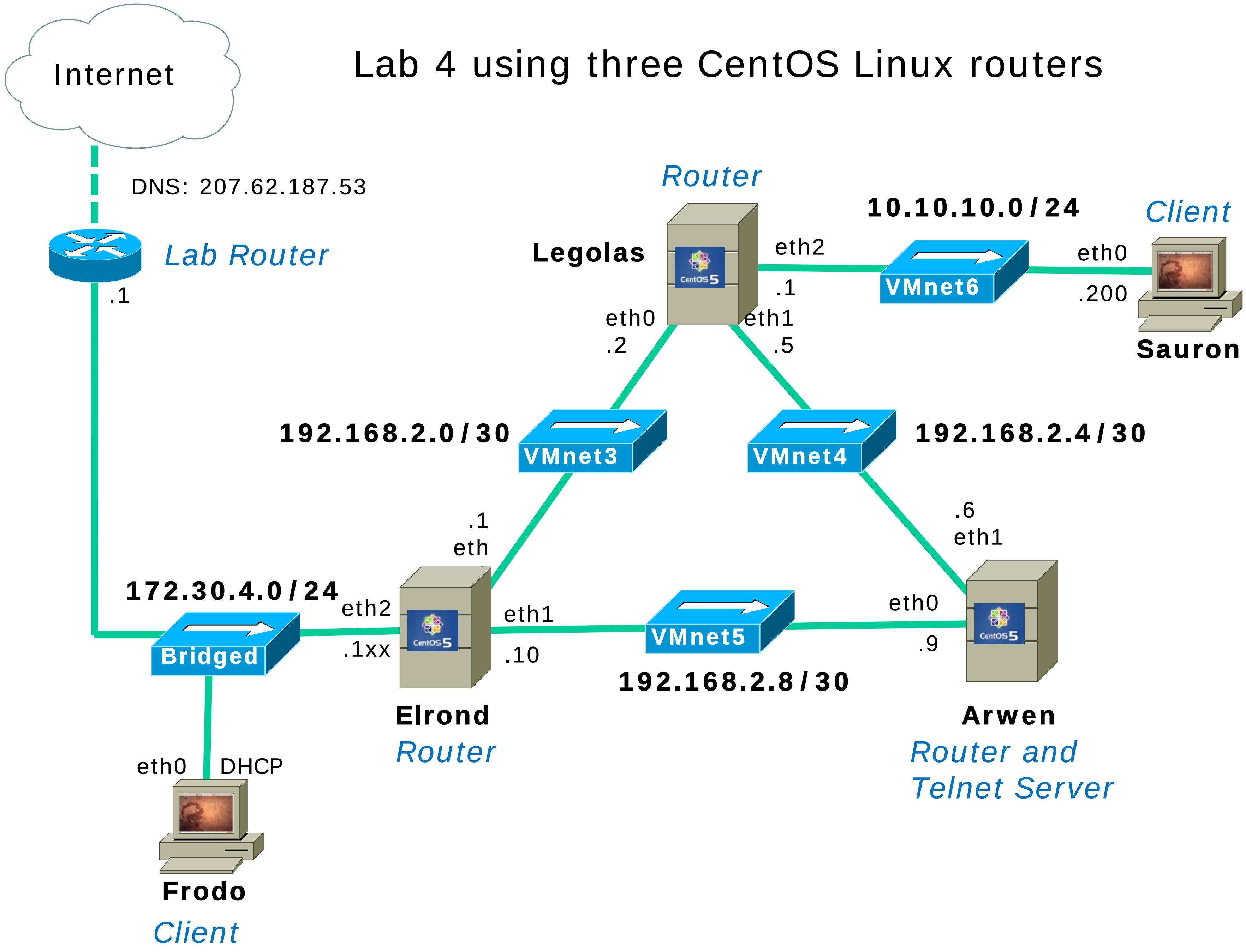
Other

- Virtualization
  - Xen
  - Metarouter
- Special_Login

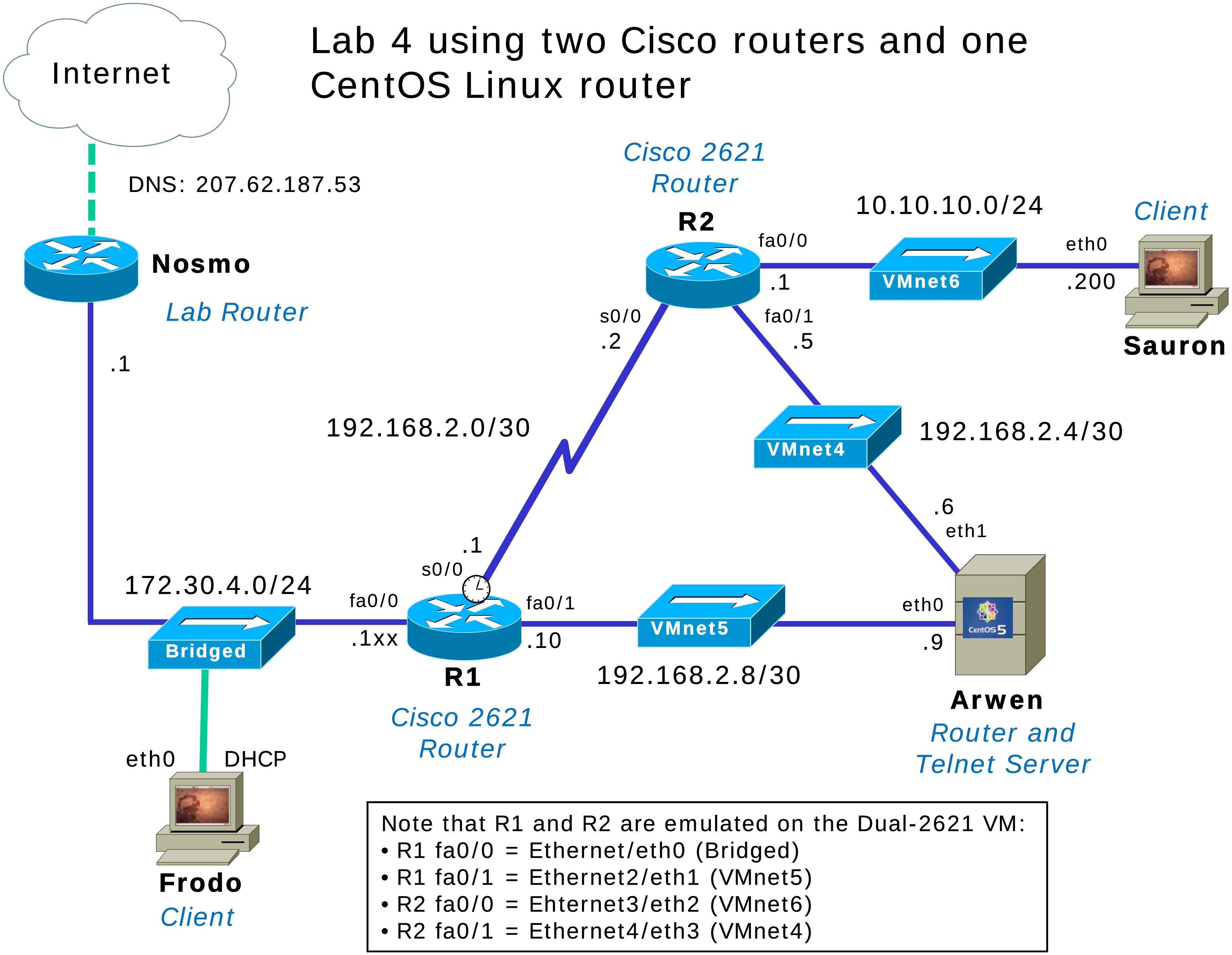
Online wiki documentation areas

Dynamips
Dynagen

Lab 4 using three CentOS Linux routers

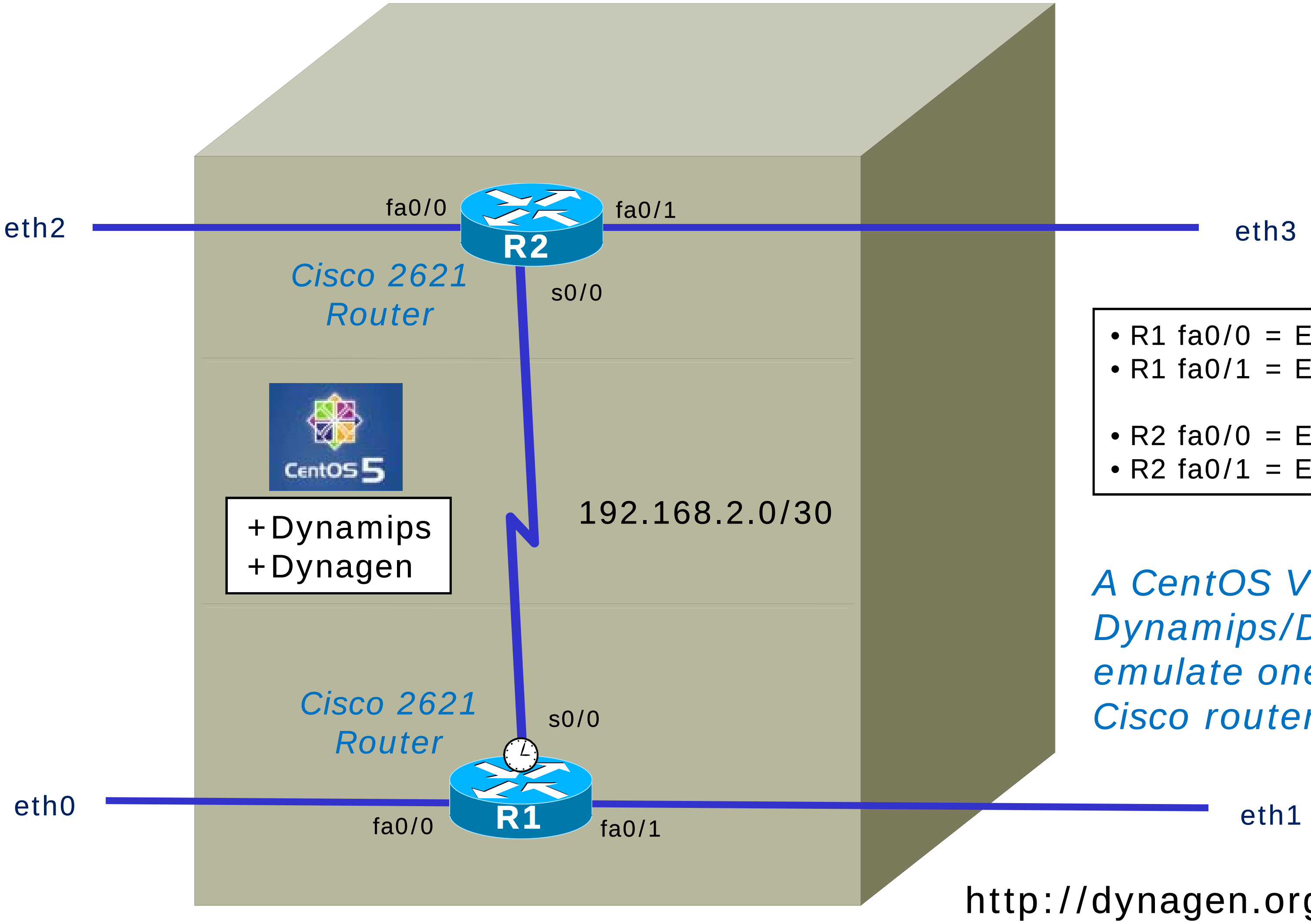


Lab 4 using two Cisco routers and one CentOS Linux router



Note that R1 and R2 are emulated on the Dual-2621 VM:
- R1 fa0/0 = Ethernet/eth0 (Bridged)
- R1 fa0/1 = Ethernet2/eth1 (VMnet5)
- R2 fa0/0 = Ehternet3/eth2 (VMnet6)
- R2 fa0/1 = Ethernet4/eth3 (VMnet4)

The Dual-c2621s VM



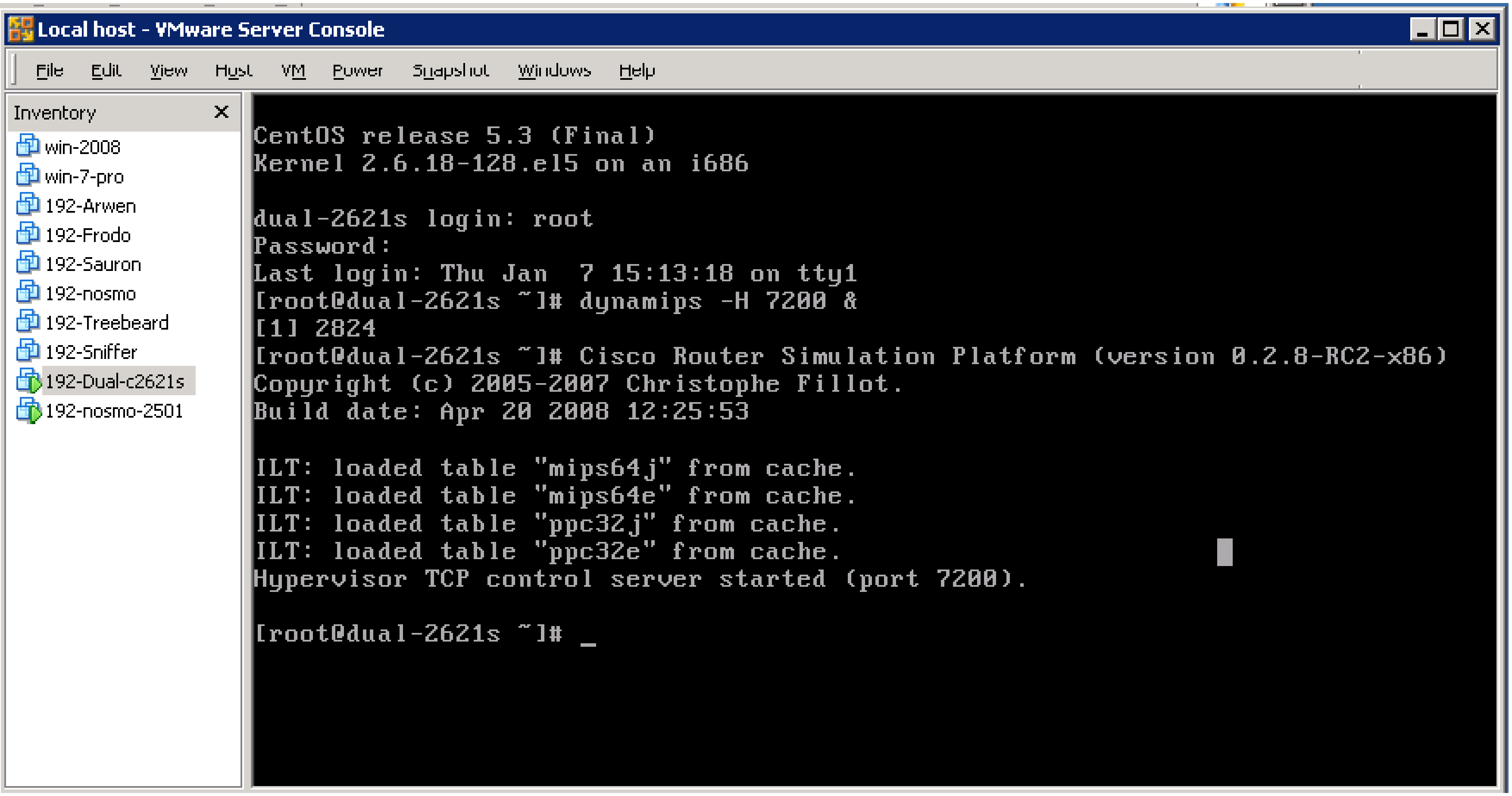
- R1 fa0/0 = Ethernet/eth0
- R1 fa0/1 = Ethernet2/eth1

- R2 fa0/0 = Ehternet3/eth2
- R2 fa0/1 = Ethernet4/eth3

A CentOS VM that runs Dynamips/Dynagen to emulate one or more Cisco routers

http://dynagen.org/tutorial.htm

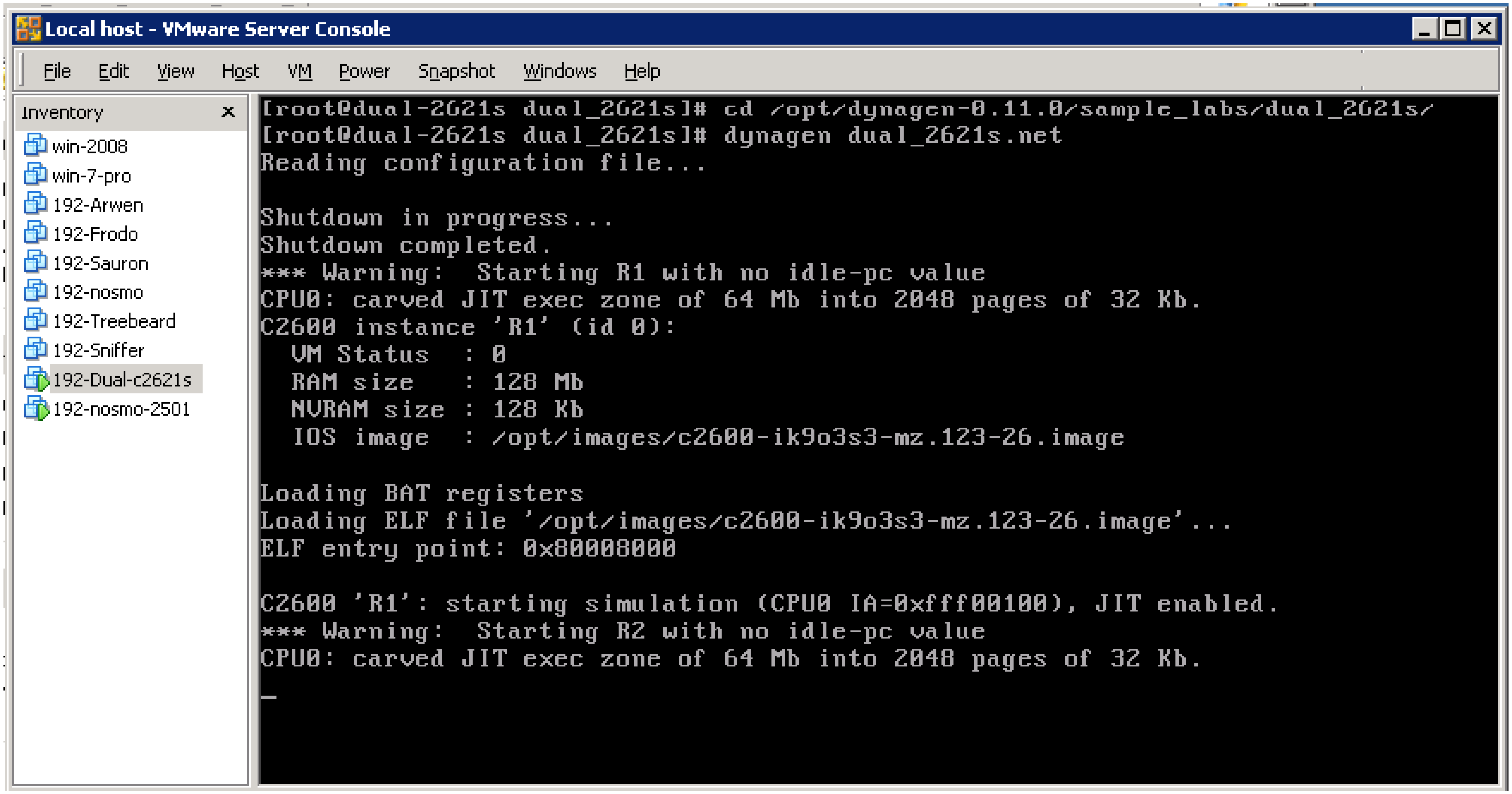
The Dual-c2621s VM



```
CentOS release 5.3 (Final)
Kernel 2.6.18-128.el5 on an i686

dual-2621s login: root
Password:
Last login: Thu Jan  7 15:13:18 on tty1
[root@dual-2621s ~]# dynamips -H 7200 &
[1] 2824
[root@dual-2621s ~]# Cisco Router Simulation Platform (version 0.2.8-RC2-x86)
Copyright (c) 2005-2007 Christophe Fillot.
Build date: Apr 20 2008 12:25:53

ILT: loaded table "mips64j" from cache.
ILT: loaded table "mips64e" from cache.
ILT: loaded table "ppc32j" from cache.
ILT: loaded table "ppc32e" from cache.
Hypervisor TCP control server started (port 7200).

[root@dual-2621s ~]# _
```

*Use **dynamips –H 7200 &** to run the Dynamips hardware emulator and listen using port 7200*

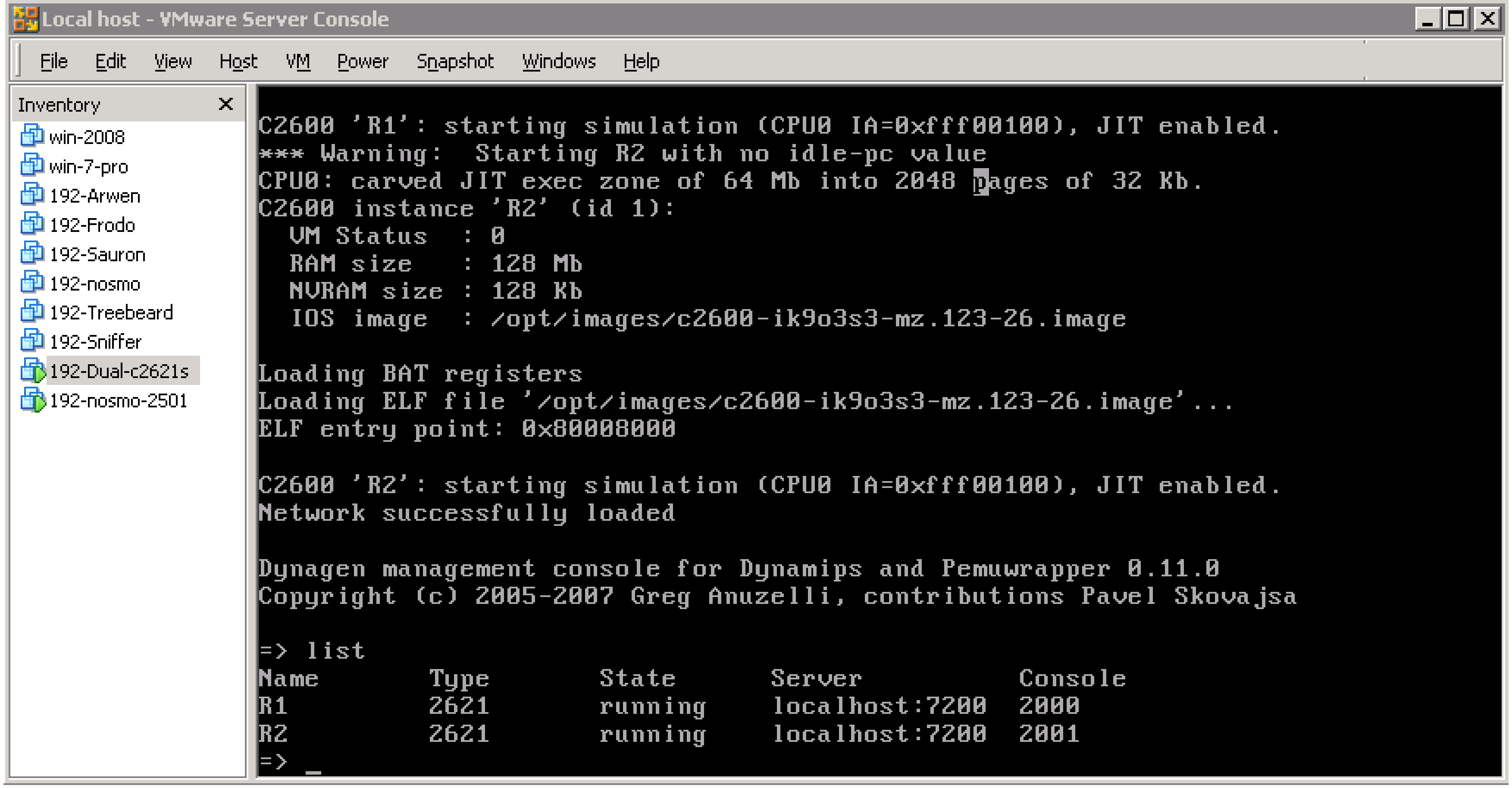
The Dual-c2621s VM



```
[root@dual-2621s dual_2621s]# cd /opt/dynagen-0.11.0/sample_labs/dual_2621s/
[root@dual-2621s dual_2621s]# dynagen dual_2621s.net
Reading configuration file...

Shutdown in progress...
Shutdown completed.
*** Warning:  Starting R1 with no idle-pc value
CPU0: carved JIT exec zone of 64 Mb into 2048 pages of 32 Kb.
C2600 instance 'R1' (id 0):
  VM Status   : 0
  RAM size    : 128 Mb
  NVRAM size  : 128 Kb
  IOS image   : /opt/images/c2600-ik9o3s3-mz.123-26.image

Loading BAT registers
Loading ELF file '/opt/images/c2600-ik9o3s3-mz.123-26.image'...
ELF entry point: 0x80008000

C2600 'R1': starting simulation (CPU0 IA=0xfff00100), JIT enabled.
*** Warning:  Starting R2 with no idle-pc value
CPU0: carved JIT exec zone of 64 Mb into 2048 pages of 32 Kb.
_
```

*Change directory to where the Dynagen configuration files are then use **dynagen dual-2621s.net** to start up two 2621 virtual routers*

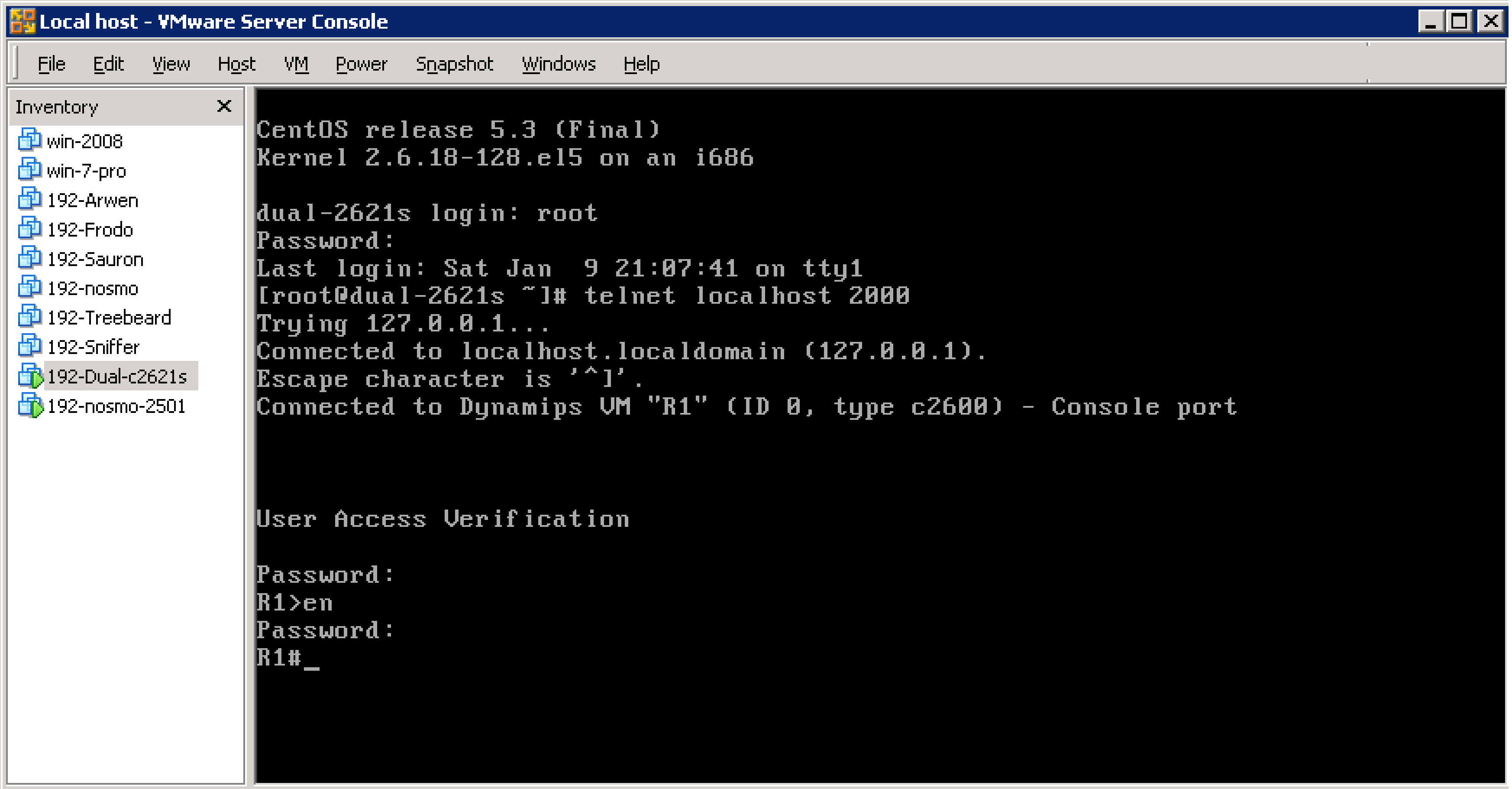
The Dual-c2621s VM



```
C2600 'R1': starting simulation (CPU0 IA=0xfff00100), JIT enabled.
*** Warning:  Starting R2 with no idle-pc value
CPU0: carved JIT exec zone of 64 Mb into 2048 pages of 32 Kb.
C2600 instance 'R2' (id 1):
  VM Status   : 0
  RAM size    : 128 Mb
  NVRAM size  : 128 Kb
  IOS image   : /opt/images/c2600-ik9o3s3-mz.123-26.image

Loading BAT registers
Loading ELF file '/opt/images/c2600-ik9o3s3-mz.123-26.image'...
ELF entry point: 0x80008000

C2600 'R2': starting simulation (CPU0 IA=0xfff00100), JIT enabled.
Network successfully loaded

Dynagen management console for Dynamips and Pemuwrapper 0.11.0
Copyright (c) 2005-2007 Greg Anuzelli, contributions Pavel Skovajsa

=> list
Name        Type        State       Server              Console
R1          2621        running     localhost:7200      2000
R2          2621        running     localhost:7200      2001
=> _
```

*Use **list** command to show the virtual routers and the ports they are listening on*

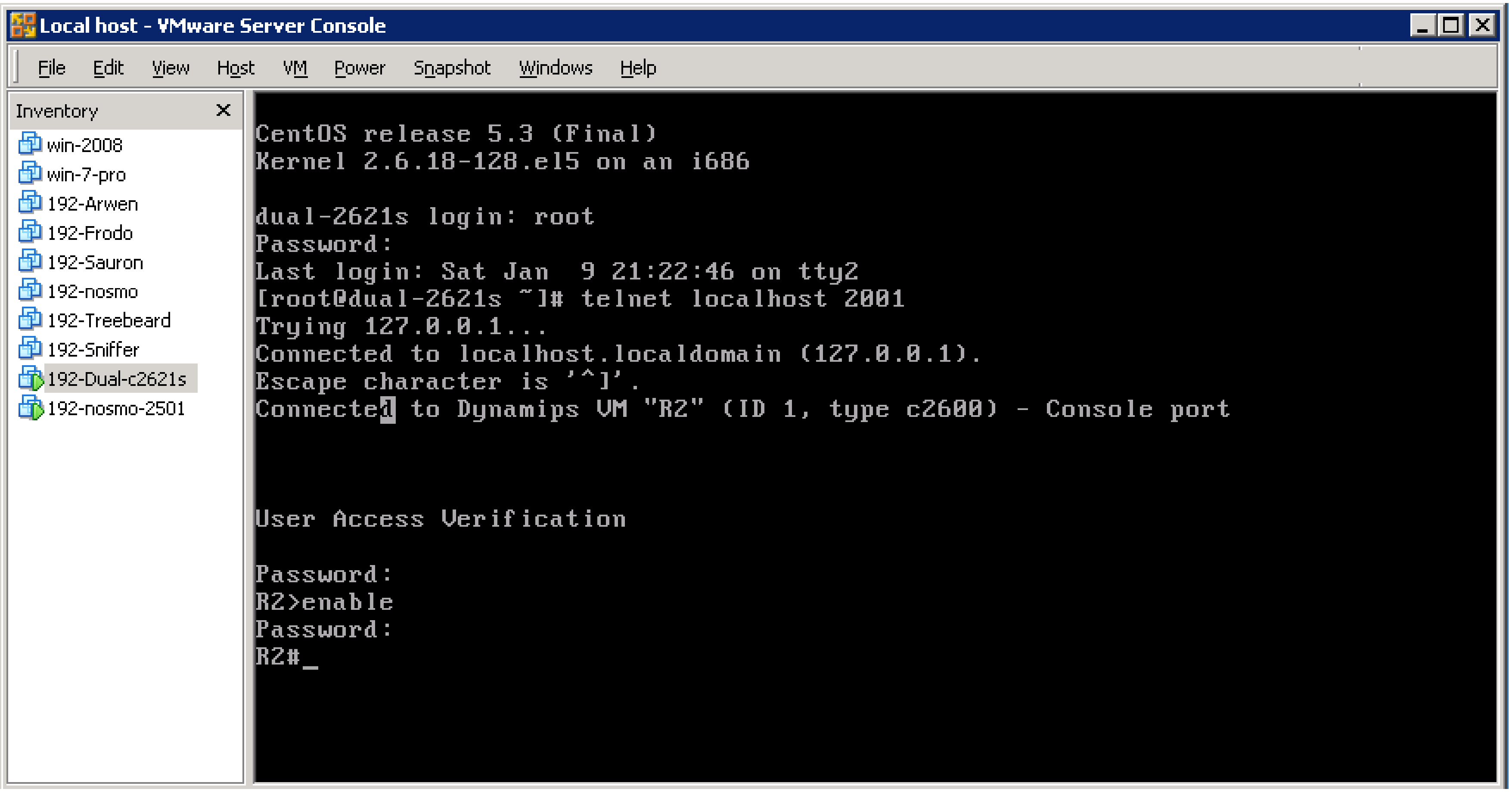
The Dual-c2621s VM



```
CentOS release 5.3 (Final)
Kernel 2.6.18-128.el5 on an i686

dual-2621s login: root
Password:
Last login: Sat Jan  9 21:07:41 on tty1
[root@dual-2621s ~]# telnet localhost 2000
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
Connected to Dynamips VM "R1" (ID 0, type c2600) - Console port



User Access Verification

Password:
R1>en
Password:
R1#_
```

*Use **telnet localhost 2000** command to get to the R1 console (using a separate virtual terminal is handy)*

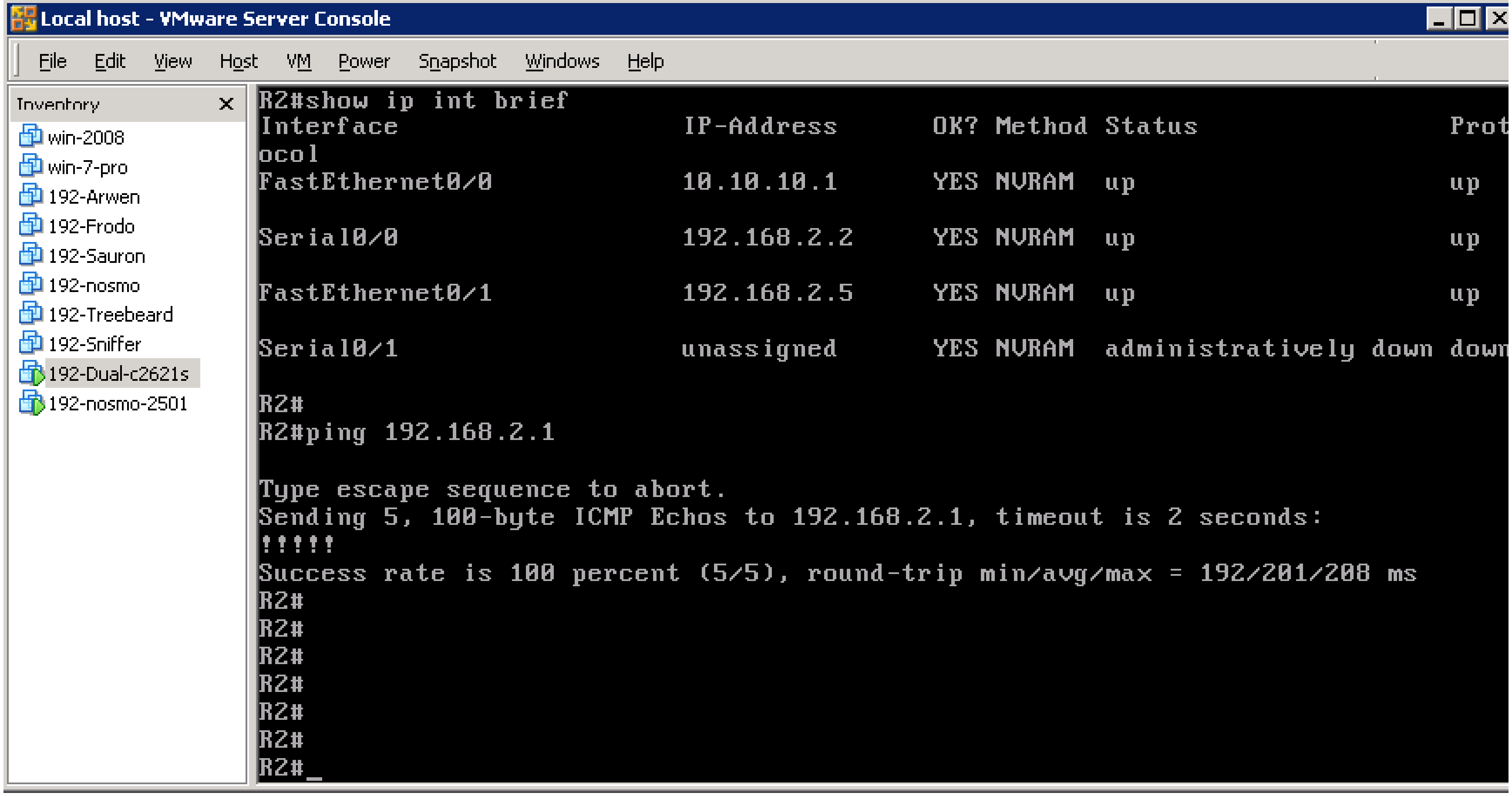
The Dual-c2621s VM



```
CentOS release 5.3 (Final)
Kernel 2.6.18-128.el5 on an i686

dual-2621s login: root
Password:
Last login: Sat Jan  9 21:22:46 on tty2
[root@dual-2621s ~]# telnet localhost 2001
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
Connected to Dynamips VM "R2" (ID 1, type c2600) - Console port



User Access Verification

Password:
R2>enable
Password:
R2#_
```

*Use **telnet localhost 2001** command to get to the R2 console (using a separate virtual terminal is handy)*

The Dual-c2621s VM



```
R2#show ip int brief
Interface                  IP-Address      OK? Method Status                Prot
ocol
FastEthernet0/0            10.10.10.1      YES NVRAM  up                    up

Serial0/0                  192.168.2.2     YES NVRAM  up                    up

FastEthernet0/1            192.168.2.5     YES NVRAM  up                    up

Serial0/1                  unassigned      YES NVRAM  administratively down down

R2#
R2#ping 192.168.2.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 192/201/208 ms
R2#
R2#
R2#
R2#
R2#
R2#
R2#_
```

You can use the Cisco IOS commands now and the interfaces can be connected to other VMs or to your physical network!

Exercise – Dynamips/Dynagen

1. Open and browse to the cis192 VMs on the D drive
2. Add the VM named 192-Dual-2621's by selecting its .vmx file
3. Disconnect the eth0 interface using VM settings to avoid duplicate IP addresses
4. Power on the 192-Dual-2621s VM
5. In tty1, start Dynamips with **dynamips –H 7200 &**
6. Start Dynagen using custom configuration file:

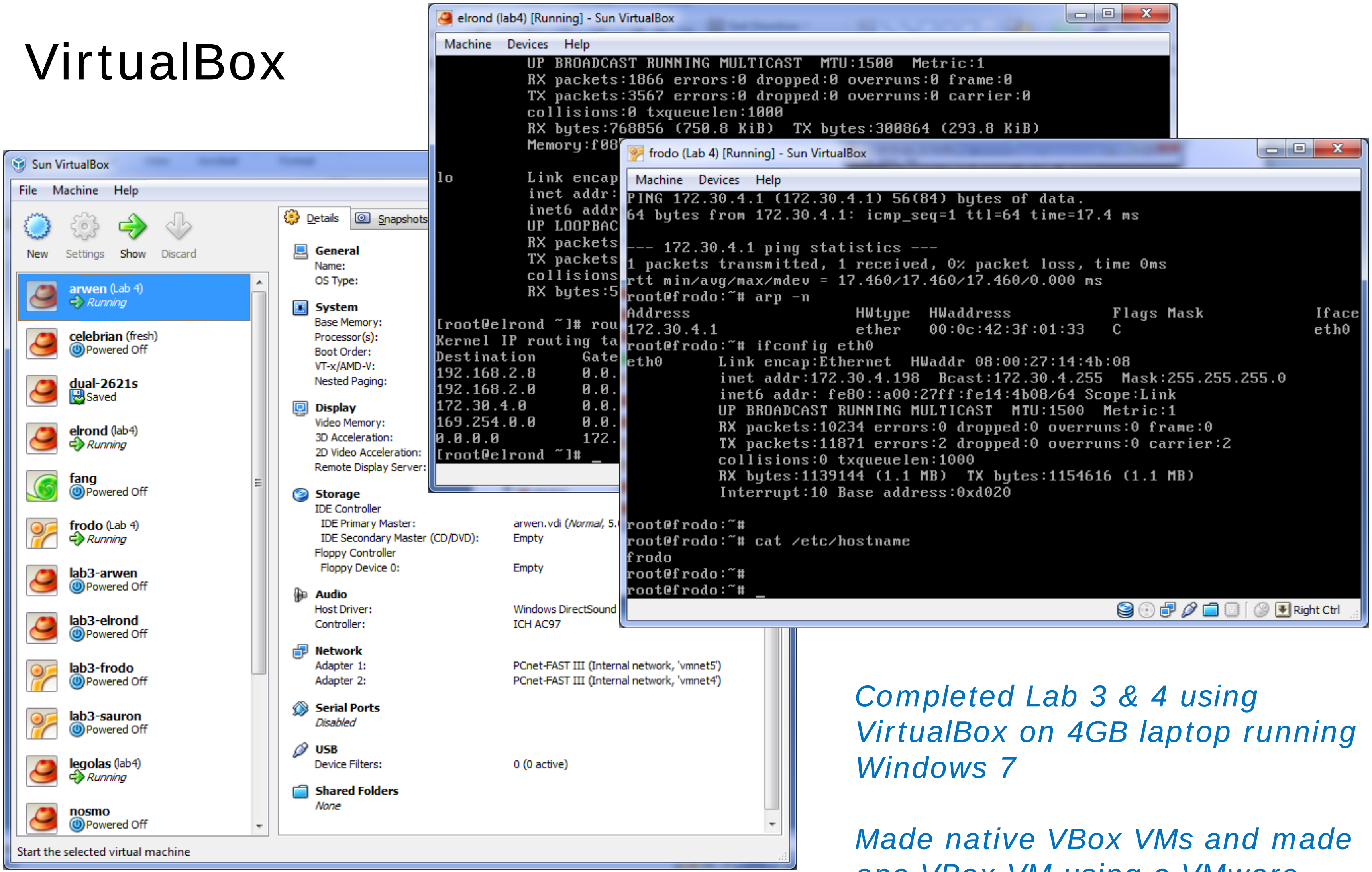
   cd /opt/dynagen-0.11.0/sample_labs/dual_2621s/

   dynagen dual_2621s.net
7. Type **list** to see the routers
8. In tty2, **telnet localhost 2000** and login to R1 (cisco/class)
9. In tty3, **telnet localhost 2001** and login to R2 (cisco/class)
10. On R1, try pinging R2 (**ping 192.168.2.2**) from R1 and showing the routing table using **show ip route**

Use tab completes!

VirtualBox Update

VirtualBox


Sun VirtualBox
File Machine Help
New Settings Show Discard
Details Snapshots
Start the selected virtual machine
elrond (lab4) [Running] - Sun VirtualBox
frodo (Lab 4) [Running] - Sun VirtualBox
Machine Devices Help
Right Ctrl

arwen (Lab 4) Running
celebrian (fresh) Powered Off
dual-2621s Saved
elrond (lab4) Running
fang Powered Off
frodo (Lab 4) Running
lab3-arwen Powered Off
lab3-elrond Powered Off
lab3-frodo Powered Off
lab3-sauron Powered Off
legolas (lab4) Running
nosmo Powered Off

General
Name:
OS Type:

System
Base Memory:
Processor(s):
Boot Order:
VT-x/AMD-V:
Nested Paging:

Display
Video Memory:
3D Acceleration:
2D Video Acceleration:
Remote Display Server:

Storage
IDE Controller
IDE Primary Master: arwen.vdi (*Normal*, 5.
IDE Secondary Master (CD/DVD): Empty
Floppy Controller
Floppy Device 0: Empty

Audio
Host Driver: Windows DirectSound
Controller: ICH AC97

Network
Adapter 1: PCnet-FAST III (Internal network, 'vmnet5')
Adapter 2: PCnet-FAST III (Internal network, 'vmnet4')

Serial Ports
Disabled

USB
Device Filters: 0 (0 active)

Shared Folders
None

```
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1866 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3567 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:768856 (750.8 KiB)  TX bytes:300864 (293.8 KiB)
          Memory:f08

lo        Link encap
          inet addr:
          inet6 addr
          UP LOOPBAC
          RX packets
          TX packets
          collisions
          RX bytes:5

[root@elrond ~]# rou
Kernel IP routing ta
Destination     Gate
192.168.2.8     0.0.
192.168.2.0     0.0.
172.30.4.0      0.0.
169.254.0.0     0.0.
0.0.0.0         172.
[root@elrond ~]# _
```

```
PING 172.30.4.1 (172.30.4.1) 56(84) bytes of data.
64 bytes from 172.30.4.1: icmp_seq=1 ttl=64 time=17.4 ms

--- 172.30.4.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 17.460/17.460/17.460/0.000 ms
root@frodo:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.30.4.1               ether   00:0c:42:3f:01:33   C                     eth0
root@frodo:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:14:4b:08
          inet addr:172.30.4.198  Bcast:172.30.4.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe14:4b08/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10234 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11871 errors:2 dropped:0 overruns:0 carrier:2
          collisions:0 txqueuelen:1000
          RX bytes:1139144 (1.1 MB)  TX bytes:1154616 (1.1 MB)
          Interrupt:10 Base address:0xd020

root@frodo:~#
root@frodo:~# cat /etc/hostname
frodo
root@frodo:~#
root@frodo:~# _
```

Completed Lab 3 & 4 using VirtualBox on 4GB laptop running Windows 7

Made native VBox VMs and made one VBox VM using a VMware Server virtual .vmdk hard drive

VirtualBox internal virtual networks

```
http://srackham.wordpress.com/cloning-and-copying-virtualbox-virtual-machines/

C:\Users\Administrator>cd C:\Program Files\Sun\VirtualBox

C:\Program Files\Sun\VirtualBox>vboxmanage modifyvm elrond --intnet3 rivendell
Sun VirtualBox Command Line Management Interface Version 3.1.4
(C) 2005-2010 Sun Microsystems, Inc.
All rights reserved.


C:\Program Files\Sun\VirtualBox>vboxmanage modifyvm elrond --intnet4 mordor
Sun VirtualBox Command Line Management Interface Version 3.1.4
(C) 2005-2010 Sun Microsystems, Inc.
All rights reserved.

c:\Program Files\Sun\VirtualBox>vboxmanage modifyvm elrond --intnet1 vmnet3
Sun VirtualBox Command Line Management Interface Version 3.1.4
(C) 2005-2010 Sun Microsystems, Inc.
All rights reserved.


c:\Program Files\Sun\VirtualBox>vboxmanage modifyvm elrond --intnet2 vmnet4
Sun VirtualBox Command Line Management Interface Version 3.1.4
(C) 2005-2010 Sun Microsystems, Inc.
All rights reserved.
```

Command line needed to create internal virtual networks which can then be selected from the GUI tool

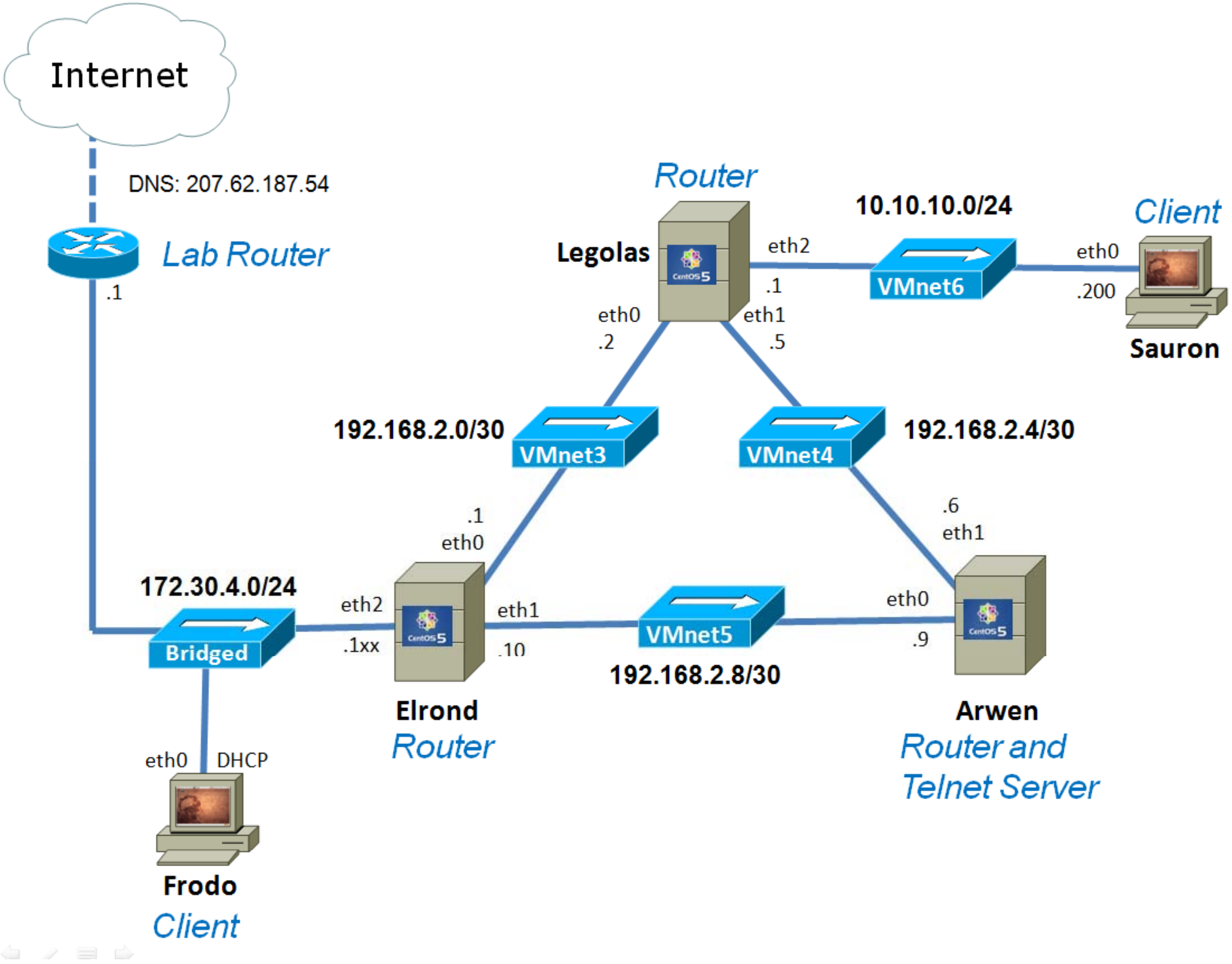
VirtualBox VM cloning

```
http://srackham.wordpress.com/cloning-and-copying-virtualbox-virtual-machines/

C:\Users\Administrator>cd C:\Program Files\Sun\VirtualBox

C:\Program Files\Sun\VirtualBox>vboxmanage clonevdi frodo.vdi sauron.vdi
Sun VirtualBox Command Line Management Interface Version 3.1.4
(C) 2005-2010 Sun Microsystems, Inc.
All rights reserved.

0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Clone hard disk created in format 'VDI'. UUID: 29106587-7426-4c00-a2f3-bbc8464b6
843
```

Command line used to clone VMs. This makes a unique copy that will not have duplicate UUID information.

Note, this is similar to copying a VMware Server VM, running the new VM, then selecting “Create”

The network used for Lab 4



VirtualBox Demo

Start up Lab 4 VMs, reconfigure Elrond eth2 to 172.30.1.125

Dynamic Routing Protocols

Routed Protocol

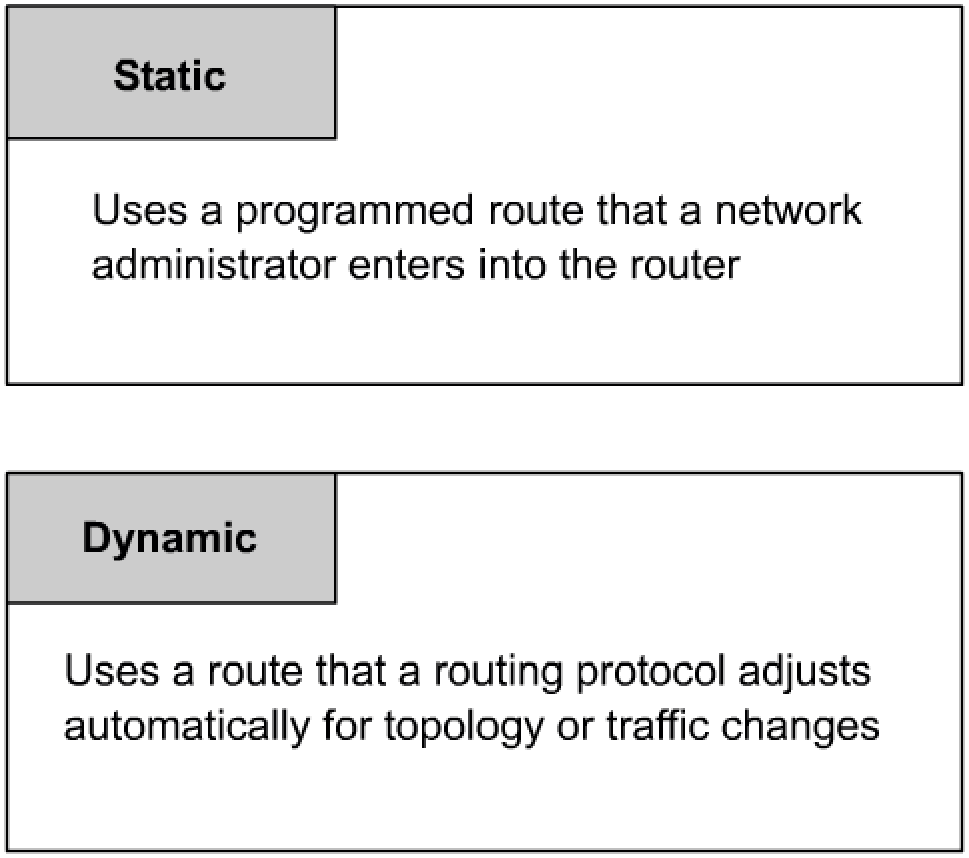
- IP is a routed protocol
- A routed protocol is a layer 3 protocol that contains network addressing information.
- This network addressing information is used by routers to determine the which interface, which next router, to forward this packet.

IP Header

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

Routing Types

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



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

Note, for Lab 3 we had to add static routes manually on the Shire hosts so that they could reach the non-directly connected Rivendell and Mordor networks.

Dynamic vs static routing

- For very small networks, static routes provide a quick and easy method to set up the routing tables.
- In Lab 3, static routes were used to reach the two inner private networks from the Shire hosts.
- As the number of networks grow and change, it becomes increasingly difficult to maintain routing tables using only static routes. With 10's or 100's of routers the setup and ongoing administration can quickly become a nightmare.
- At a certain point the investment in setting up dynamic routing becomes very attractive.
- We will set up dynamic routing in Lab 4.

Routing Protocols

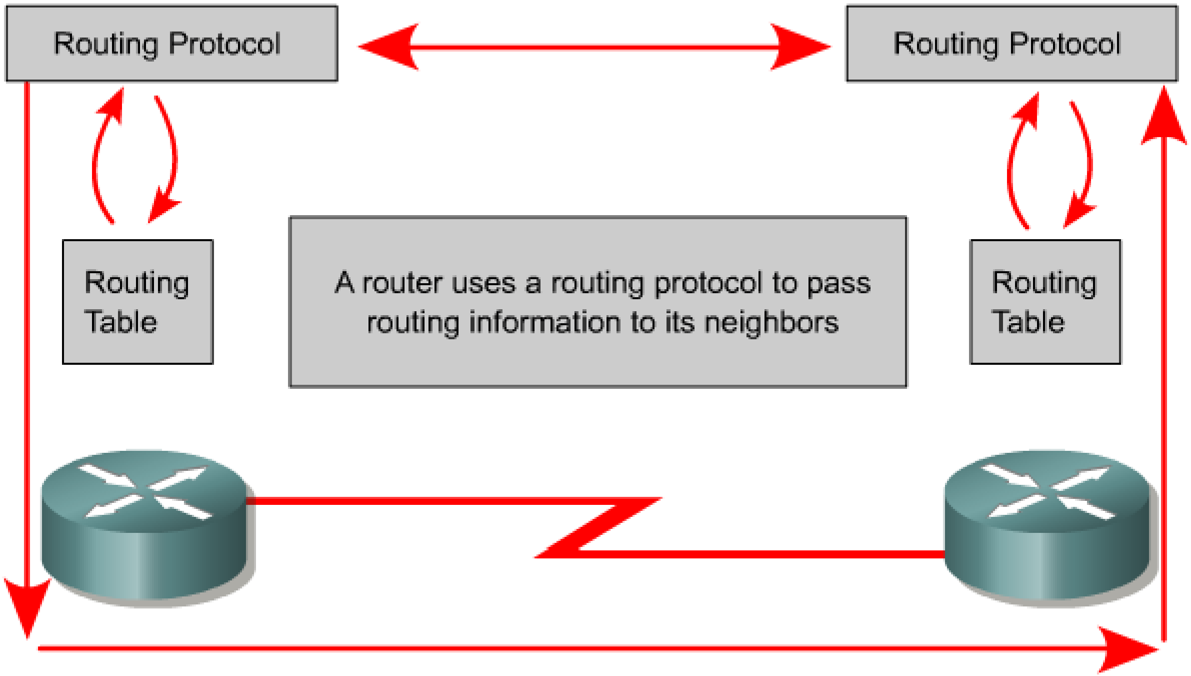
*After doing lab 3 can you imagine **manually** setting up and maintaining static routes on dozens or evens hundreds of routers!*

- 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 about in the Cabrillo Cisco networking classes.

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

Routing Protocols – CIS 82 / CST 312


Routing Protocol
Routing Protocol
Routing Table
A router uses a routing protocol to pass routing information to its neighbors
Routing Table

The whole idea is to automate making correct routing tables without the need to manually set static routes on multiple routers.

- The goal of a routing protocol is to build and maintain the routing table.
- This table contains the learned networks and associated ports for those networks.
- Routers use routing protocols to manage information received from other routers, information learned from the configuration of its own interfaces, along with manually configured routes.

Linux Implementations

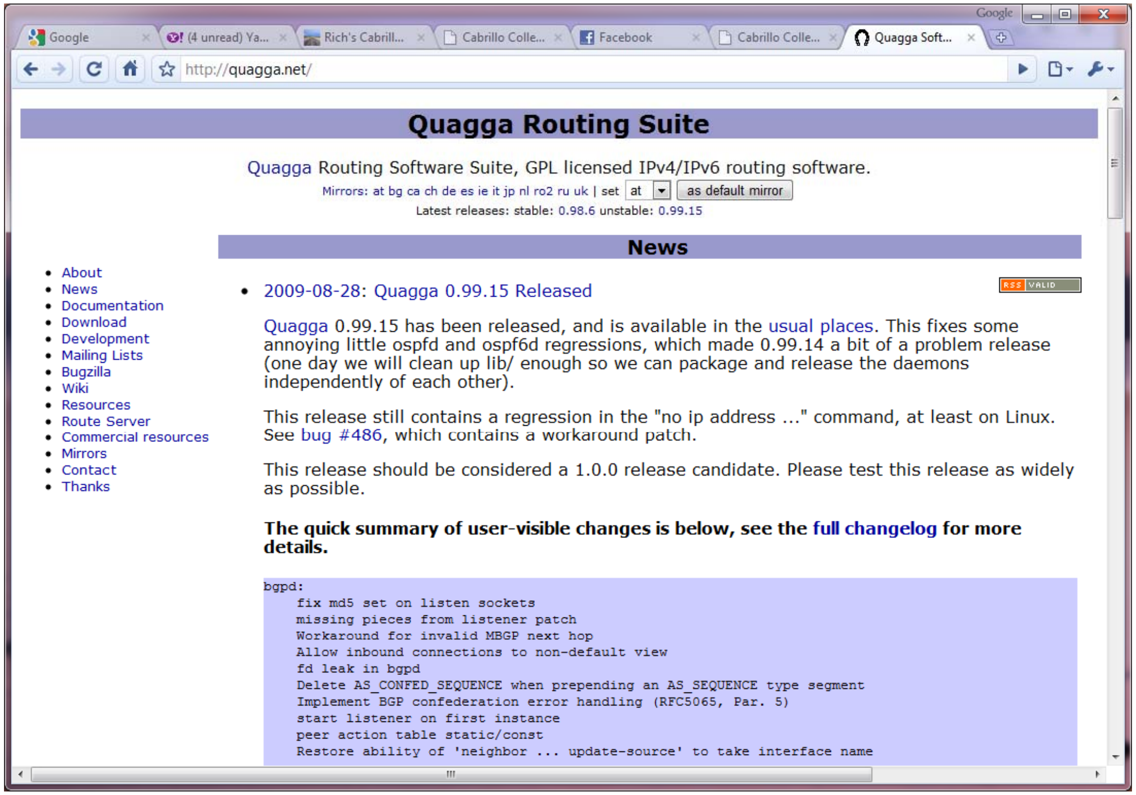
Some dynamic routing software options

- routed – an early and widespread RIPv1 implementation
- gated – multiple routing protocols (no longer open source)
- zebra – GNU licensed (BGP-4, RIPv1, RIPv2, OSPFv2)
- quagga - Fork of zebra (BGPv4+, RIPv1, RIPv2, RIPng, OSPFv2, OSPFv3)

RIPv1 is classless, uses broadcasts (RFC 1058)
RIPv2 supports CIDR (subnet masks), multicasts and authentication (RFC 2453)
RIPng = RIP Next Generation with IPv6 support (RFC 2080)

OSPF is Link-State protocol (RFC 2328 and 5340)

Quagga – A fork of GNU Zebra
http://quagga.net/



Quagga Routing Suite

Quagga Routing Software Suite, GPL licensed IPv4/IPv6 routing software.

Mirrors: at bg ca ch de es ie it jp nl ro2 ru uk | set at as default mirror

Latest releases: stable: 0.98.6 unstable: 0.99.15

- About
- News
- Documentation
- Download
- Development
- Mailing Lists
- Bugzilla
- Wiki
- Resources
- Route Server
- Commercial resources
- Mirrors
- Contact
- Thanks

News

RSS VALID

- 2009-08-28: Quagga 0.99.15 Released

Quagga 0.99.15 has been released, and is available in the usual places. This fixes some annoying little ospfd and ospf6d regressions, which made 0.99.14 a bit of a problem release (one day we will clean up lib/ enough so we can package and release the daemons independently of each other).

This release still contains a regression in the "no ip address ..." command, at least on Linux. See bug #486, which contains a workaround patch.

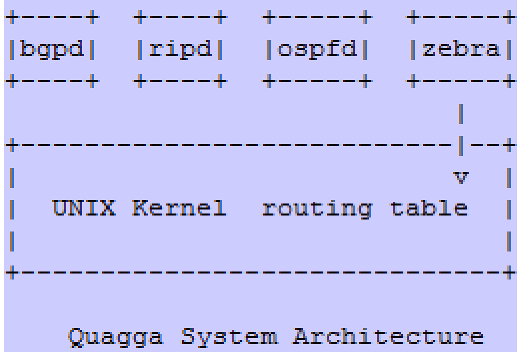
This release should be considered a 1.0.0 release candidate. Please test this release as widely as possible.

The quick summary of user-visible changes is below, see the full changelog for more details.

```
bgpd:
    fix md5 set on listen sockets
    missing pieces from listener patch
    Workaround for invalid MBGP next hop
    Allow inbound connections to non-default view
    fd leak in bgpd
    Delete AS_CONFED_SEQUENCE when prepending an AS_SEQUENCE type segment
    Implement BGP confederation error handling (RFC5065, Par. 5)
    start listener on first instance
    peer action table static/const
    Restore ability of 'neighbor ... update-source' to take interface name
```

The CLI is remarkably similar to some other routing software we study here at Cabrillo!

Note: There are a number of recipes for using Quagga in the LINUX Networking Cookbook by Carla Schroeder (O'Reilly)



```
+----+  +----+  +-----+  +-----+
|bgpd|  |ripd|  |ospfd|  |zebra|
+----+  +----+  +-----+  +-----+
                            |
+---------------------------|--+
|                           v  |
|  UNIX Kernel  routing table  |
|                              |
+------------------------------+

    Quagga System Architecture
```

Quagga – Overview

- yum installable
- Quagga has multiple daemons (services).
- They can be used like typical Linux services where you edit the configuration files in /etc and then use the **service** and **chkconfig** commands to control running the services.
- Each Quagga daemon or service (like zebra and ripd) also have individual UI shells.
- You can also use vtysh as an integrated shell for all the daemons.

With some initial testing using the Dual-2621's VM both Cisco and Quagga implementations of OSPF talk to each other – the beauty of standards!

Quagga - individual routing daemon shells

To use: telnet to localhost port 2601 for zebra or 2602 for ripd.

```
[root@legolas ~]# telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.

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


User Access Verification

Password:
legolas> en
legolas#
```

zebra service

Logging in to the shell

Enable privileged mode

Privileged mode prompt

Quagga – vtysh as an integrated Shell

Or use vtysh for an integrated shell

Show eth0 information

```
[root@legolas quagga]# vtysh

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

legolas.localdomain# sh int eth0
Interface eth0 is up, line protocol detection is disabled
  index 2 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:0c:29:7c:18:f5
  inet 192.168.2.2/30 broadcast 192.168.2.3
  inet6 fe80::20c:29ff:fe7c:18f5/64
    input packets 10923, bytes 1096902, dropped 0, multicast packets 0
    input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 0, missed 0
    output packets 8480, bytes 950760, dropped 0
    output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
    collisions 0
legolas.localdomain#
```

There is a vtysh configuration file

```
[root@legolas quagga]# cat /etc/quagga/vtysh.conf
!
! Sample configuration file for vtysh.
!
!service integrated-vtysh-config
!hostname quagga-router
!username root nopassword
!
[root@legolas quagga]#
```

Quagga – A fork of GNU Zebra

```
[root@legolas ~]# telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.

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


User Access Verification

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

K>* 0.0.0.0/0 via 192.168.2.1, eth0
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, eth0
R>* 172.30.4.0/24 [120/2] via 192.168.2.1, eth0, 03:24:42
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, 03:24:42
legolas#
```

zebra service

Show the routing table

The default gateway shows as a kernel route, each NIC is shown as directly connected, and the other routes were added using RIPv2

Quagga – zebra daemon configuration

Quagga shell

```
legolas# sh run

Current configuration:
!
hostname legolas
password <password>
log file /var/log/quagga/zebra.log
log stdout
!
interface eth0
 ipv6 nd suppress-ra
!
interface eth1
 ipv6 nd suppress-ra
!
interface eth2
 ipv6 nd suppress-ra
!
interface lo
!
interface sit0
 ipv6 nd suppress-ra
!
ip forwarding
!
!
line vty
!
end
legolas#
```

Show the running configuration in the vtysh or cat the configuration file

Linux shell

```
[root@legolas quagga]# cat /etc/quagga/zebra.conf
hostname legolas
password <password>
log stdout
log file /var/log/quagga/zebra.log
[root@legolas quagga]#
```

Suppresses IPv6 router advertisement transmissions on a local area network (Ethernet) interface.

IP forwarding is on

Quagga – ripd daemon configuration

Linux shell

```
[root@legolas ~]# cat /etc/quagga/ripd.conf
!
! Zebra configuration saved from vty
!   2009/02/25 16:36:10
!
hostname legolas(ripd)
password <password>
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
 version 2
 redistribute connected
 redistribute static
 network eth0
 network eth1
!
!line vty
!
[root@legolas ~]#
```

Quagga shell

```
legolas(ripd)# sh run

Current configuration:
!
hostname legolas(ripd)
password <password>
log file /var/log/quagga/ripd.log
!
debug rip events
debug rip zebra
!
router rip
 version 2
 redistribute connected
 redistribute static
 network eth0
 network eth1
!
line vty
!
end
legolas(ripd)#
```

*The actual configuration file and the **show running-config** output.*

Quagga – A fork of GNU Zebra

Configuration , command completion and ? help is similar to other routing software we study at Cabrillo

```
legolas# conf t
legolas(config)# hostname R1
R1(config)# hostname legolas
legolas(config)# ip
  forwarding   Turn on IP forwarding
  prefix-list  Build a prefix list
  protocol     Apply route map to PROTO
  route        Establish static routes
legolas(config)# ip forw
legolas(config)# ip forwarding
  <cr>
legolas(config)# ip forwarding
legolas(config)#
```

Enter configuration mode (note that commands and arguments may be abbreviated

Use ? to see what could come next on the command

Command completion with tab

Quagga – A fork of GNU Zebra

ripd service

Using the ripd shell to check RIP information

```
[root@legolas ~]# telnet localhost 2602
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.

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


User Access Verification

Password:
legolas(ripd)> enable
legolas(ripd)#
legolas(ripd)# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

     Network            Next Hop         Metric From            Tag Time
C(r) 10.10.10.0/24      0.0.0.0               1 self              0
R(n) 172.30.4.0/24      192.168.2.1           2 192.168.2.1       0 02:31
C(i) 192.168.2.0/30     0.0.0.0               1 self              0
C(i) 192.168.2.4/30     0.0.0.0               1 self              0
R(n) 192.168.2.8/30     192.168.2.1           2 192.168.2.1       0 02:31
legolas(ripd)#
```

Show routing table

Seeing RIP routes indicates RIP is working between routers

Quagga – Some RIP troubleshooting

ripd service

```
legolas(ripd)# show ip rip status
Routing Protocol is "rip"
  Sending updates every 30 seconds with +/-50%, next due in 14 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Default redistribution metric is 1
  Redistributing: connected static
  Default version control: send version 2, receive any version
    Interface        Send  Recv   Key-chain
    eth0             2     1 2
    eth1             2     1 2
  Routing for Networks:
    eth0
    eth1
  Routing Information Sources:
    Gateway          BadPackets BadRoutes  Distance Last Update
    192.168.2.1              0         0       120   00:00:14
    192.168.2.6            481         0       120   00:00:11
  Distance: (default is 120)
legolas(ripd)#
```

If your routing table is not getting any RIP routes then check the rip status. Any BadPackets indicate the incoming RIP updates are being ignored!

Quagga – Some RIP troubleshooting

```
[root@legolas ~]# cat /etc/quagga/ripd.conf
!
! Zebra configuration saved from vty
!   2009/02/25 16:36:10
!
hostname legolas(ripd)
password <password>
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
!
[root@legolas ~]# service ripd restart
Shutting down ripd:                                        [  OK  ]
Starting ripd:                                             [  OK  ]
```

The BadPackets were caused by unauthenticated routing updates

The fix: If you are not going to authenticate incoming updates then add this to the configuration file or the routing tables will never update

Restart service if changes made to configuration file

Quagga – Some RIP troubleshooting

After changing the ripd configuration file, restart the service so the changes will take effect

```
[root@legolas ~]# service ripd restart
Shutting down ripd:                                        [  OK  ]
Starting ripd:                                             [  OK  ]
```

And login again to the shell to check the RIP status

```
[root@legolas ~]# telnet localhost 2602
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.

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


User Access Verification

Password:
legolas(ripd)> en
legolas(ripd)#
```

Quagga – Some RIP troubleshooting

```
legolas(ripd)# sh ip rip status
Routing Protocol is "rip"
  Sending updates every 30 seconds with +/-50%, next due in 29 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Default redistribution metric is 1
  Redistributing: connected static
  Default version control: send version 2, receive any version
    Interface        Send  Recv   Key-chain
    eth0             2     1 2
    eth1             2     1 2
  Routing for Networks:
    eth0
    eth1
  Routing Information Sources:
    Gateway          BadPackets BadRoutes  Distance Last Update
    192.168.2.1              0         0       120   00:00:03
    192.168.2.6              0         0       120   00:00:02
  Distance: (default is 120)
legolas(ripd)#
```

Now RIP routes will be inserted into the routing table

Quagga – Debugging

```
legolas(ripd)# debug rip zebra
legolas(ripd)# debug rip event
```

Debugging shows RIP events is log file

```
[root@legolas ~]# tail -f /var/log/quagga/ripd.log
2009/02/26 09:12:56 RIP: RECV packet from 192.168.2.1 port 520 on eth0
2009/02/26 09:13:04 RIP: update timer fire!
2009/02/26 09:13:04 RIP: SEND UPDATE to eth0 ifindex 2
2009/02/26 09:13:04 RIP: multicast announce on eth0
2009/02/26 09:13:04 RIP: update routes on interface eth0 ifindex 2
2009/02/26 09:13:04 RIP: SEND to  224.0.0.9.520
2009/02/26 09:13:04 RIP: SEND UPDATE to eth1 ifindex 3
2009/02/26 09:13:04 RIP: multicast announce on eth1
2009/02/26 09:13:04 RIP: update routes on interface eth1 ifindex 3
2009/02/26 09:13:04 RIP: SEND to  224.0.0.9.520
2009/02/26 09:13:24 RIP: RECV packet from 192.168.2.6 port 520 on eth1
2009/02/26 09:13:30 RIP: update timer fire!
2009/02/26 09:13:30 RIP: SEND UPDATE to eth0 ifindex 2
2009/02/26 09:13:30 RIP: multicast announce on eth0
2009/02/26 09:13:30 RIP: update routes on interface eth0 ifindex 2
< snipped >
```

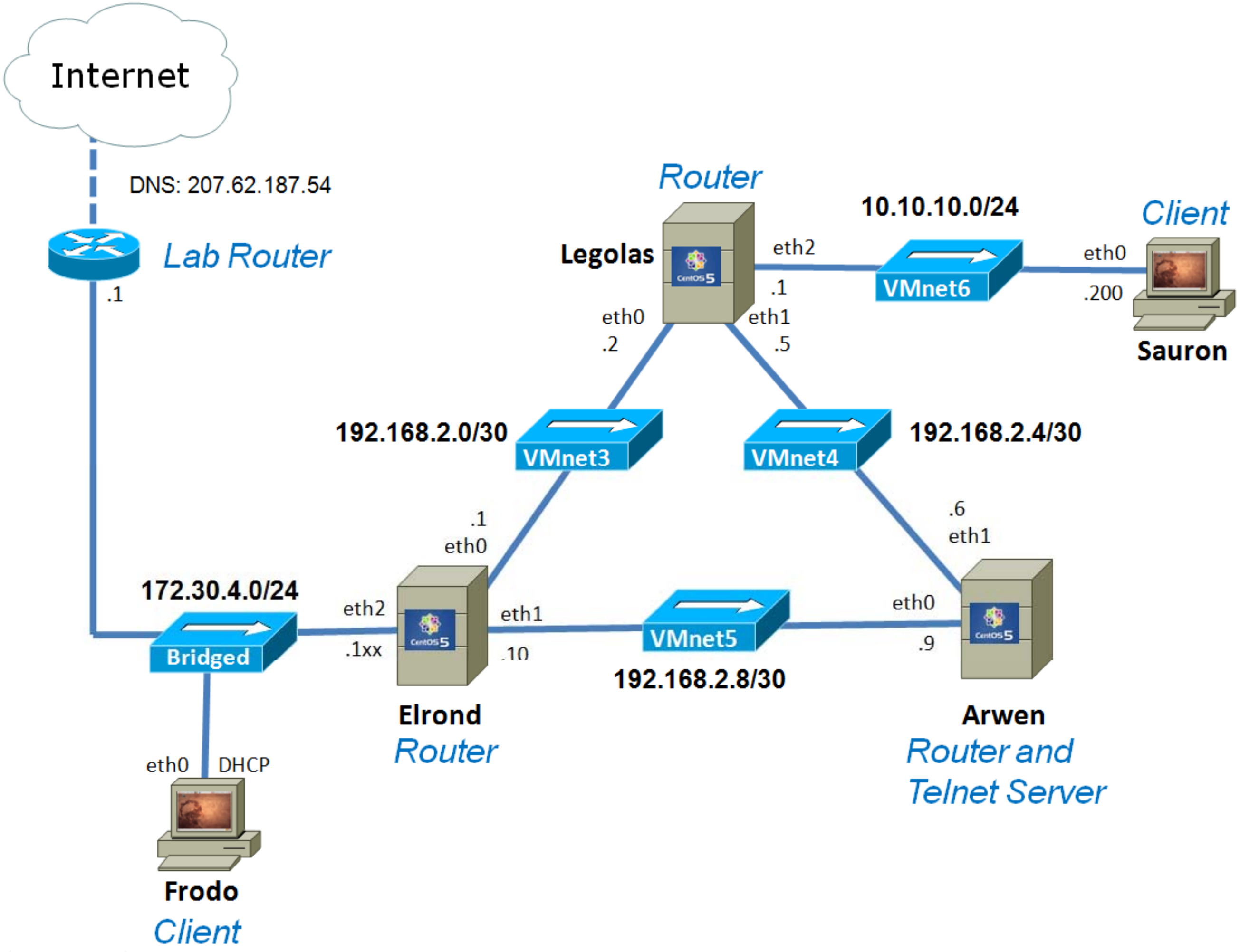
***-f** option on the **tail** command shows real-time additions to the log. Use Ctrl-C to end*

Skills needed for Lab 4!

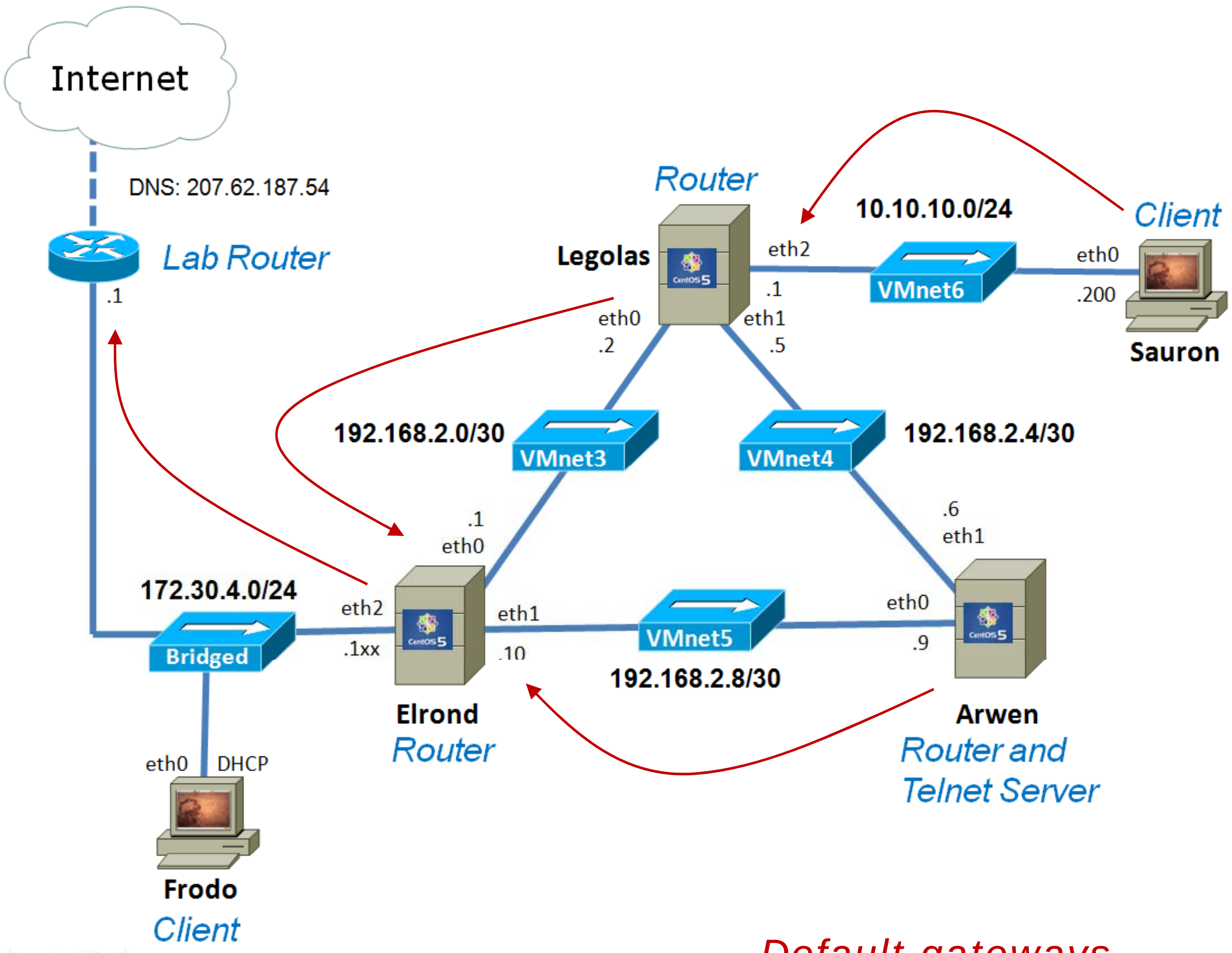
- Adding NICs
- Changing VMware host memory usage
- Cabling NICs
- Getting the graphical desktop
- Modifying the firewall
- Changing SELinux mode
- Installing software
- Managing daemons
- Using Sniffer VM

Lab 4 is due in two weeks. There is an extra credit portion

The network used for Lab 4

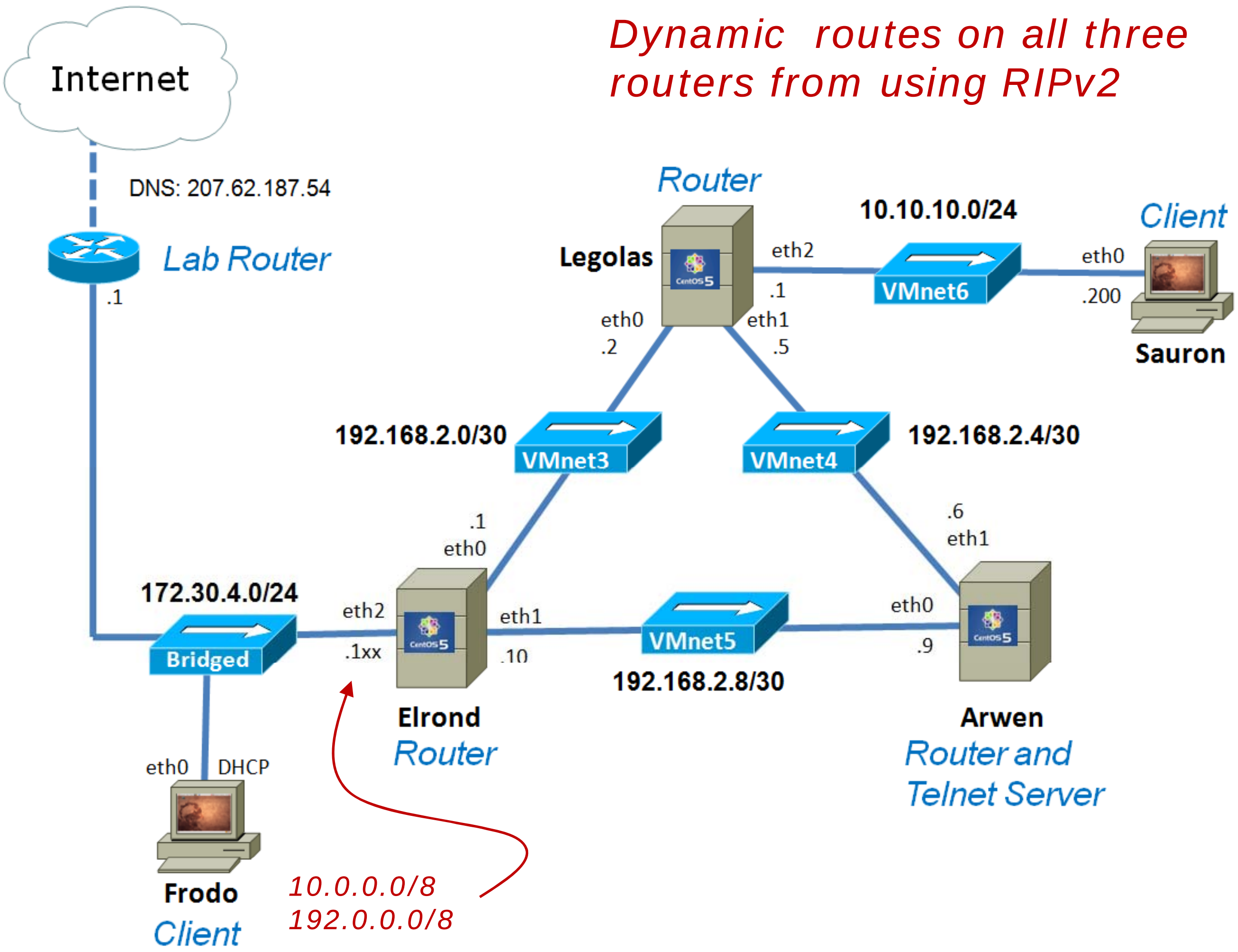


The network used for Lab 4



Default gateways

The network used for Lab 4



Dynamic routes on all three routers from using RIPv2

10.0.0.0/8
192.0.0.0/8

Static routes on Frodo

Adding another NIC

(Without going to Fry's)

- Use the **Add Hardware Wizard** to add new hardware, like NICs, to your VMs

- The VM needs to be powered off

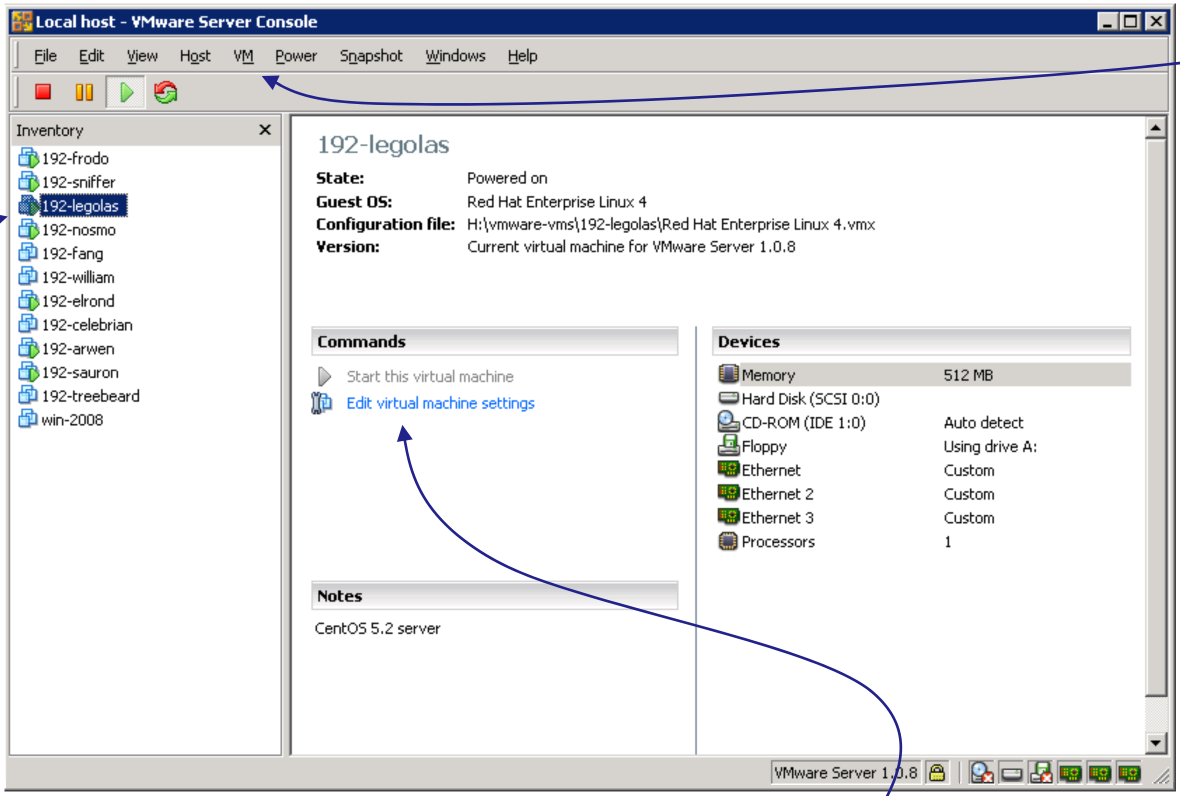
- VMware calls the NIC an **Ethernet Adapter**

- Available from **Virtual Machine Settings** dialog box

Getting to VM Settings Dialog Box

1) Use VM menu and select Settings...

2) Right click on VM and select Settings...



192-legolas

State:	Powered on
Guest OS:	Red Hat Enterprise Linux 4
Configuration file:	H:\vmware-vms\192-legolas\Red Hat Enterprise Linux 4.vmx
Version:	Current virtual machine for VMware Server 1.0.8

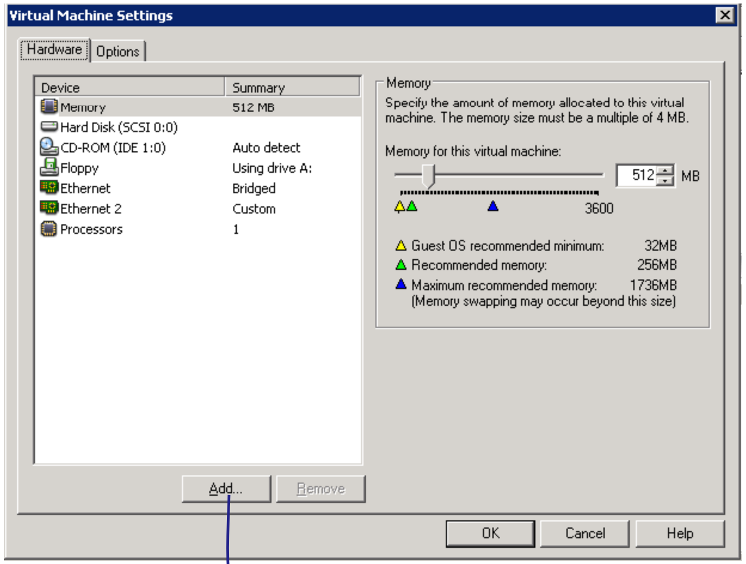
Notes: CentOS 5.2 server

Devices	
Memory	512 MB
Hard Disk (SCSI 0:0)	
CD-ROM (IDE 1:0)	Auto detect
Floppy	Using drive A:
Ethernet	Custom
Ethernet 2	Custom
Ethernet 3	Custom
Processors	1

3) Click on Edit virtual machine settings link

Adding NIC with Add Hardware Wizard

(Without going to Fry's)



Virtual Machine Settings

Hardware | Options

Device	Summary
Memory	512 MB
Hard Disk (SCSI 0:0)	
CD-ROM (IDE 1:0)	Auto detect
Floppy	Using drive A:
Ethernet	Bridged
Ethernet 2	Custom
Processors	1

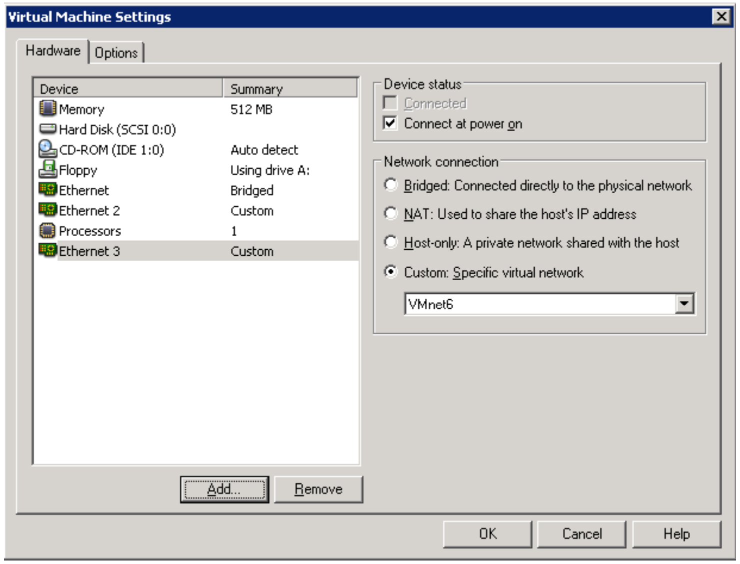
Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine: 512 MB

3600

△ Guest OS recommended minimum:	32MB
△ Recommended memory:	256MB
▲ Maximum recommended memory: (Memory swapping may occur beyond this size)	1736MB



Virtual Machine Settings

Hardware | Options

Device	Summary
Memory	512 MB
Hard Disk (SCSI 0:0)	
CD-ROM (IDE 1:0)	Auto detect
Floppy	Using drive A:
Ethernet	Bridged
Ethernet 2	Custom
Processors	1
Ethernet 3	Custom

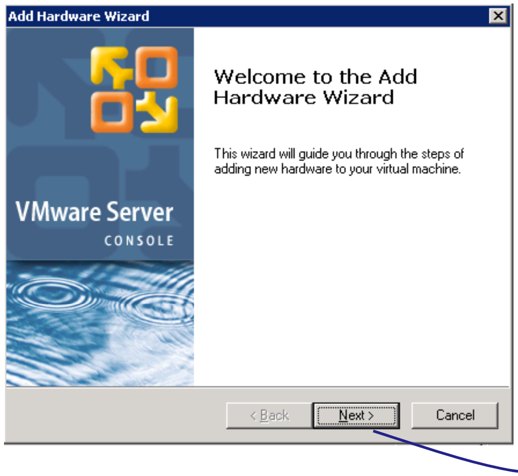
Device status

- □ Connected
- ☑ Connect at power on

Network connection

- ○ Bridged: Connected directly to the physical network
- ○ NAT: Used to share the host's IP address
- ○ Host-only: A private network shared with the host
- ● Custom: Specific virtual network

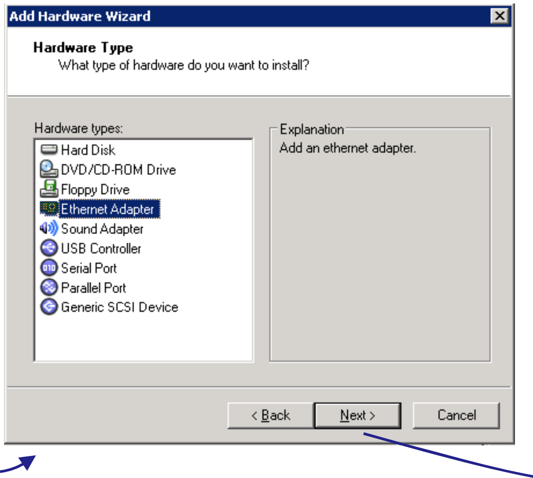
VMnet6



Add Hardware Wizard

Welcome to the Add Hardware Wizard

This wizard will guide you through the steps of adding new hardware to your virtual machine.



Add Hardware Wizard

Hardware Type

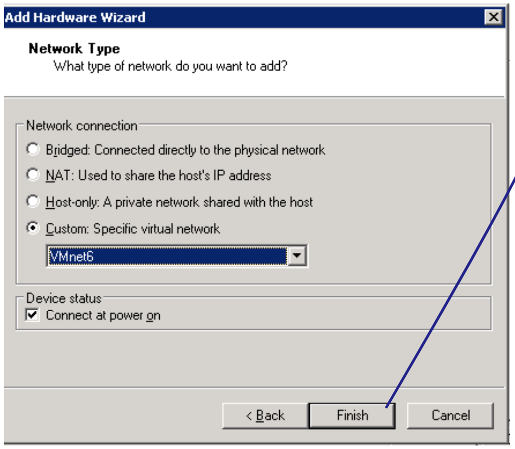
What type of hardware do you want to install?

Hardware types:

- Hard Disk
- DVD/CD-ROM Drive
- Floppy Drive
- Ethernet Adapter
- Sound Adapter
- USB Controller
- Serial Port
- Parallel Port
- Generic SCSI Device

Explanation

Add an ethernet adapter.



Add Hardware Wizard

Network Type

What type of network do you want to add?

Network connection

- ○ Bridged: Connected directly to the physical network
- ○ NAT: Used to share the host's IP address
- ○ Host-only: A private network shared with the host
- ● Custom: Specific virtual network

VMnet6

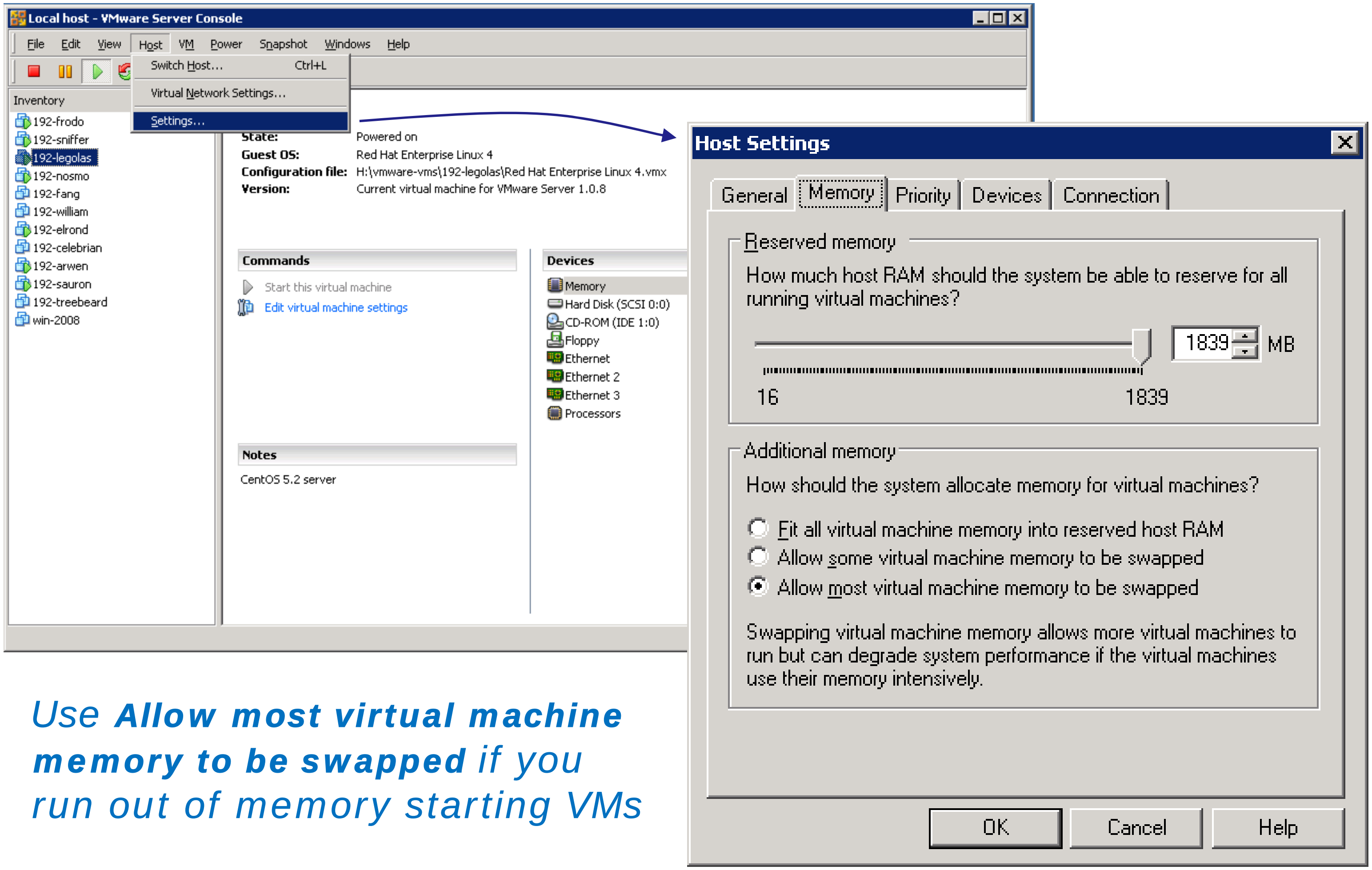
Device status

- ☑ Connect at power on

Exercise

1. Shut down Legolas if it is running
2. Add a third NIC
3. Connect it initially to VMnet6 (this is arbitrary and can be changed later when re-cabling)

VMware Host Memory Usage



State: Powered on
Guest OS: Red Hat Enterprise Linux 4
Configuration file: H:\vmware-vms\192-legolas\Red Hat Enterprise Linux 4.vmx
Version: Current virtual machine for VMware Server 1.0.8

Notes: CentOS 5.2 server

Reserved memory

How much host RAM should the system be able to reserve for all running virtual machines?

1839 MB

Additional memory

How should the system allocate memory for virtual machines?

○ Fit all virtual machine memory into reserved host RAM
○ Allow some virtual machine memory to be swapped
● Allow most virtual machine memory to be swapped

Swapping virtual machine memory allows more virtual machines to run but can degrade system performance if the virtual machines use their memory intensively.

*Use **Allow most virtual machine memory to be swapped** if you run out of memory starting VMs*

Exercise

1. Check your VMware host settings to show your current memory allocation setting.
2. Don't change now

Cabling NICs

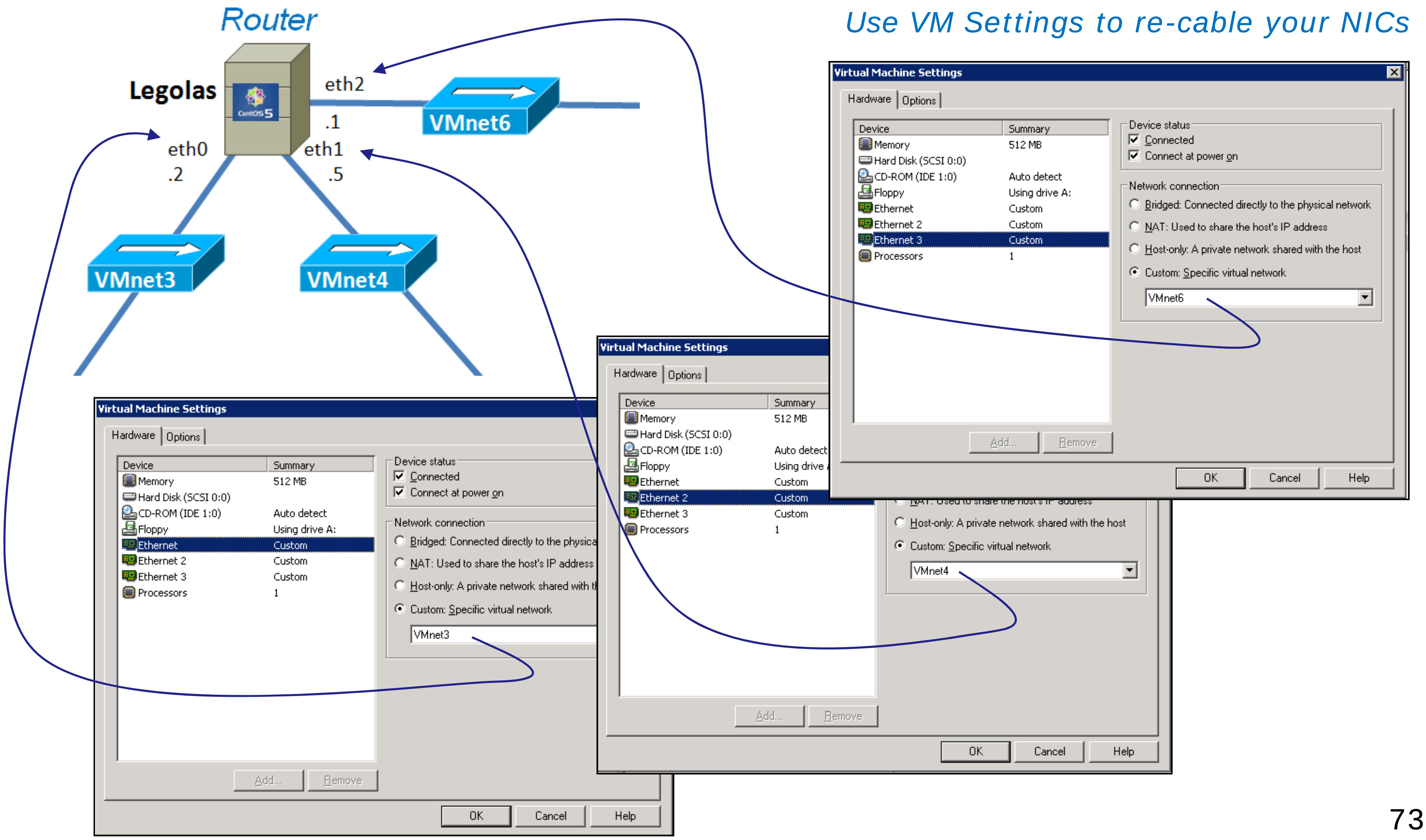
(A must for Lab 4)

- Cabling in the **real world** involves connecting the NICs with an Ethernet LAN cable to various hubs or switches.

- Cabling in the VMware **virtual world involves** configuring the Ethernet Adapters to various VMnets.

Cabling NICs

(A must for Lab 4)



Use VM Settings to re-cable your NICs

Virtual Machine Settings

Hardware | Options

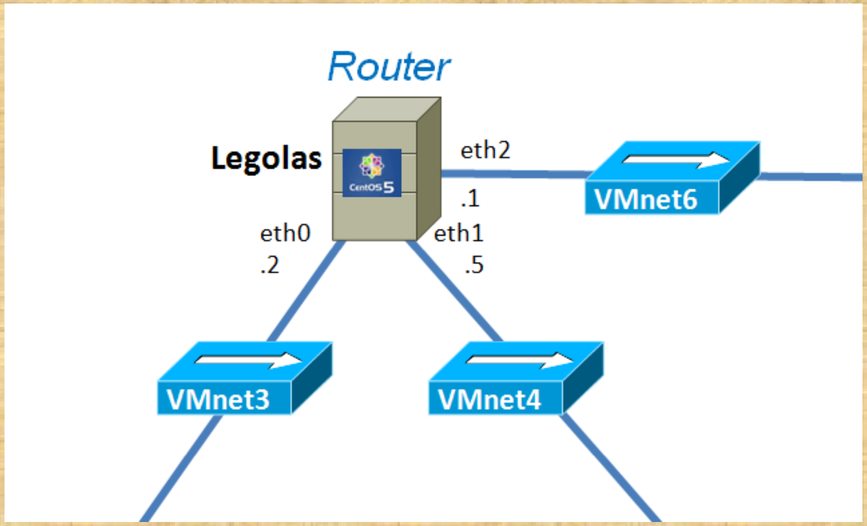
Device	Summary
Memory	512 MB
Hard Disk (SCSI 0:0)	
CD-ROM (IDE 1:0)	Auto detect
Floppy	Using drive A:
Ethernet	Custom
Ethernet 2	Custom
Ethernet 3	Custom
Processors	1

Device status
☑ Connected
☑ Connect at power on

Network connection
○ Bridged: Connected directly to the physical network
○ NAT: Used to share the host's IP address
○ Host-only: A private network shared with the host
● Custom: Specific virtual network

Ethernet: VMnet3
Ethernet 2: VMnet4
Ethernet 3: VMnet6

Exercise



1. Power on Legolas

2. Note: we can re-cable with the VMs running just like we can with real computers

3. Cable eth0 to VMnet3

4. Cable eth1 to VMnet4

5. Cable eth2 to VMnet6

Run Levels
(Centos)

- The CentOS VMs: Elrond, Celebrian, Legolas and Arwen
- Configured to startup in run level 3 (virtual tty terminal console)
- Use **runlevel** command to display previous and current run levels

```
[root@legolas ~]# runlevel
3 5
[root@legolas ~]#
```

Run Levels
(Centos)

Initial run level is configured in /etc/inittab

```
[root@legolas ~]# cat /etc/inittab
#
# inittab       This file describes how the INIT process should set up
#               the system in a certain run-level.
#
# Author:       Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#               Modified for RHS Linux by Marc Ewing and Donnie Barnes
#

# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# When our UPS tells us power has failed, assume we have a few minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

# If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"


# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Run xdm in runlevel 5
x:5:respawn:/etc/X11/prefdm -nodaemon
[root@legolas ~]#
```

```
# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:
```

Run Levels
(Centos)

To get to graphical Gnome desktop:

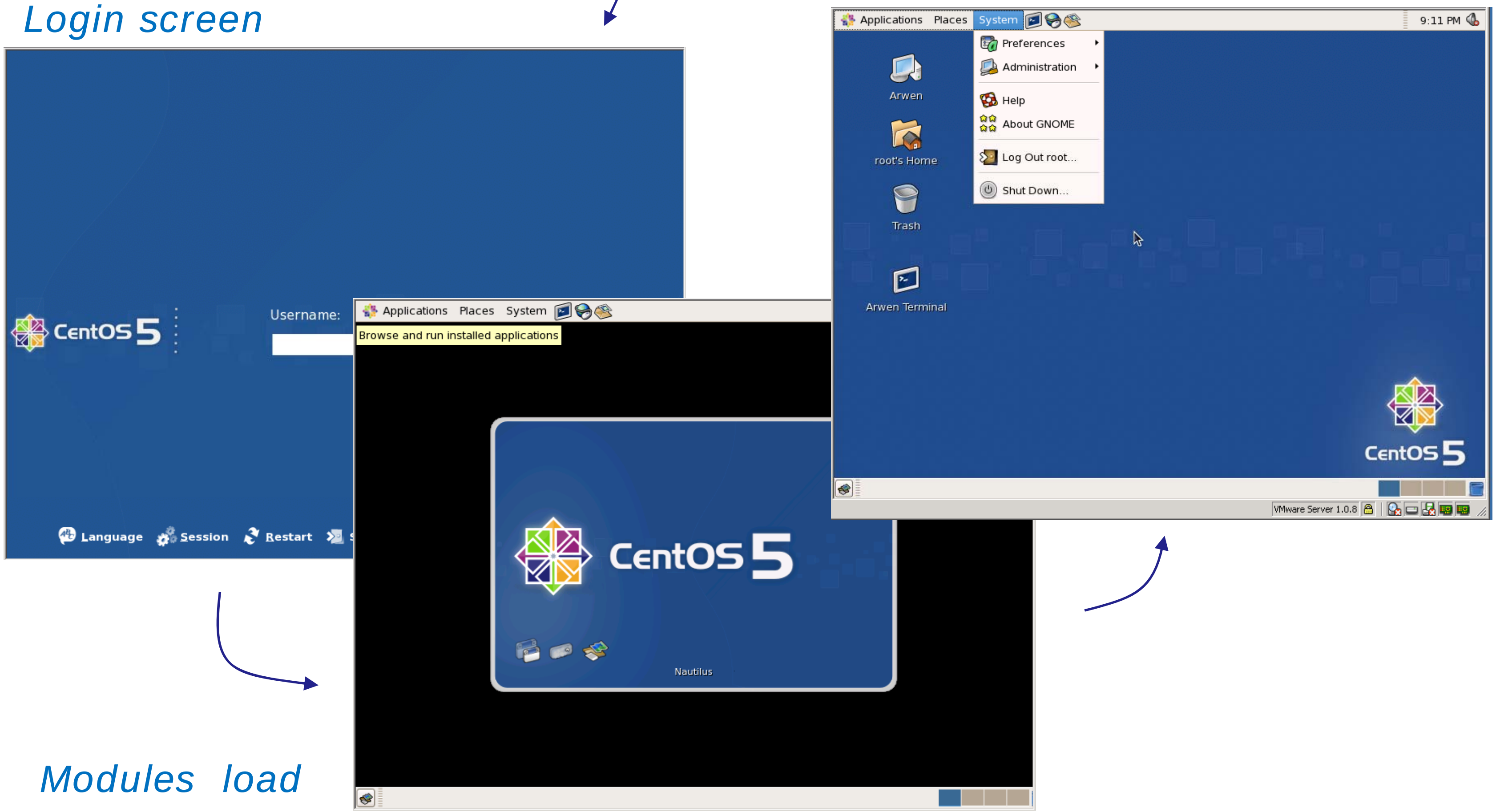
1. Using **startx**
   - Log in as root on the virtual tty terminal
   - Type **startx** (no need to login again)
   - Use **ctrl-alt-f**n (n=1-7) to toggle virtual terminals and desktop
   - To exit, **System menu > Logout**

2. Using **init 5**
   - Log in as root on the virtual tty terminal
   - Type **init 5**
   - Login on login screen
   - Use **ctrl-alt-f**n (n=1-7) to toggle virtual terminals and desktop
   - To exit, **System menu > Logout or Shutdown**

Run Levels – Getting desktop via init 5

(Centos)

```
[root@legolas ~]# init 5
```



Login screen

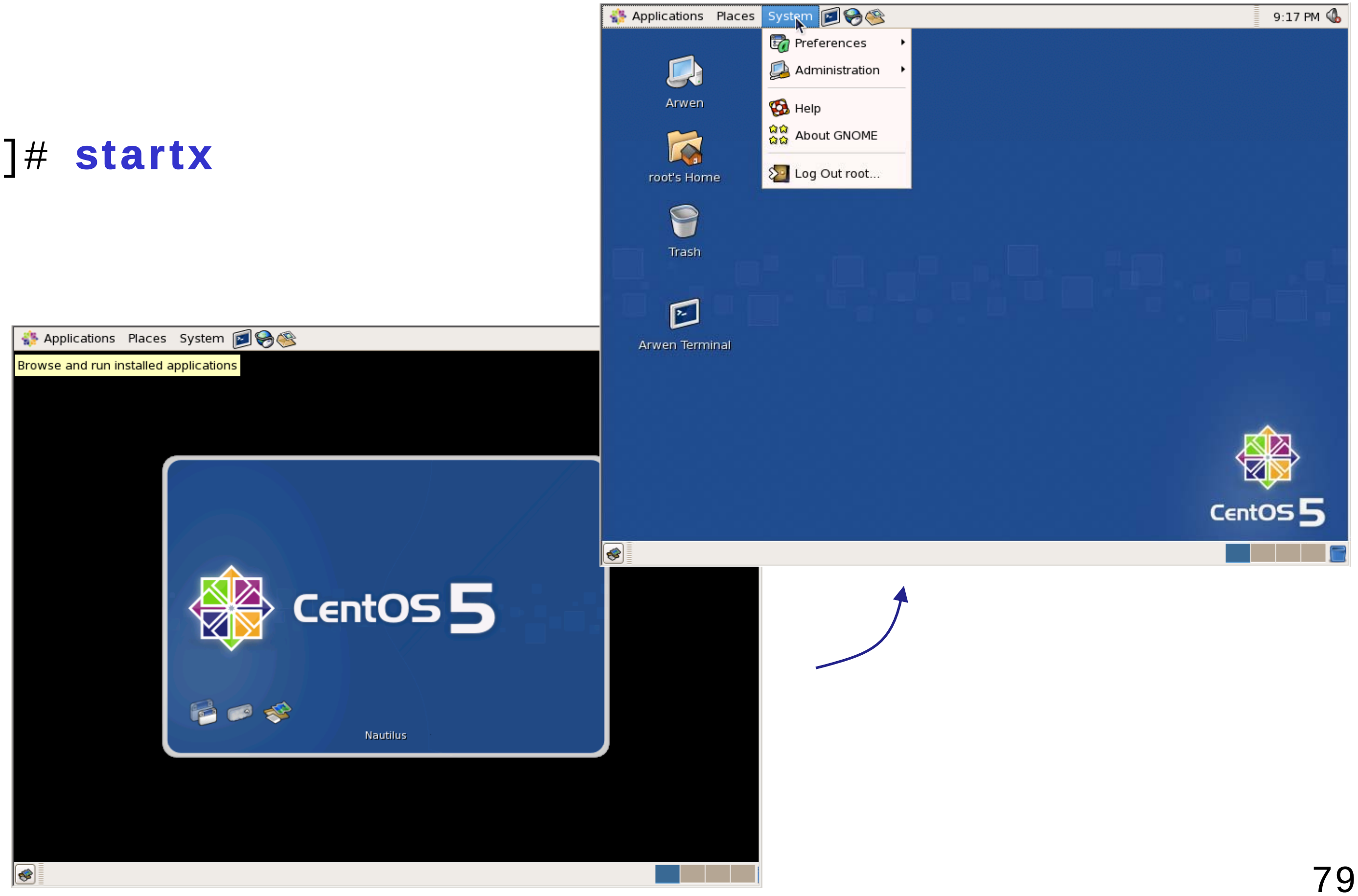
Modules load

Both Logoff and Shutdown options

Run Levels – Getting desktop via startx

(Centos)

```
[root@legolas ~]# startx
```



Just Logoff option

Modules load

Exercise

1. Power on Legolas
2. Login as root on virtual tty console
3. Use **runlevel** to display run level
4. Use **startx** to get Gnome desktop
5. Use **Ctrl-Alt-F**n (n=1-7) keys to toggle terminals and desktop
6. Logout of Gnome desktop (back to virtual tty console)
7. Use **init 5** to get to run level 5
8. Login to Desktop session
9. Use **Ctrl-Alt-F**n (n=1-7) keys to toggle terminals and desktop
10. Logout of Gnome desktop (back to login screen)
11. **Ctrl-Alt-F2**
12. Use **runlevel** to display run level
13. Use **init 3** to return to run level 3

Modifying the Firewall
(Centos)

- RIP needs UDP port 520 open to work properly
- We want our routers to forward, not block DNS name resolution queries and responses (UDP port 53).
- For the Telent Server, we need the Telnet port open (TCP port 23)

Modifying the Firewall
(Centos)

Default firewall

```
[root@celebrian ~]# iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere
2    ACCEPT     icmp --  anywhere             anywhere            icmp any
3    ACCEPT     esp  --  anywhere             anywhere
4    ACCEPT     ah   --  anywhere             anywhere
5    ACCEPT     udp  --  anywhere             224.0.0.251         udp dpt:mdns
6    ACCEPT     udp  --  anywhere             anywhere            udp dpt:ipp
7    ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ipp
8    ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
9    ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
10   REJECT     all  --  anywhere             anywhere            reject-with icmp-host-
   prohibited
[root@celebrian ~]#
```

Note that forwarded packets get sent through the INPUT filter (blocks DNS requests that should be forwarded)

... and no openings for RIP or Telnet

Modifying the Firewall
(Centos)

Default firewall

```
[root@celebrian ~]# 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@celebrian ~]#
```

Note that forwarded packets get sent through the INPUT filter (blocks DNS requests that should be forwarded)

... and no openings for RIP or Telnet

Modifying the Firewall
(Centos)

Modified firewall

```
[root@arwen ~]# iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain RH-Firewall-1-INPUT (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere
2    ACCEPT     icmp --  anywhere             anywhere            icmp any
3    ACCEPT     esp  --  anywhere             anywhere
4    ACCEPT     ah   --  anywhere             anywhere
5    ACCEPT     udp  --  anywhere             224.0.0.251         udp dpt:mdns
6    ACCEPT     udp  --  anywhere             anywhere            udp dpt:ipp
7    ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ipp
8    ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
9    ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
10   ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:telnet
11   ACCEPT     udp  --  anywhere             anywhere            state NEW udp dpt:router
12   REJECT     all  --  anywhere             anywhere            reject-with icmp-host-
   prohibited
[root@arwen ~]#
```

No filtering now on forwarded packets

RIP and Telnet ports open

Modifying the Firewall
(Centos)

Modified firewall

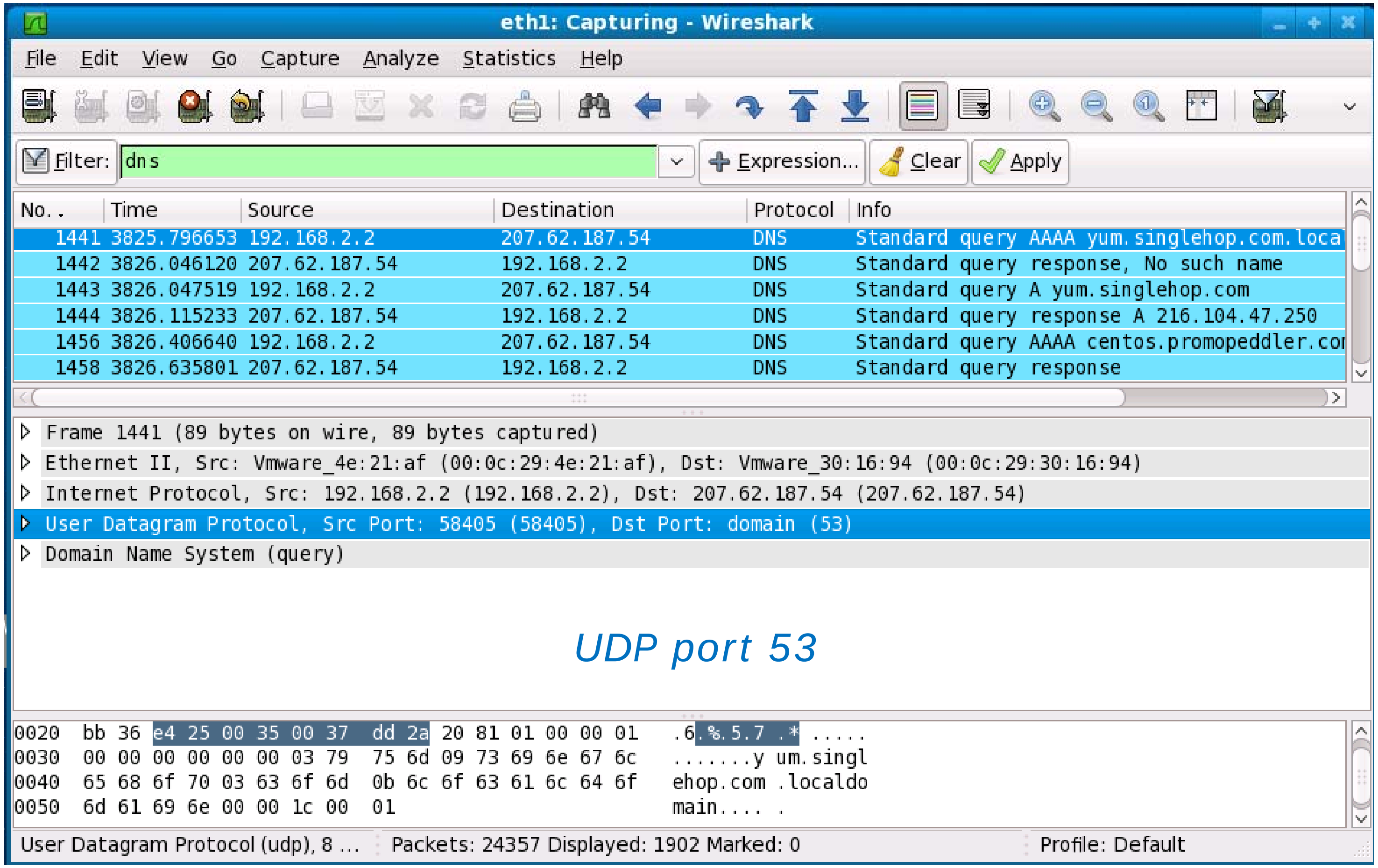
```
[root@arwen ~]# cat /etc/sysconfig/iptables
# Generated by iptables-save v1.3.5 on Thu Feb 26 08:22:29 2009
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [946:71747]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp -m icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p esp -j ACCEPT
-A RH-Firewall-1-INPUT -p ah -j ACCEPT
-A RH-Firewall-1-INPUT -d 224.0.0.251 -p udp -m udp --dport 5353 -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 RELATED,ESTABLISHED -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 23 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m state --state NEW -m udp --dport 520 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Thu Feb 26 08:22:29 2009
[root@arwen ~]#
```

No filtering now on any forwarded packets

RIP (UDP port 520) and Telnet (TCP port 23) ports open

Modifying the Firewall
(Centos)

We would like DNS queries to be passed through the routers


eth1: Capturing - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: dns
Expression... Clear Apply

```
No. .  Time        Source         Destination    Protocol  Info
  1441 3825.796653 192.168.2.2    207.62.187.54  DNS       Standard query AAAA yum.singlehop.com.loca
  1442 3826.046120 207.62.187.54  192.168.2.2    DNS       Standard query response, No such name
  1443 3826.047519 192.168.2.2    207.62.187.54  DNS       Standard query A yum.singlehop.com
  1444 3826.115233 207.62.187.54  192.168.2.2    DNS       Standard query response A 216.104.47.250
  1456 3826.406640 192.168.2.2    207.62.187.54  DNS       Standard query AAAA centos.promopeddler.cor
  1458 3826.635801 207.62.187.54  192.168.2.2    DNS       Standard query response
```

```
▷ Frame 1441 (89 bytes on wire, 89 bytes captured)
▷ Ethernet II, Src: Vmware_4e:21:af (00:0c:29:4e:21:af), Dst: Vmware_30:16:94 (00:0c:29:30:16:94)
▷ Internet Protocol, Src: 192.168.2.2 (192.168.2.2), Dst: 207.62.187.54 (207.62.187.54)
▷ User Datagram Protocol, Src Port: 58405 (58405), Dst Port: domain (53)
▷ Domain Name System (query)
```

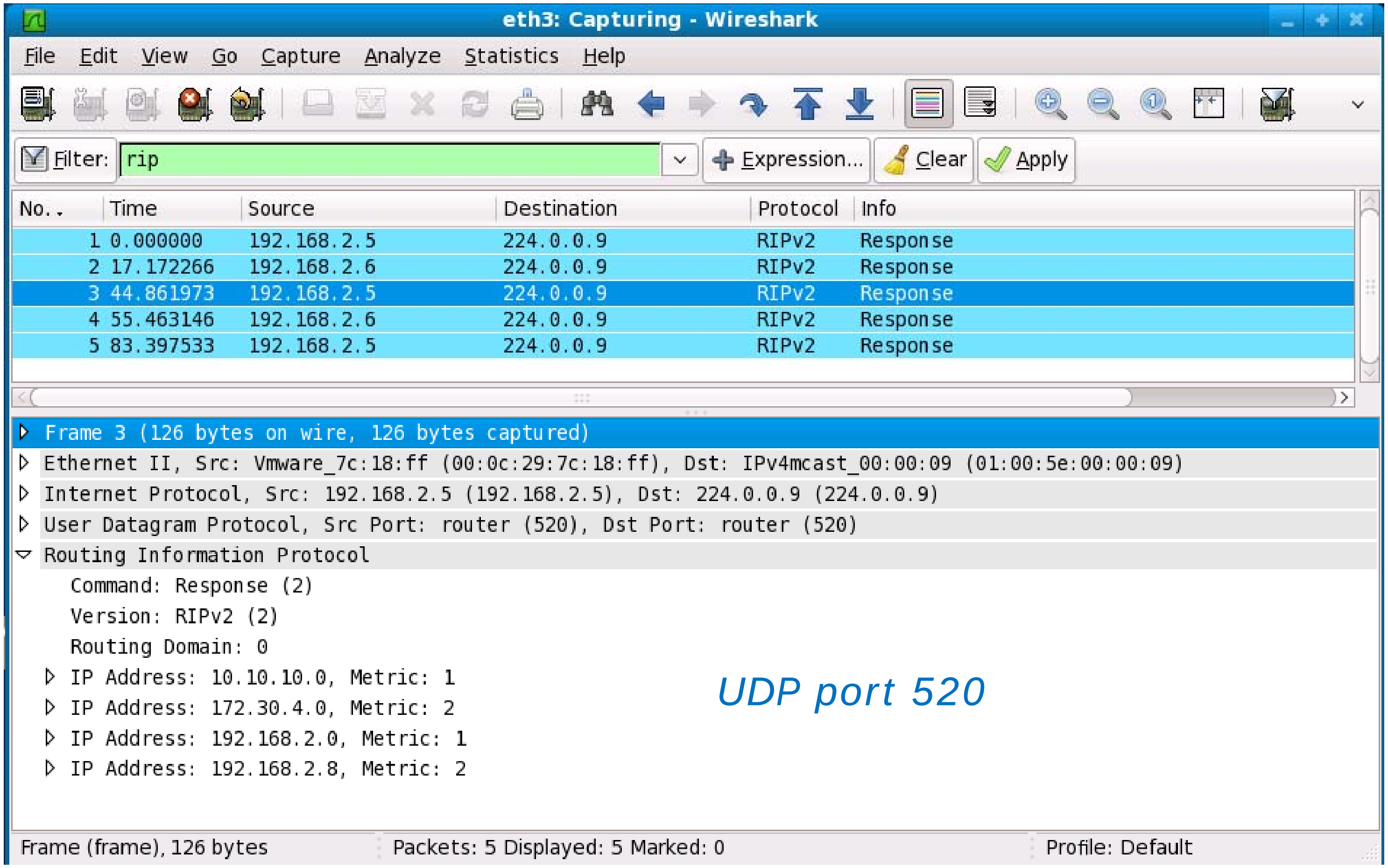
UDP port 53

```
0020  bb 36 e4 25 00 35 00 37  dd 2a 20 81 01 00 00 01   .6.%.5.7 .* .....
0030  00 00 00 00 00 00 03 79  75 6d 09 73 69 6e 67 6c   .......y um.singl
0040  65 68 6f 70 03 63 6f 6d  0b 6c 6f 63 61 6c 64 6f   ehop.com .localdo
0050  6d 61 69 6e 00 00 1c 00  01                        main.... .
```

User Datagram Protocol (udp), 8 ... Packets: 24357 Displayed: 1902 Marked: 0 Profile: Default

Modifying the Firewall
(Centos)

We would like RIP updates to be passed between the routers


eth3: Capturing - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: rip
Expression... Clear Apply

```
No. .  Time          Source         Destination   Protocol  Info
     1 0.000000      192.168.2.5    224.0.0.9     RIPv2     Response
     2 17.172266     192.168.2.6    224.0.0.9     RIPv2     Response
     3 44.861973     192.168.2.5    224.0.0.9     RIPv2     Response
     4 55.463146     192.168.2.6    224.0.0.9     RIPv2     Response
     5 83.397533     192.168.2.5    224.0.0.9     RIPv2     Response
```

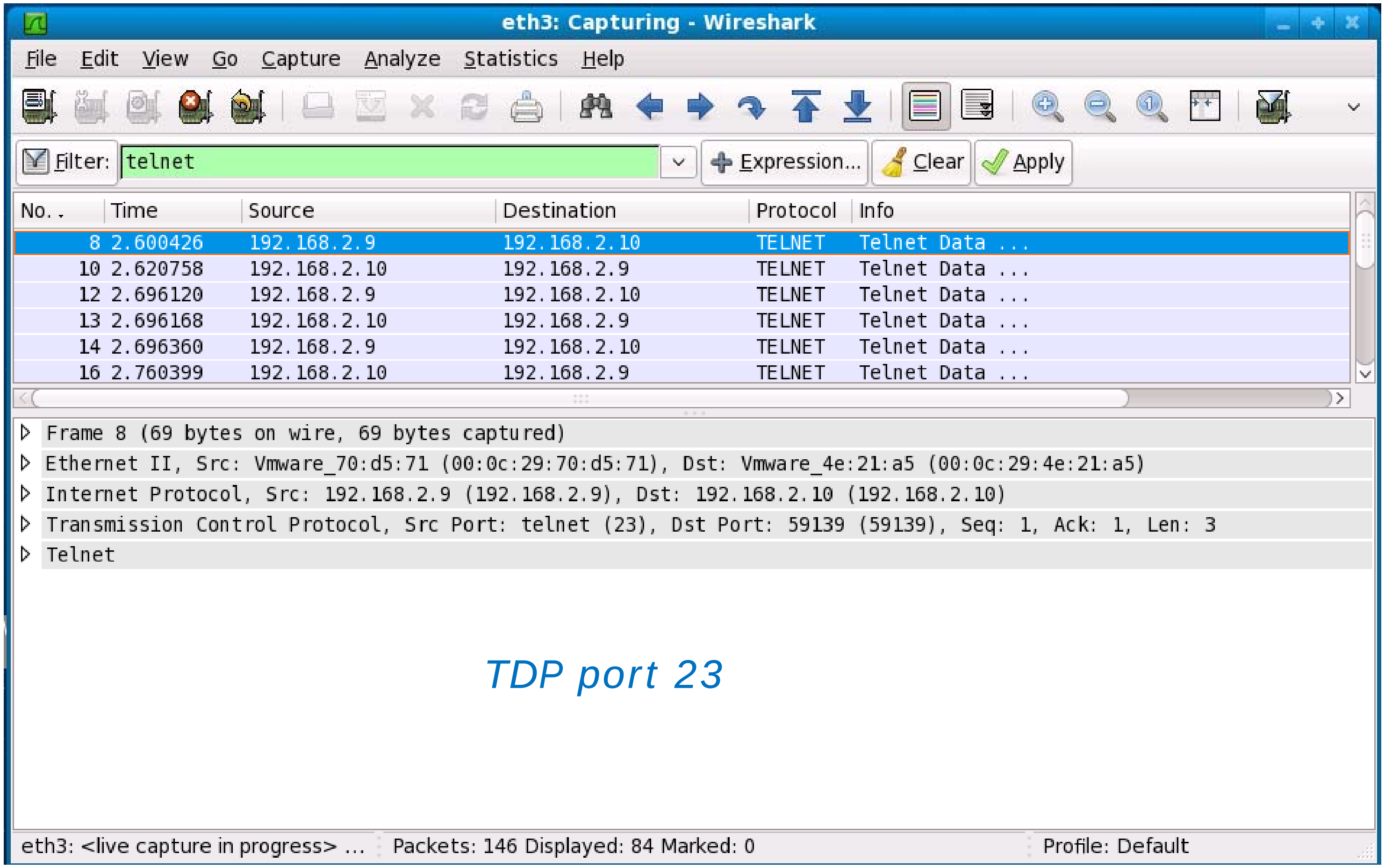
```
▷ Frame 3 (126 bytes on wire, 126 bytes captured)
▷ Ethernet II, Src: Vmware_7c:18:ff (00:0c:29:7c:18:ff), Dst: IPv4mcast_00:00:09 (01:00:5e:00:00:09)
▷ Internet Protocol, Src: 192.168.2.5 (192.168.2.5), Dst: 224.0.0.9 (224.0.0.9)
▷ User Datagram Protocol, Src Port: router (520), Dst Port: router (520)
▽ Routing Information Protocol
    Command: Response (2)
    Version: RIPv2 (2)
    Routing Domain: 0
  ▷ IP Address: 10.10.10.0, Metric: 1
  ▷ IP Address: 172.30.4.0, Metric: 2
  ▷ IP Address: 192.168.2.0, Metric: 1
  ▷ IP Address: 192.168.2.8, Metric: 2
```

Frame (frame), 126 bytes   Packets: 5 Displayed: 5 Marked: 0   Profile: Default

UDP port 520

Modifying the Firewall
(Centos)

We would like Arwen to accept Telnet sessions



Filter: telnet

No. .	Time	Source	Destination	Protocol	Info
8	2.600426	192.168.2.9	192.168.2.10	TELNET	Telnet Data ...
10	2.620758	192.168.2.10	192.168.2.9	TELNET	Telnet Data ...
12	2.696120	192.168.2.9	192.168.2.10	TELNET	Telnet Data ...
13	2.696168	192.168.2.10	192.168.2.9	TELNET	Telnet Data ...
14	2.696360	192.168.2.9	192.168.2.10	TELNET	Telnet Data ...
16	2.760399	192.168.2.10	192.168.2.9	TELNET	Telnet Data ...

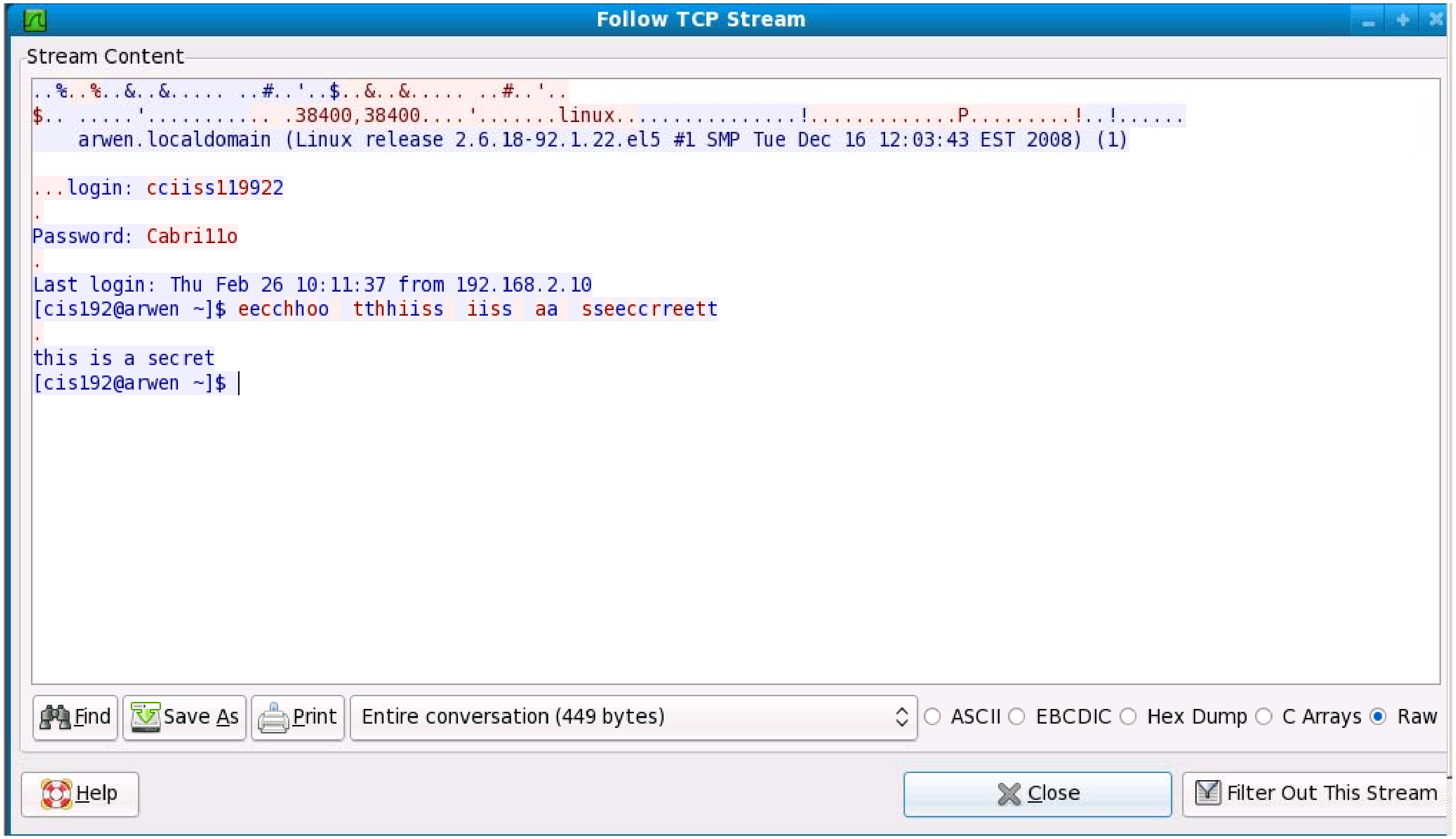
```
▷ Frame 8 (69 bytes on wire, 69 bytes captured)
▷ Ethernet II, Src: Vmware_70:d5:71 (00:0c:29:70:d5:71), Dst: Vmware_4e:21:a5 (00:0c:29:4e:21:a5)
▷ Internet Protocol, Src: 192.168.2.9 (192.168.2.9), Dst: 192.168.2.10 (192.168.2.10)
▷ Transmission Control Protocol, Src Port: telnet (23), Dst Port: 59139 (59139), Seq: 1, Ack: 1, Len: 3
▷ Telnet
```

TDP port 23

eth3: <live capture in progress> ... Packets: 146 Displayed: 84 Marked: 0 Profile: Default

Modifying the Firewall
(Centos)

BTW ... this is why we use SSH!
We are using a Telnet server in Lab 4 so we don't forget why!

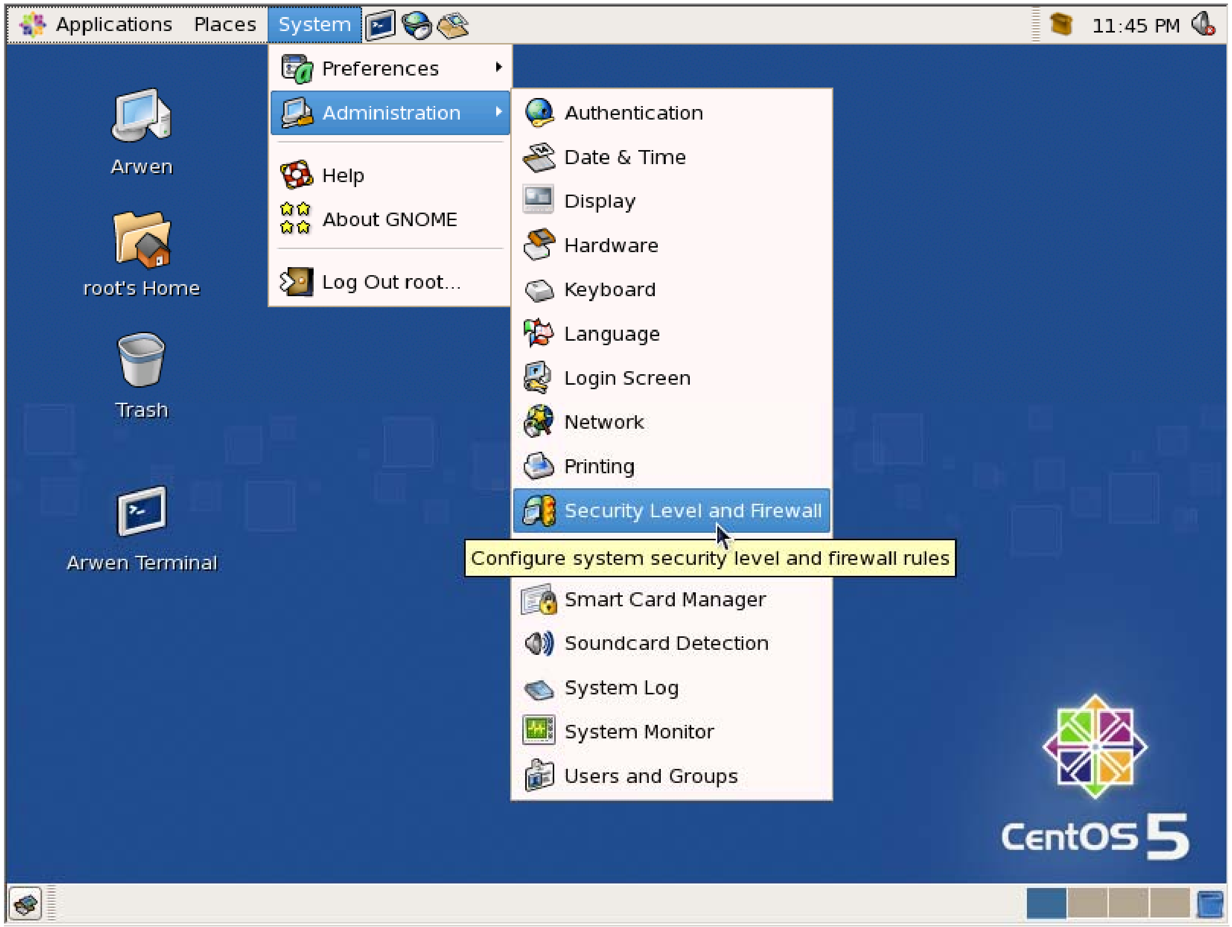


```
..%..%..&..&..... ..#..'..$..&..&..... ..#..'..
$.. .....'.......... .38400,38400....'......linux...............!............P.........!..!......
    arwen.localdomain (Linux release 2.6.18-92.1.22.el5 #1 SMP Tue Dec 16 12:03:43 EST 2008) (1)

...login: cciiss119922
.
Password: Cabrillo
.
Last login: Thu Feb 26 10:11:37 from 192.168.2.10
[cis192@arwen ~]$ eecchhoo  tthhiiss  iiss  aa  sseeccrreett
.
this is a secret
[cis192@arwen ~]$ |
```

Entire conversation (449 bytes) ○ ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays ● Raw

Modifying the Firewall
(Centos)

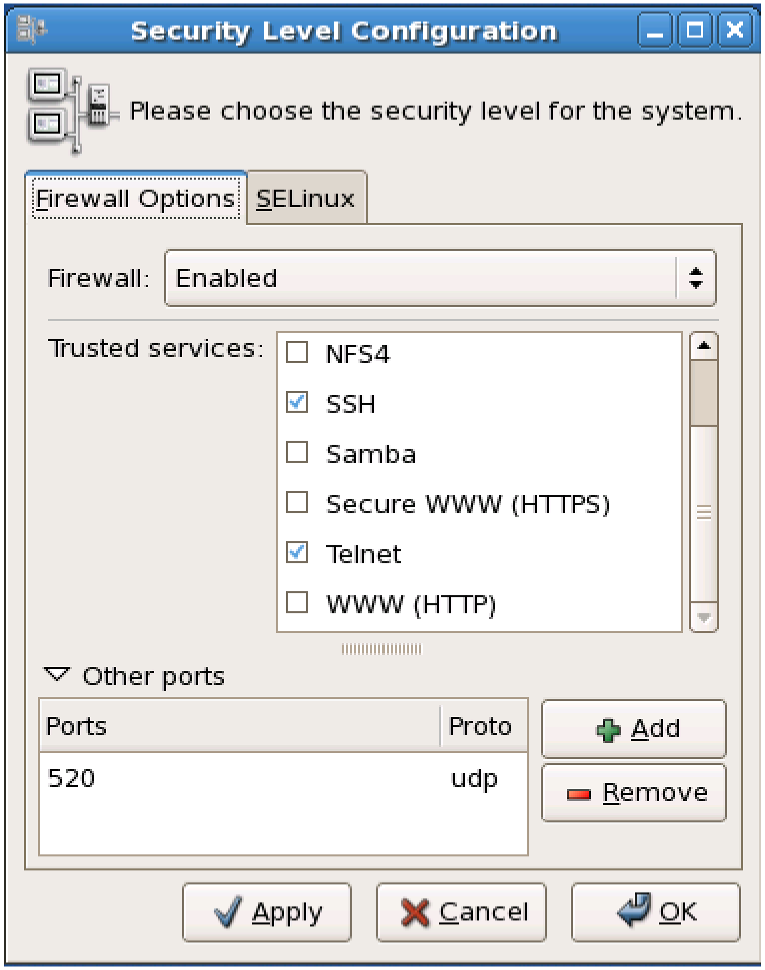
The Red Hat family has a Security Level and Firewall utility



Configure system security level and firewall rules

Modifying the Firewall
(Centos)

Security Level Configuration Utility



Please choose the security level for the system.

Firewall: Enabled

Trusted services:
- ☐ NFS4
- ☑ SSH
- ☐ Samba
- ☐ Secure WWW (HTTPS)
- ☑ Telnet
- ☐ WWW (HTTP)

▽ Other ports

Ports	Proto
520	udp

Trusted = firewall will accept new connections from the outside to this application (port)

SSH port is open already on CentOS VMs

Telnet port is needs to be opened on just Arwen for Lab 4

UDP 520 needs to be open for RIP

Modifying the Firewall
(Centos)

To stop filtering forwarded packets do the following:

```
[root@legolas ~]#  iptables -D FORWARD 1
[root@legolas ~]#  iptables -P FORWARD ACCEPT
[root@legolas ~]#  iptables-save > /etc/sysconfig/iptables
[root@legolas ~]# service iptables restart
Flushing firewall rules:                                  [  OK  ]
Setting chains to policy ACCEPT: filter                   [  OK  ]
Unloading iptables modules:                               [  OK  ]
Applying iptables firewall rules:                         [  OK  ]
Loading additional iptables modules: ip_conntrack_netbios_n[  OK  ]
[root@legolas ~]#
```

More on iptables in future lessons. What we did here was delete rule 1 on the FORWARD filter, make sure the FORWARD policy was set to ACCEPT all packets. The settings were saved to the configuration file and finally iptables restarted to use the new settings

Exercise

1. Revert Arwen to snapshot
2. Modify the firewall on Arwen to:
   - Open UDP port 520 for RIP
   - Open TCP port 23 for Telnet
   - Remove any filtering on forwarded packets

Modifying SELinux
(Centos)

- One way to save configuration files from the Quagga shell is to set the policy from Enforcing to Permissive
- A better way would be to find the settings so SELinux could be left in Enforcing mode!

but we will do that in later labs ....

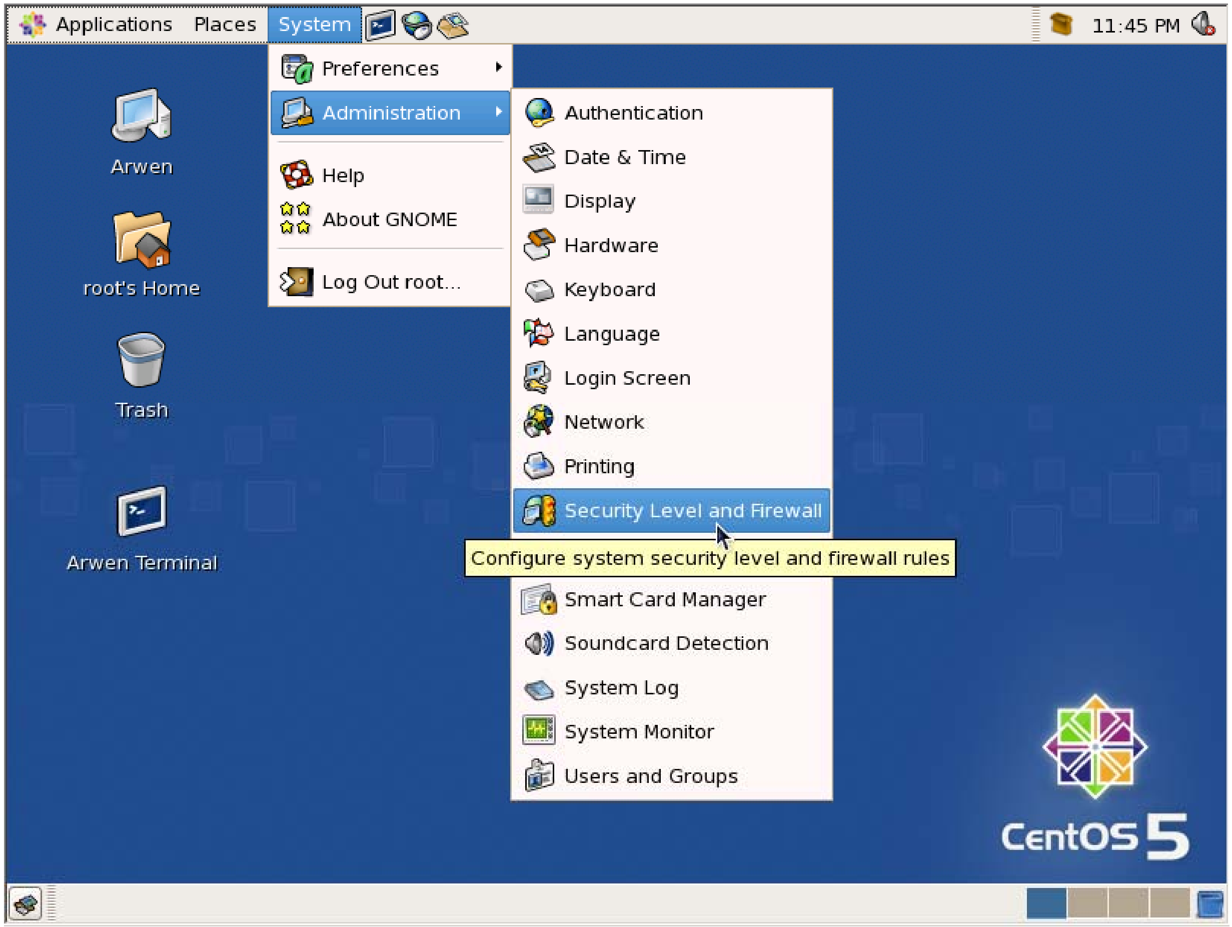
Modifying SELinux
(Centos)

SELinux policy = Enforcing

```
[root@legolas ~]# telnet localhost 2602
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.

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


User Access Verification

Password:
legolas(ripd)> en
legolas(ripd)# wr
Can't open configuration file /etc/quagga/ripd.conf.sWi7Dl.
legolas(ripd)#
```

Modifying SELinux
(Centos)

The Red Hat family has a Security Level and Firewall utility

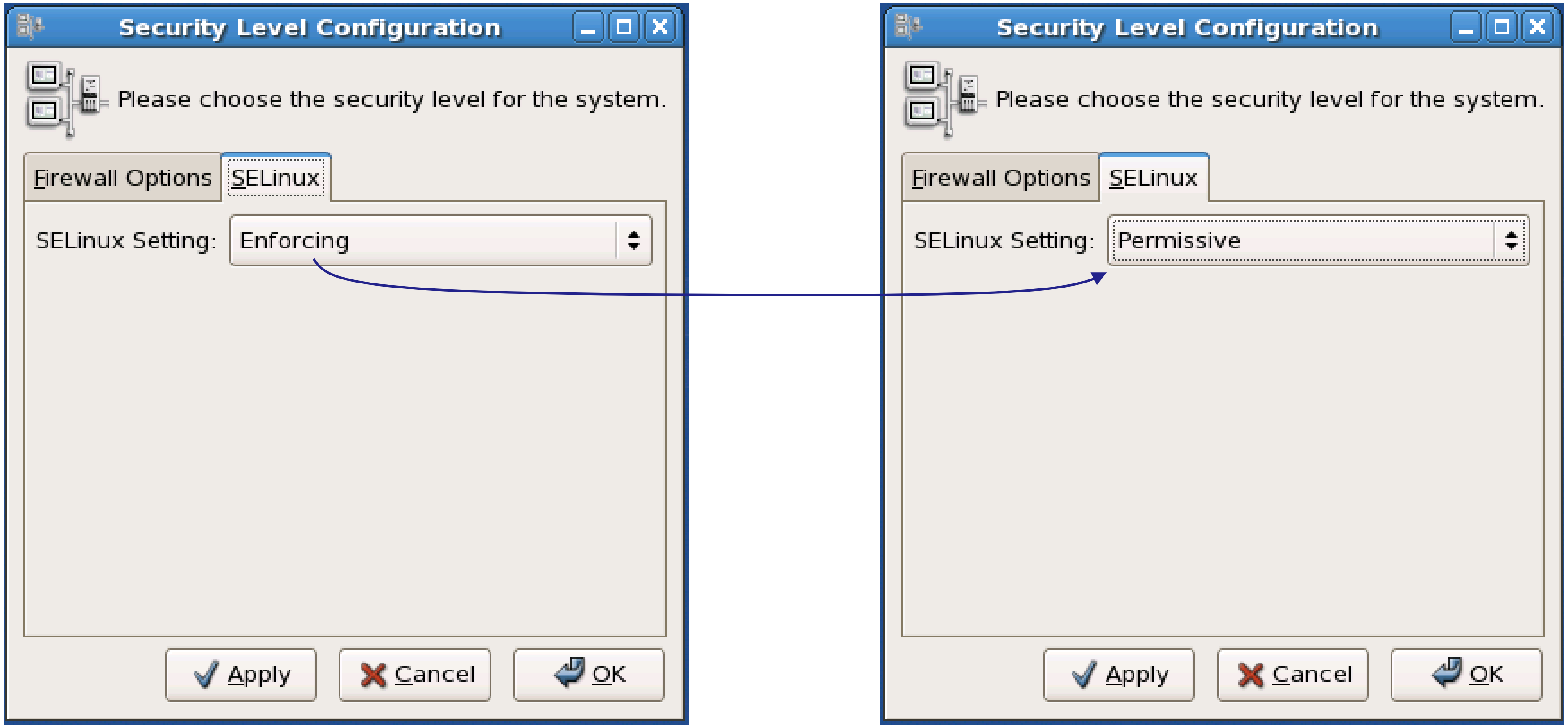


Configure system security level and firewall rules

Modifying SELinux

(Centos)

Changing the SELinux policy from Enforcing to Permissive will allow write to be done from the Quagga shell



Security Level Configuration

Please choose the security level for the system.

SELinux Setting: Enforcing

Security Level Configuration

Please choose the security level for the system.

SELinux Setting: Permissive

Modifying SELinux
(Centos)

SELinux policy = Permissive

```
[root@legolas ~]# telnet localhost 2602
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.

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


User Access Verification

Password:
Password:
legolas(ripd)> en
legolas(ripd)# wr
Configuration saved to /etc/quagga/ripd.conf
legolas(ripd)#
```

Exercise

1. Set the SELinux security level to Permissive

Installing Software on a VM that is not connected to the Internet

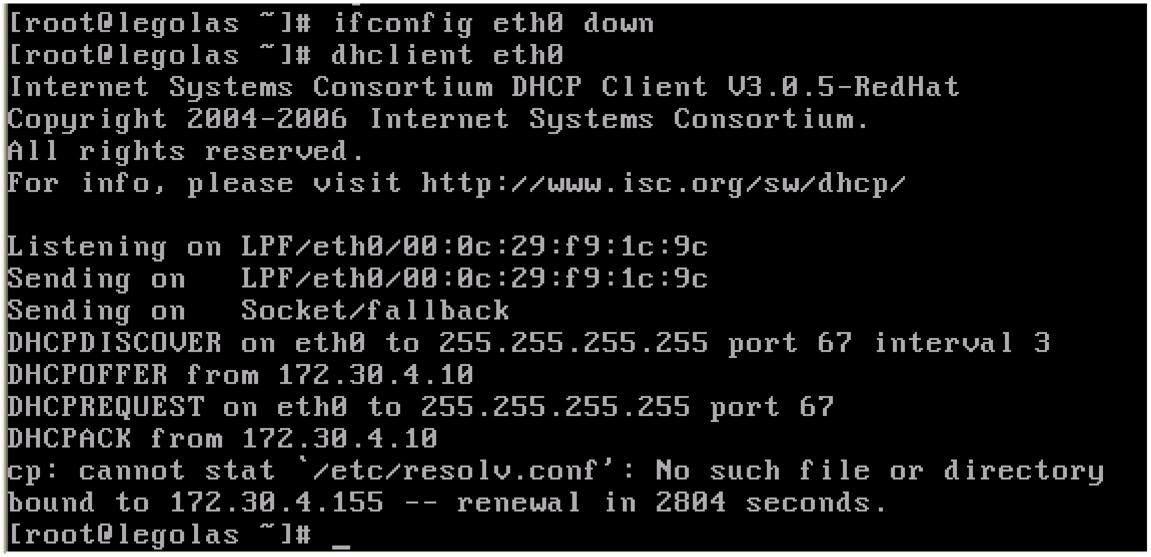
Just cable it temporarily to the Shire network and use dhclient to get an IP address on the Shire 172.30.4.0/24 network

1. Use **ifconfig eth0 down**
2. Re-cable eth0 from VMnet3 to Bridged network.
3. Use **dhclient eth0** to join the Shire network[1].
4. Use **yum install quagga** to install the routing software.
5. Arwen additionally needs the Telnet service so use **yum install telnet-server** after installing quagga.
6. Use **dhclient –r** to release DHCP address.
7. Use **ifconfig eth0 down**
8. Re-cable eth0 from Bridged back to the VMnet3 network.
9. Use **service network restart** to restore static IP settings again.

[1] I've noticed that **dhclient** on the newer CentOS distros will ignore the default gateway from the DHCP server if a different one is specified in /etc/sysconfig/networks. If this happens use **route add default gw 172.30.4.1** to add it manually

Installing Software on a VM that is not connected to the Internet

- *Bringing down the currently configured interface*
- *Re-cable the interface to the Shire network*
- *Using DHCP to get an IP address*



```
[root@legolas ~]# ifconfig eth0 down
[root@legolas ~]# dhclient eth0
Internet Systems Consortium DHCP Client V3.0.5-RedHat
Copyright 2004-2006 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/eth0/00:0c:29:f9:1c:9c
Sending on   LPF/eth0/00:0c:29:f9:1c:9c
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 3
DHCPOFFER from 172.30.4.10
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 172.30.4.10
cp: cannot stat `/etc/resolv.conf': No such file or directory
bound to 172.30.4.155 -- renewal in 2804 seconds.
[root@legolas ~]# _
```

Installing Software on a VM that is not connected to the Internet

Use yum to download and install package

```
[root@legolas ~]# yum install quagga
Loading "fastestmirror" plugin
Determining fastest mirrors
 * base: mirrors.usc.edu
 * updates: centos.mirrors.redwire.net
 * addons: mirror.stanford.edu
 * extras: mirror.dhsrv.com
base                      100% |=========================| 1.1 kB    00:00
updates                   100% |=========================|  951 B    00:00
primary.xml.gz            100% |=========================| 374 kB    00:00
updates   : ################################################## 805/805
addons                    100% |=========================|  951 B    00:00
extras                    100% |=========================| 1.1 kB    00:00
Setting up Install Process
Parsing package install arguments
Resolving Dependencies
--> Running transaction check
---> Package quagga.i386 0:0.98.6-5.el5 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved
```

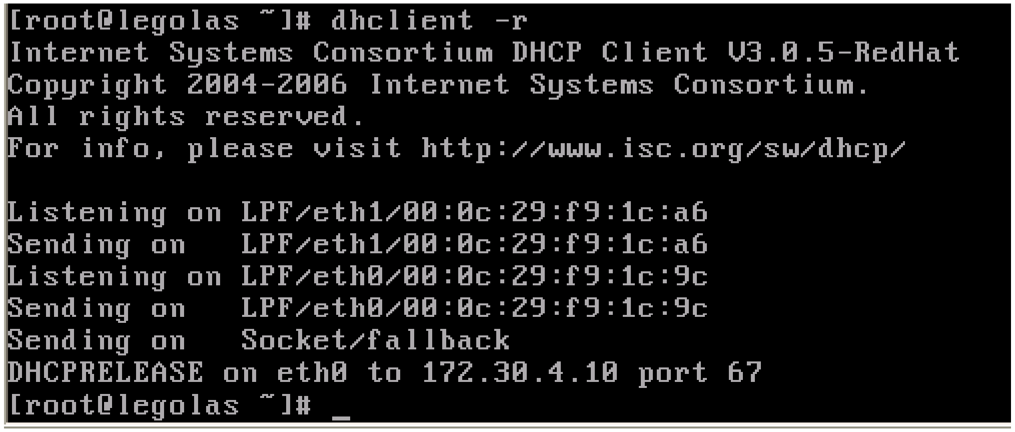
Installing Software on a VM that is not connected to the Internet

yum checks for dependencis, downloads and installs

```
=============================================================================
 Package                 Arch       Version          Repository        Size
=============================================================================
Installing:
 quagga                  i386       0.98.6-5.el5     base              1.1 M

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

Total download size: 1.1 M
Is this ok [y/N]: y
Downloading Packages:
(1/1): quagga-0.98.6-5.el 100% |=========================| 1.1 MB    00:00
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: quagga                       ######################### [1/1]

Installed: quagga.i386 0:0.98.6-5.el5
Complete!
[root@legolas ~]#
```

Installing Software on a VM that is not connected to the Internet

- *Release DHCP address with* ***dhclient -r***



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

Listening on LPF/eth1/00:0c:29:f9:1c:a6
Sending on   LPF/eth1/00:0c:29:f9:1c:a6
Listening on LPF/eth0/00:0c:29:f9:1c:9c
Sending on   LPF/eth0/00:0c:29:f9:1c:9c
Sending on   Socket/fallback
DHCPRELEASE on eth0 to 172.30.4.10 port 67
[root@legolas ~]# _
```

- *Re-cable VM back into your lab network*
- *Use* ***service network restart*** *to restore previous "permanent" static settings or redo manually if done using temporary method*

Exercise

1. Install Quagga on Legolas using **yum install quagga**

Managing Quagga Services (CentOS)

Zebra service configuration file

```
[root@legolas quagga]# cat /etc/quagga/zebra.conf
hostname legolas
password <password>
log stdout
log file /var/log/quagga/zebra.log
```

Managing Quagga Services (CentOS)

```
[root@legolas ~]# cat /etc/quagga/ripd.conf
!
! Zebra configuration saved from vty
!   2009/02/25 16:36:10
!
hostname legolas(ripd)
password <password>
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
 version 2
 redistribute connected
 redistribute static
 network eth0
 network eth1
!
!line vty
!
[root@legolas ~]#
```

ripd service configuration file

Managing Quagga Services (CentOS)

Set permissions on configuration files

```
[root@arwen ~]# chown quagga:quaggavt /etc/quagga/*.conf
[root@arwen ~]#
```

Managing Quagga Services
(CentOS)

Start Quagga services (after editing configuration files)

```
[root@legolas quagga]# service zebra start
Starting zebra: Nothing to flush.
                                                           [  OK  ]
[root@legolas quagga]# service ripd start
Starting ripd:                                             [  OK  ]
```

Configure Quagga services to automatically start at system boot

```
[root@legolas quagga]# chkconfig zebra on
[root@legolas quagga]# chkconfig ripd on
[root@legolas quagga]# chkconfig --list  zebra
zebra           0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@legolas quagga]# chkconfig --list  ripd
ripd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Managing Quagga Services (CentOS)

Check it service are running

```
[root@legolas ~]# service zebra status
zebra (pid 11186) is running...

[root@legolas ~]# service ripd status
ripd (pid 14104) is running...
```

```
[root@legolas ~]# ps -ef | grep quagga
quagga    4569     1  0 Feb25 ?        00:00:00 /usr/sbin/zebra -d -A 127.0.0.1 -f /etc/quagga/zebra.conf
quagga   10889     1  0 15:50 ?        00:00:00 /usr/sbin/ripd -d -A 127.0.0.1 -f /etc/quagga/ripd.conf
root     10954 10920  0 16:05 pts/0    00:00:00 grep quagga
```

Exercise

1. Set up zebra.conf and ripd.conf in /etc/quagga

2. Change ownership of the configuration files
   chown quagga:quaggavt /etc/quagga/*.conf

3. Startup zebra and ripd services

4. Configure them to start automatically

5. telnet localhost 2601

6. telnet localhost 2602

Installing and Configuring Telnet

Install the Telnet package on Arwen

```
[root@arwen ~]# yum install telnet-server
```

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       = 192.168.2.10
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
        disable         = no
}
[root@arwen ~]#
```

Installing and Configuring Telnet

Start or restart service

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

Automatically start at system boot

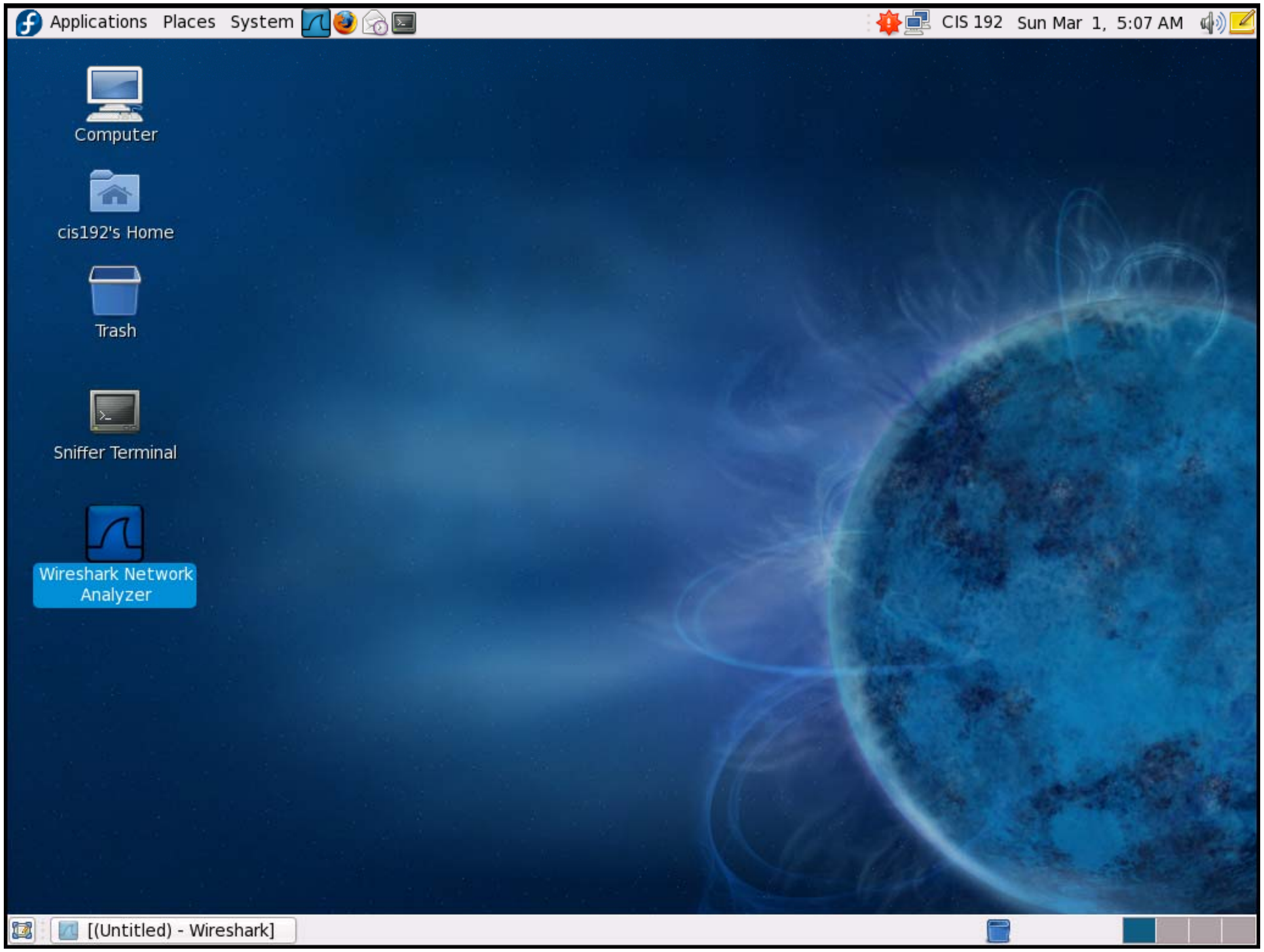
```
[root@arwen ~]# chkconfig xinetd on
[root@arwen ~]# chkconfig --list xinetd
xinetd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@arwen ~]#
```

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:    on
        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

Using Sniffer


Applications Places System
CIS 192 Sun Mar 1, 5:07 AM
Computer
cis192's Home
Trash
Sniffer Terminal
Wireshark Network Analyzer
[(Untitled) - Wireshark]

Fedora 10 VM with Wireshark installed.

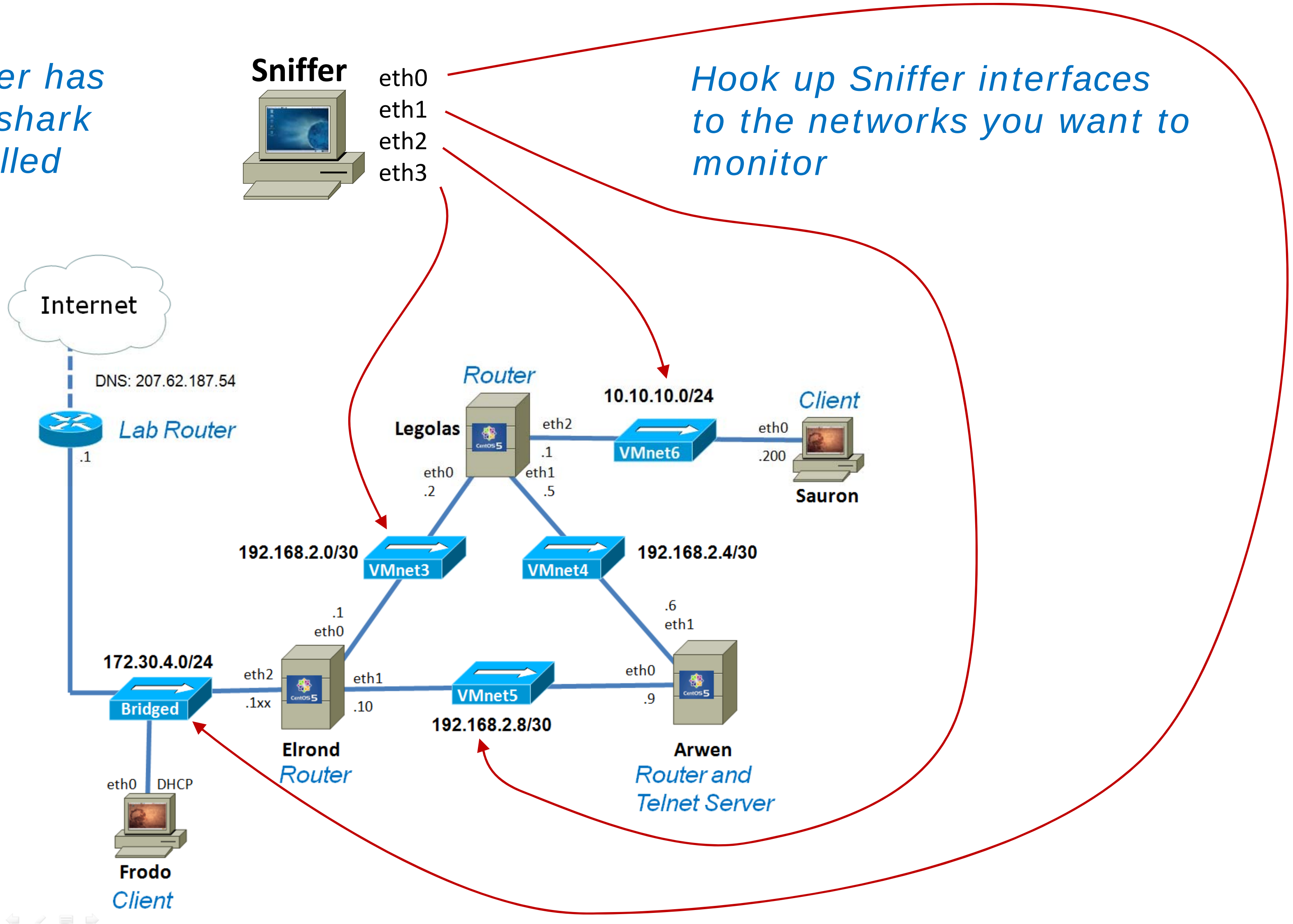
Four interfaces:

Ethernet = eth0
Ethernet 2 = eth1
Ethernet 3 = eth2
Ethernet 4 = eth3

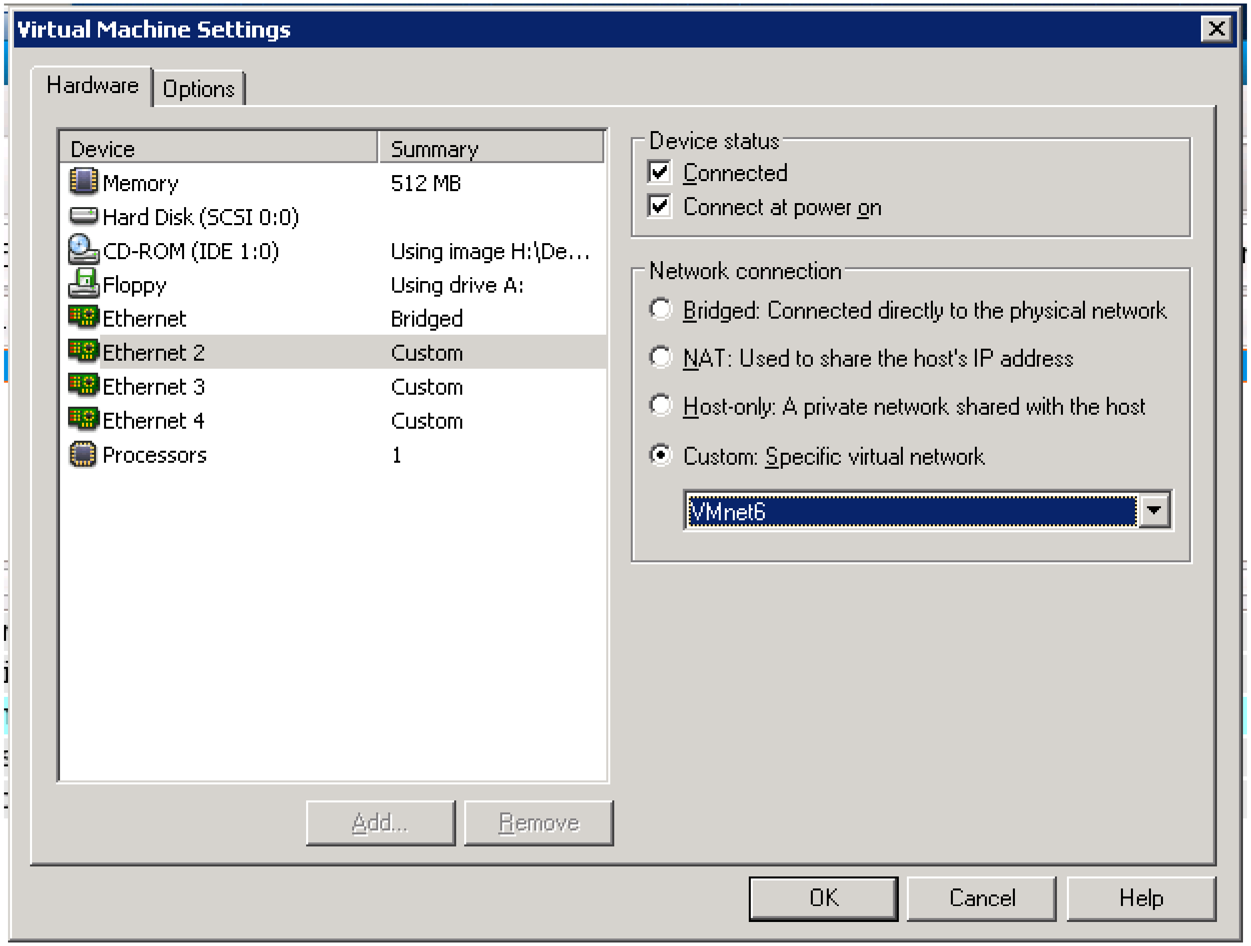
Using Sniffer

Sniffer has Wireshark installed

Hook up Sniffer interfaces to the networks you want to monitor



Using Sniffer



Virtual Machine Settings

Device	Summary
Memory	512 MB
Hard Disk (SCSI 0:0)	
CD-ROM (IDE 1:0)	Using image H:\De...
Floppy	Using drive A:
Ethernet	Bridged
Ethernet 2	Custom
Ethernet 3	Custom
Ethernet 4	Custom
Processors	1

Device status

☑ Connected
☑ Connect at power on

Network connection

○ Bridged: Connected directly to the physical network
○ NAT: Used to share the host's IP address
○ Host-only: A private network shared with the host
● Custom: Specific virtual network

VMnet6

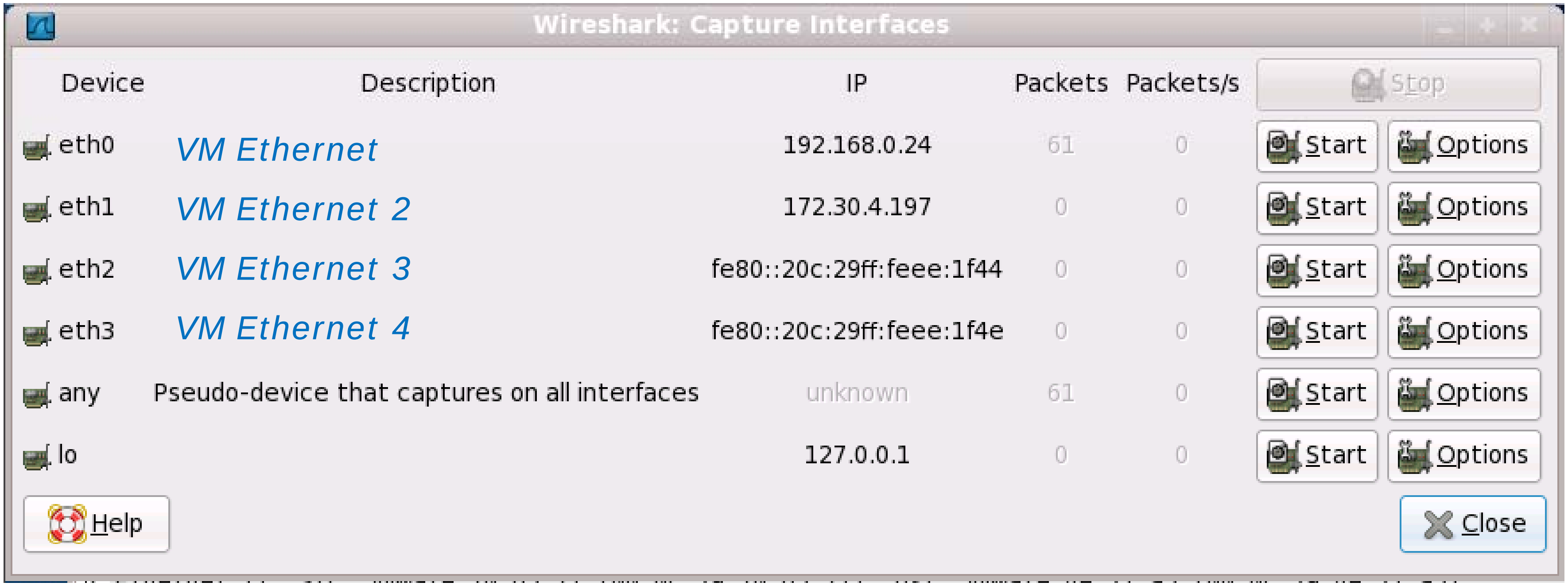
Add...
Remove
OK
Cancel
Help

Sniffer
eth0
eth1
eth2
eth3

Use VM settings to cable Sniffers interfaces to different networks

Using Sniffer

In Wireshark, choose the interface to capture packets on



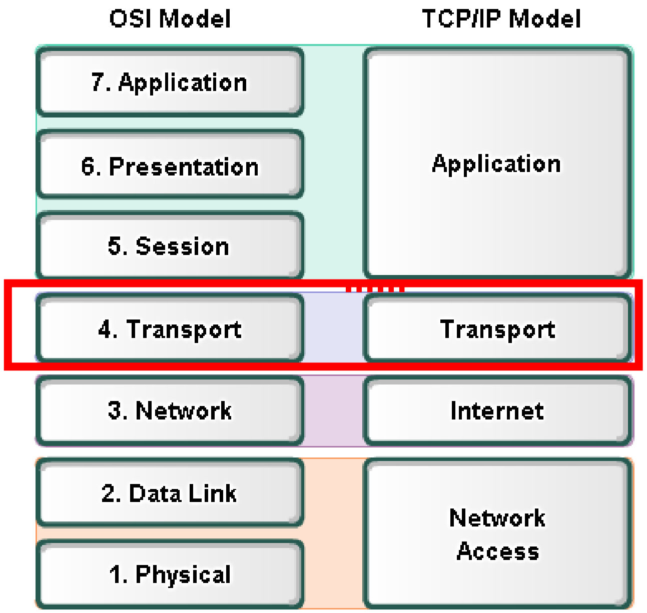
Device	Description	IP	Packets	Packets/s	Stop	
eth0	VM Ethernet	192.168.0.24	61	0	Start	Options
eth1	VM Ethernet 2	172.30.4.197	0	0	Start	Options
eth2	VM Ethernet 3	fe80::20c:29ff:feee:1f44	0	0	Start	Options
eth3	VM Ethernet 4	fe80::20c:29ff:feee:1f4e	0	0	Start	Options
any	Pseudo-device that captures on all interfaces	unknown	61	0	Start	Options
lo		127.0.0.1	0	0	Start	Options

Exercise

1. Power on Sniffer
2. Cable the first Ethernet Adapter to "bridged" (class network)
3. Capture packets using the eth0 interface to see class traffic

Transport Layer Overview

Transport Layer



TCP UDP

- The Layer 4 data stream is a:
  - logical connection between the endpoints of a network,
  - provides transport services from a host to a destination.
- **End-to-end service**.
- The transport layer also provides two protocols
  - **TCP** – Transmission Control Protocol
  - **UDP** – User Datagram Protocol
- PDU: **Segment** *(TCP)*

Lingo: Ethernet frames, IP packets, TCP segments, and UDP datagrams

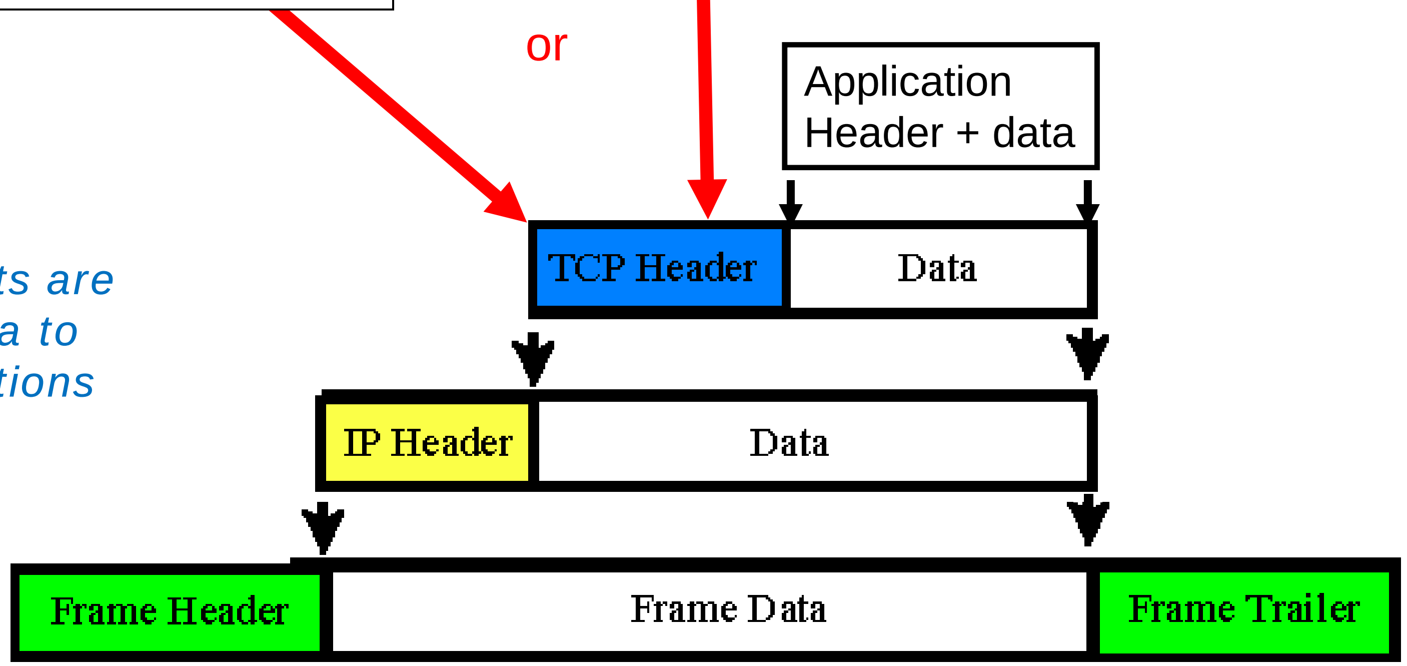
TCP Header

<table>
<tr><td colspan="4">Source Port (16 bits)</td><td colspan="6">Destination Port (16 bits)</td></tr>
<tr><td colspan="10">Sequence Number (32 bits)</td></tr>
<tr><td colspan="10">Acknowledgement Number (32 bits)</td></tr>
<tr><td>Data Offset (4 bits)</td><td>Reserved (6 bits)</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="2">Window (16 bits)</td></tr>
<tr><td colspan="8">Checksum (16 bits)</td><td colspan="2">Urgent Pointer (16 bits)</td></tr>
<tr><td colspan="10">Options and Padding</td></tr>
</table>

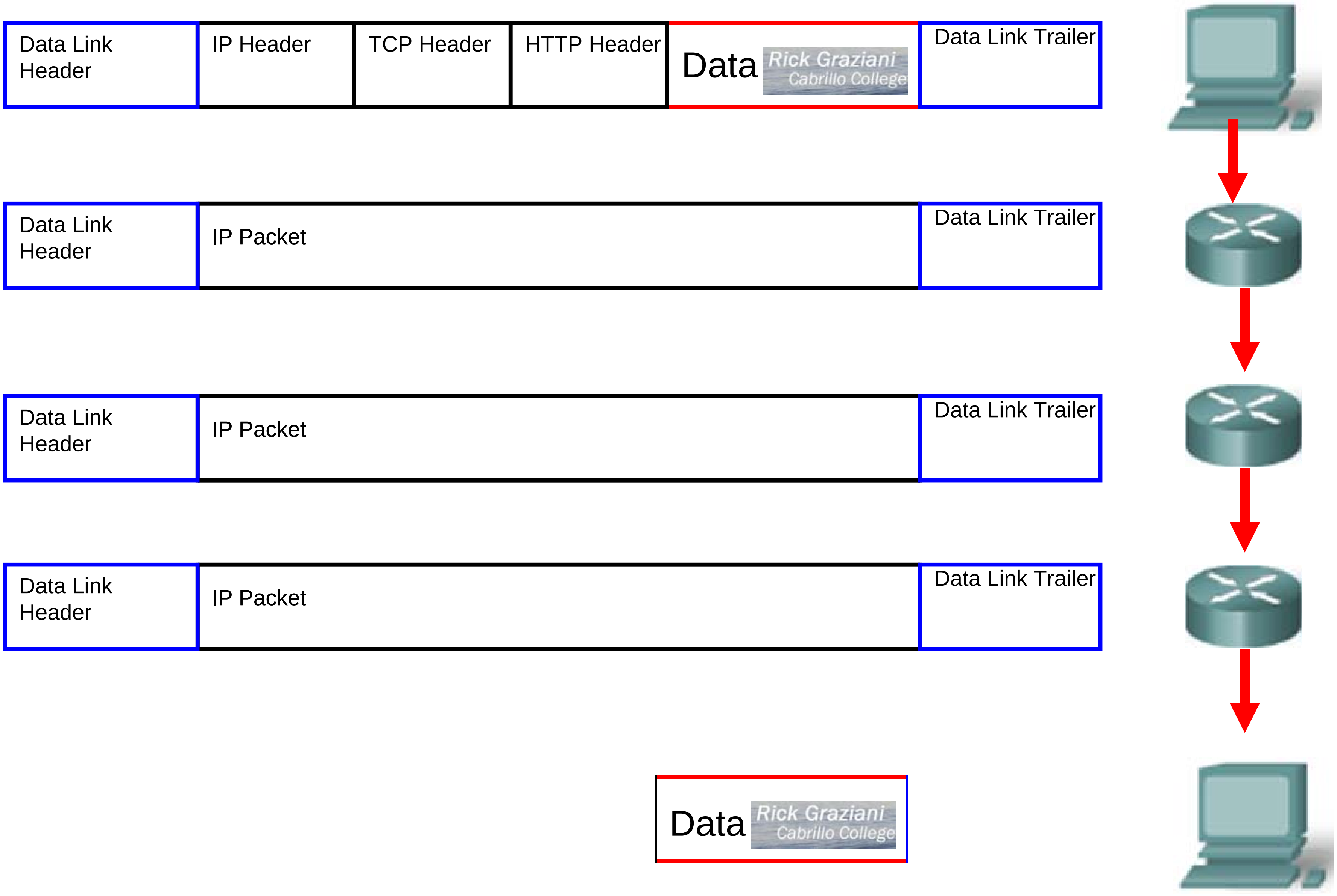
UDP Header

<table>
<tr><td>Source Port (16 bits)</td><td>Destination Port (16 bits)</td></tr>
<tr><td>Length (16 bits)</td><td>Checksum (16 bits)</td></tr>
<tr><td colspan="2">Data....</td></tr>
</table>

The source and destination ports are used to get data to specific applications



Reminder of encapsulation/decapsulation



Data Link Header	IP Header	TCP Header	HTTP Header	Data Rick Graziani Cabrillo College	Data Link Trailer

Data Link Header	IP Packet	Data Link Trailer
Data Link Header	IP Packet	Data Link Trailer
Data Link Header	IP Packet	Data Link Trailer

Data Rick Graziani Cabrillo College

Transport Layer

The Protocols

There are two primary protocols operating at the Transport layer:

User Datagram Protocol (UDP)

Connectionless *(snmp traps are "fire and forget")*

Stateless

Unreliable

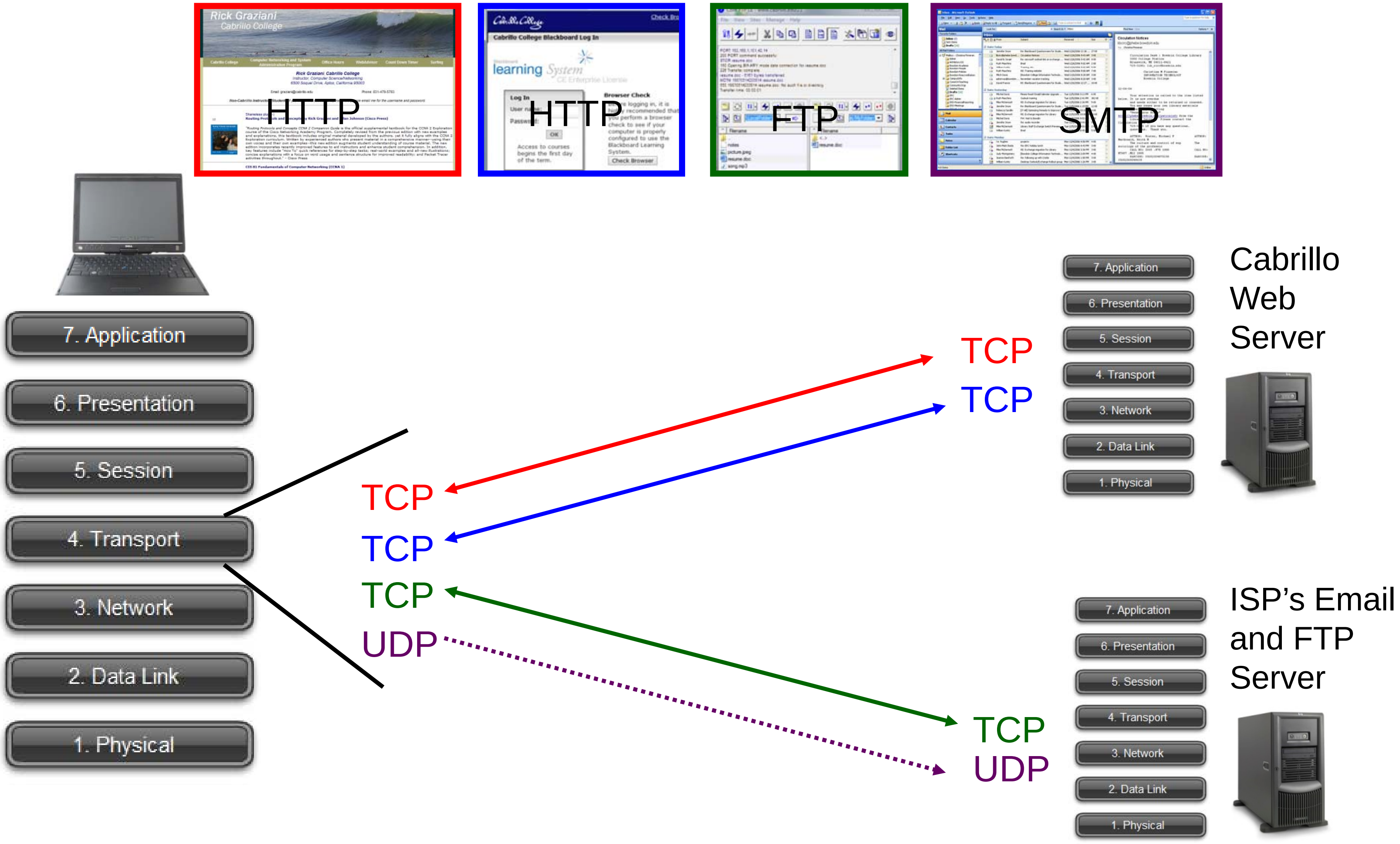
The UDP packet is called a ***packet***

Transmission Control Protocol (TCP)

Connection-oriented

Statefull *(like new or established for firewalls)*

Reliable The TCP packet is called a ***segment***



HTTP

HTTP

FTP

SMTP

7. Application

6. Presentation

5. Session

4. Transport

3. Network

2. Data Link

1. Physical

TCP

TCP

TCP

UDP

TCP

TCP

Cabrillo Web Server

7. Application

6. Presentation

5. Session

4. Transport

3. Network

2. Data Link

1. Physical

TCP

UDP

ISP’s Email and FTP Server

7. Application

6. Presentation

5. Session

4. Transport

3. Network

2. Data Link

1. Physical

- A **single client** may have multiple transport connections with multiple servers.
- Notice that **TCP is a connection-oriented** service (two-way arrow) between the hosts, whereas **UDP is a connectionless** service (one-way arrow) . (later)

Service Ports

Transport Layer

Service Ports

Defined and managed by the Internet Assigned Numbers Authority and The Internet Corporation for Assigned Names and Numbers

- Well known ports (0-1023)
- Registered ports (1024 through 49151)
- Dynamic or Private ports (49152 through 65535)

Well known ports (AKA privileged ports) are intended to only be used by system or root processes or programs executed by privileged users.

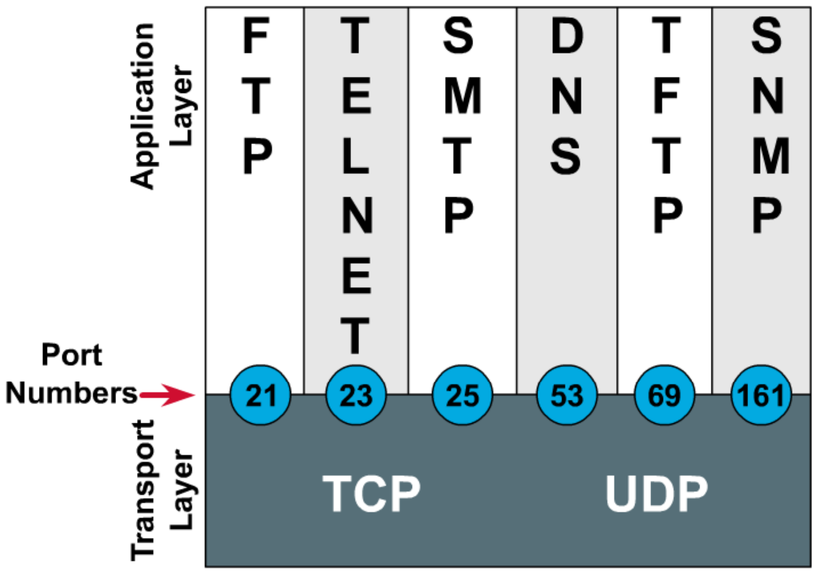
UDP Header

<table>
<tr><td>Source Port (16 bits)</td><td>Destination Port (16 bits)</td></tr>
<tr><td>Length (16 bits)</td><td>Checksum (16 bits)</td></tr>
<tr><td colspan="2">Data....</td></tr>
</table>

TCP Header

<table>
<tr><td>0</td><td colspan="7">15</td><td>16 31</td></tr>
<tr><td colspan="8">16-bit Source Port Number</td><td>16-bit Destination Port Number</td></tr>
<tr><td colspan="9">32-bit Sequence Number</td></tr>
<tr><td colspan="9">32 bit Acknowledgement Number</td></tr>
<tr><td>4-bit Header Length</td><td>6-bit (Reserved)</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>16-bit Window Size</td></tr>
<tr><td colspan="8">16-bit TCP Checksum</td><td>16-bit Urgent Pointer</td></tr>
<tr><td colspan="9">Options (if any)</td></tr>
<tr><td colspan="9">Data (if any)</td></tr>
</table>

Port Numbers

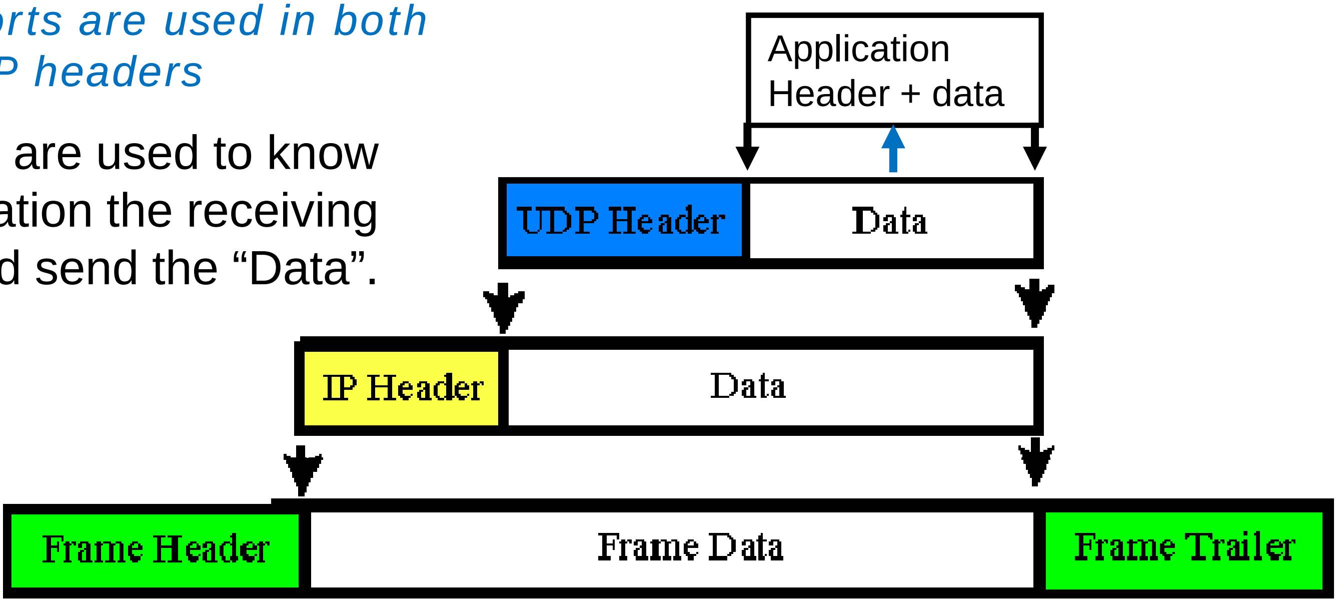


E.g. HTTP is Port 80

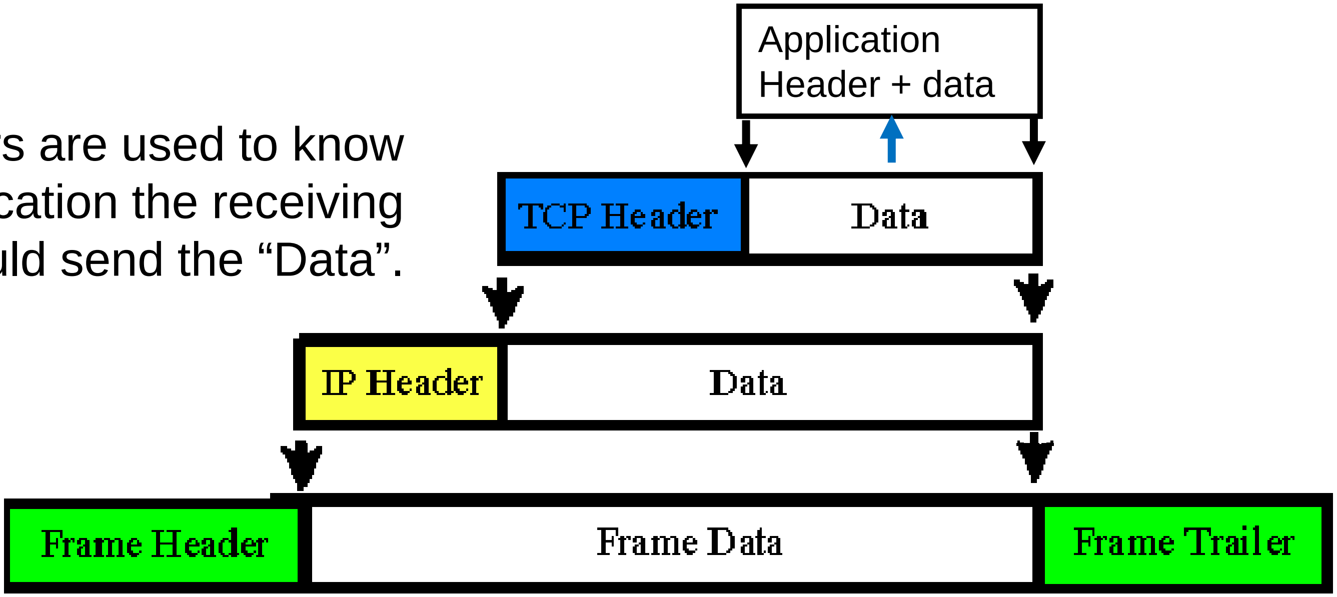
Both TCP and UDP use ports (or sockets) numbers to pass information to the upper layers.

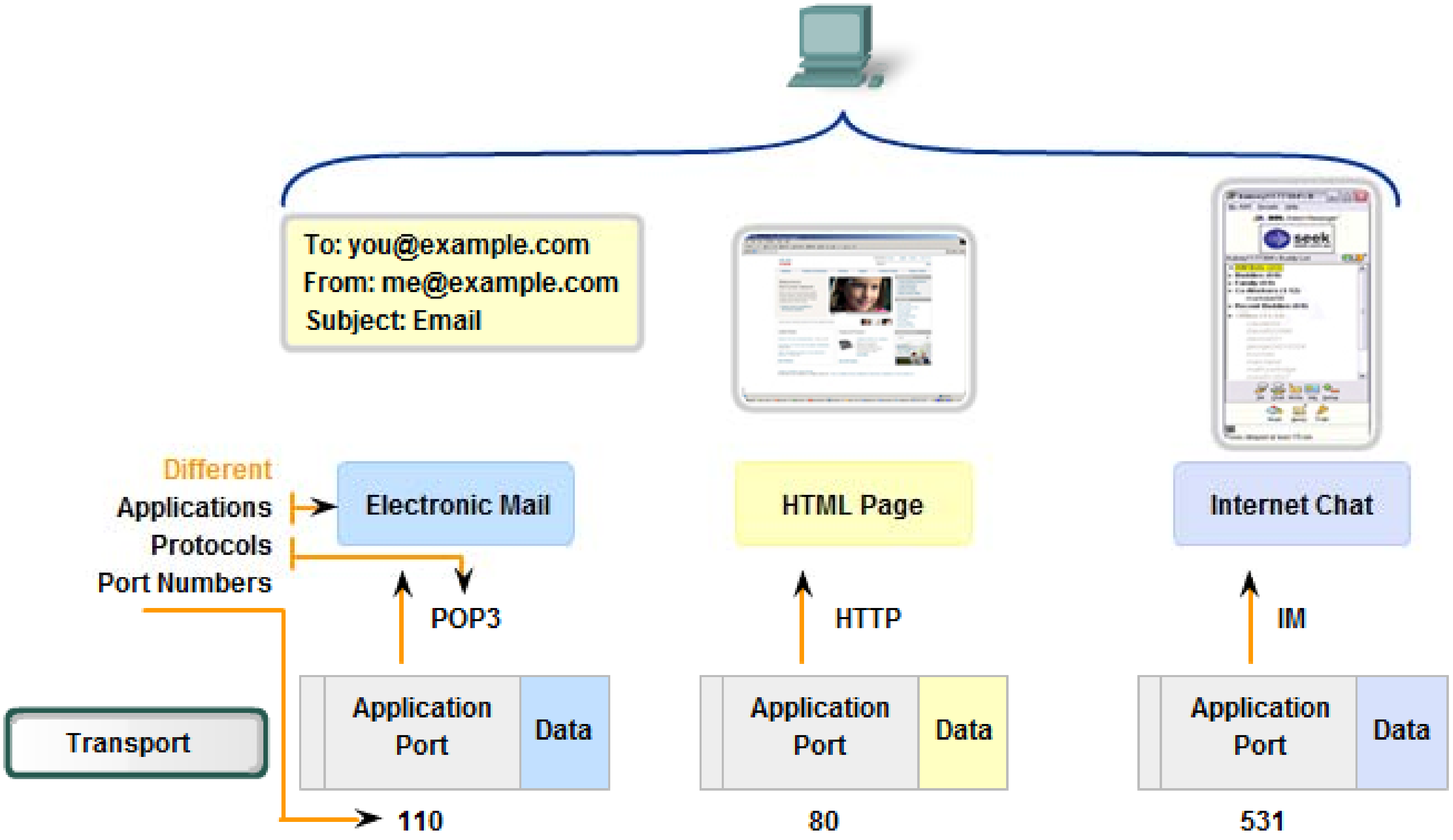
Note that ports are used in both UDP and TCP headers

Port numbers are used to know which application the receiving host should send the “Data”.



Port numbers are used to know which application the receiving host should send the “Data”.





To: you@example.com
From: me@example.com
Subject: Email

Different
Applications
Protocols
Port Numbers

Electronic Mail

HTML Page

Internet Chat

POP3

HTTP

IM

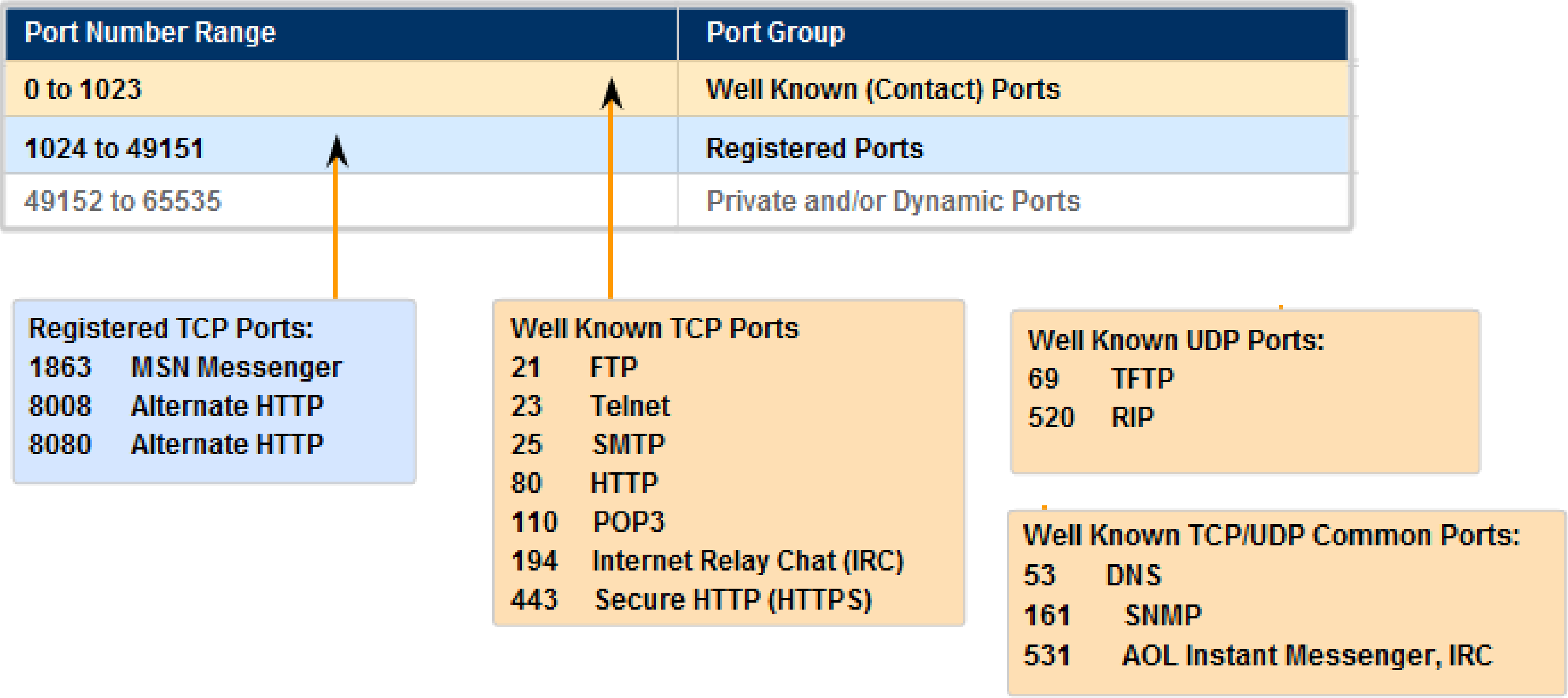
Transport

Application Port	Data
110	

Application Port	Data
80	

Application Port	Data
531	

Note that there are different port numbers, different protocols and different applications



Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

Registered TCP Ports:

1863	MSN Messenger
8008	Alternate HTTP
8080	Alternate HTTP

Well Known TCP Ports

21	FTP
23	Telnet
25	SMTP
80	HTTP
110	POP3
194	Internet Relay Chat (IRC)
443	Secure HTTP (HTTPS)

Well Known UDP Ports:

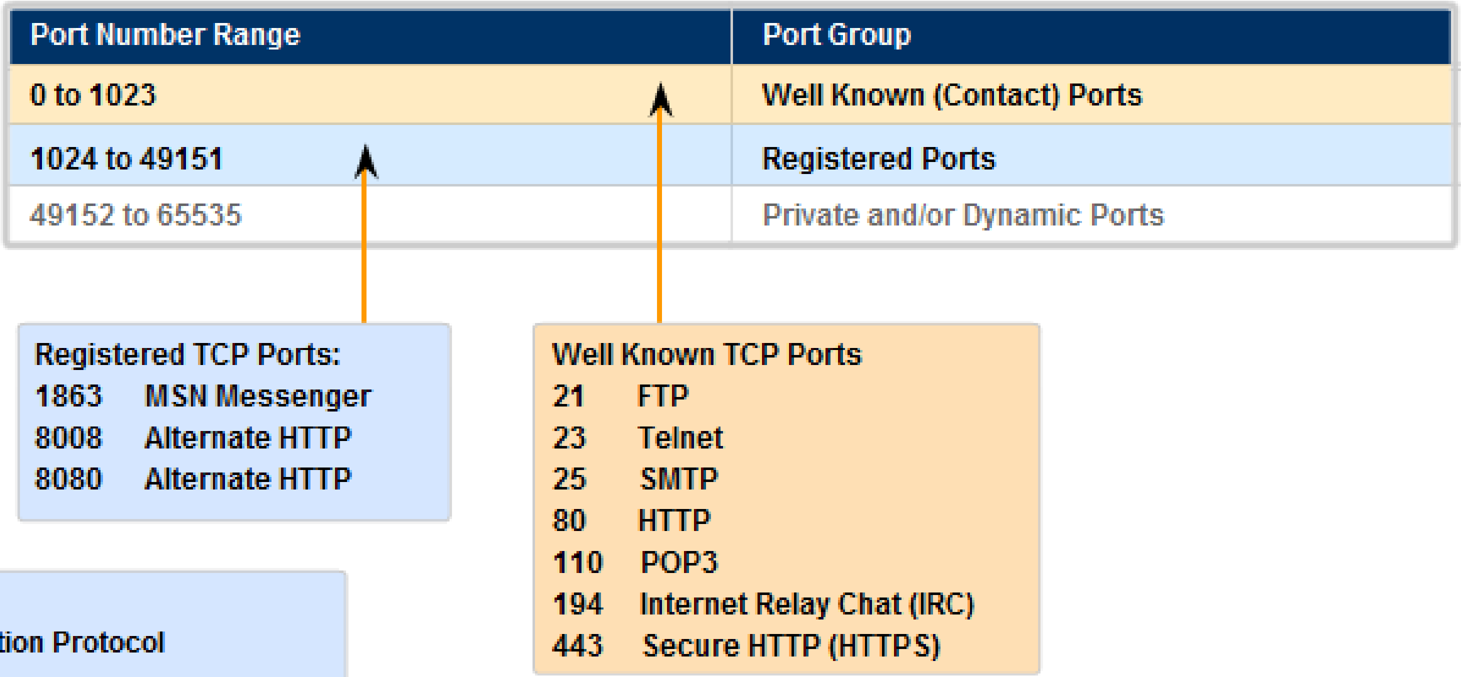
69	TFTP
520	RIP

Well Known TCP/UDP Common Ports:

53	DNS
161	SNMP
531	AOL Instant Messenger, IRC

- **Well Known Ports (Numbers 0 to 1023)**
  - Reserved for common services and applications.
  - HTTP (web server), POP3/SMTP (e-mail server) and Telnet.
    - **Client**: TCP destination port
    - **Server**: TCP source port

The well known ports are assigned to the common services



Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

Registered TCP Ports:

1863	MSN Messenger
8008	Alternate HTTP
8080	Alternate HTTP

Well Known TCP Ports

21	FTP
23	Telnet
25	SMTP
80	HTTP
110	POP3
194	Internet Relay Chat (IRC)
443	Secure HTTP (HTTPS)

Registered UDP Ports:

1812	RADIUS Authentication Protocol
2000	Cisco SCCP (VoIP)
5004	RTP (Voice and Video Transport Protocol)
5060	SIP (VoIP)

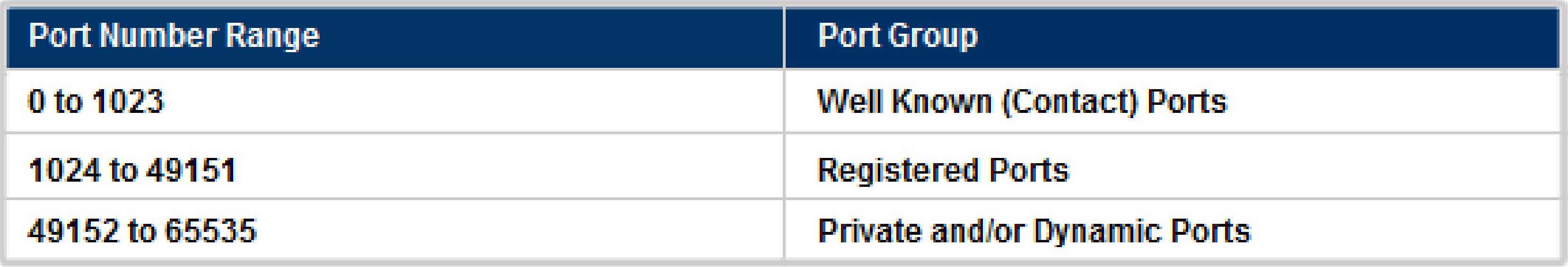
Registered TCP/UDP Common Ports:

1433	MS SQL
2948	WAP (MMS)

Registered Ports (Numbers 1024 to 49151)

- Assigned to user processes or applications.
- Non-common applications.
  - **Client**: TCP destination port
  - **Server**: TCP source port
- May also be used as dynamic or private port (next).

The well known ports are assigned to less common services



Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

- **Dynamic or Private Ports (Numbers 49152 to 65535)**
  - Also known as Ephemeral Ports
  - Usually assigned dynamically to client applications when initiating a connection.
    - **Client**: TCP source port
    - **Server**: TCP destination port
  - May also include the range of Registered Ports (Numbers 1024 to 49151)
  - Note: Some peer-to-peer file sharing programs use these ports as Register Ports. (previous slide)

The dynamic ports are used by clients for making connections

Service Ports

Well-known and registered ports listed in /etc/services

```
[root@elrond ~]# cat /etc/services | more
# /etc/services:
# $Id: services,v 1.42 2006/02/23 13:09:23 pknirsch Exp $
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
# Updated from RFC 1700, ``Assigned Numbers'' (October 1994).  Not all ports
# are included, only the more common ones.
#
# The latest IANA port assignments can be gotten from
#       http://www.iana.org/assignments/port-numbers
# The Well Known Ports are those from 0 through 1023.
# The Registered Ports are those from 1024 through 49151
# The Dynamic and/or Private Ports are those from 49152 through 65535
#
# Each line describes one service, and is of the form:
#
# service-name  port/protocol  [aliases ...]   [# comment]

tcpmux          1/tcp                           # TCP port service multiplexer
tcpmux          1/udp                           # TCP port service multiplexer
rje             5/tcp                           # Remote Job Entry
rje             5/udp                           # Remote Job Entry

< snipped >
```

Service Ports

some favorites from /etc/services file

```
< 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 >
```

Exercise

1. Browse the port definitions using **less /etc/services**

2. Browse the protocol definitions using **less/etc/protocols**

Use quit to exit the less command

Sockets

Transport Layer

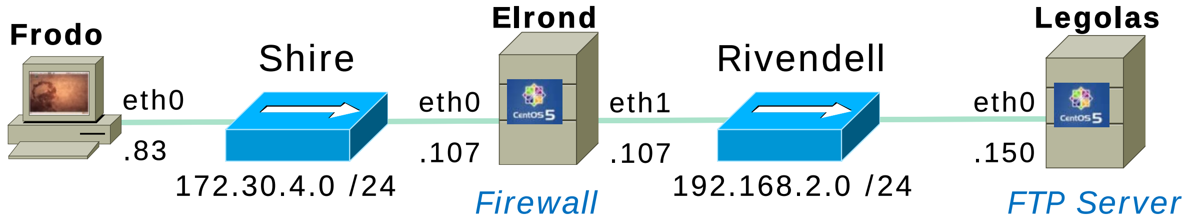
Sockets

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

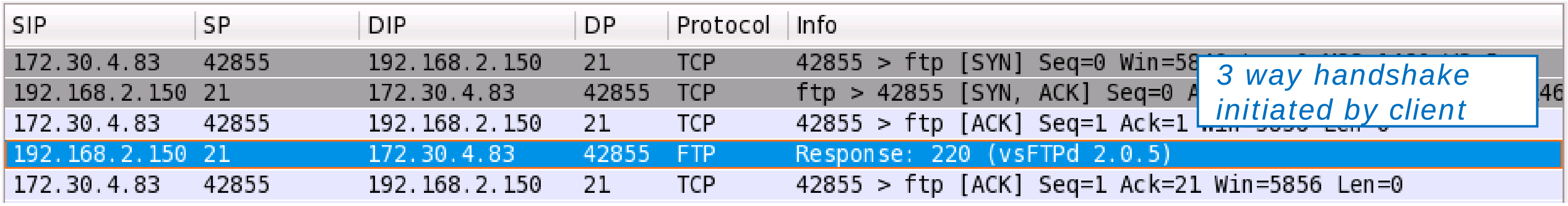
The socket is associated to a port number so that the TCP layer can identify the application to send data to.

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



```
root@frodo:~# ftp legolas
Connected to legolas.
220 (vsFTPd 2.0.5)
```

Frodo FTP's into Legolas



SIP	SP	DIP	DP	Protocol	Info
172.30.4.83	42855	192.168.2.150	21	TCP	42855 > ftp [SYN] Seq=0 Win=58
192.168.2.150	21	172.30.4.83	42855	TCP	ftp > 42855 [SYN, ACK] Seq=0 A 46
172.30.4.83	42855	192.168.2.150	21	TCP	42855 > ftp [ACK] Seq=1 Ack=1
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 220 (vsFTPd 2.0.5)
172.30.4.83	42855	192.168.2.150	21	TCP	42855 > ftp [ACK] Seq=1 Ack=21 Win=5856 Len=0

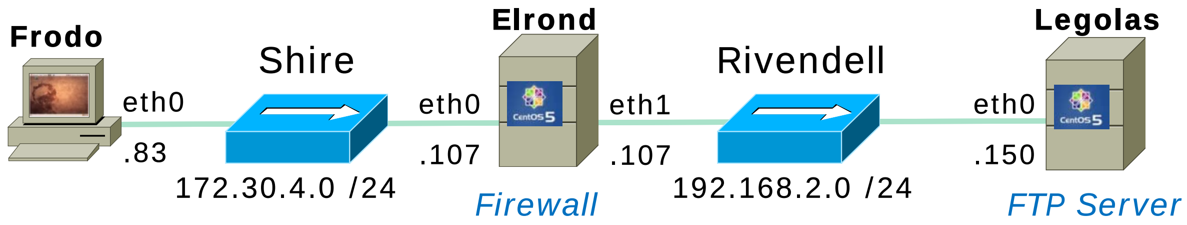
3 way handshake initiated by client

- *3 way handshake*
- *New connection initiated by client*

Socket for commands

Client	Server
172.30.4.83	192.168.2.150
42855	21

More on FTP and sockets later ...



Socket for commands

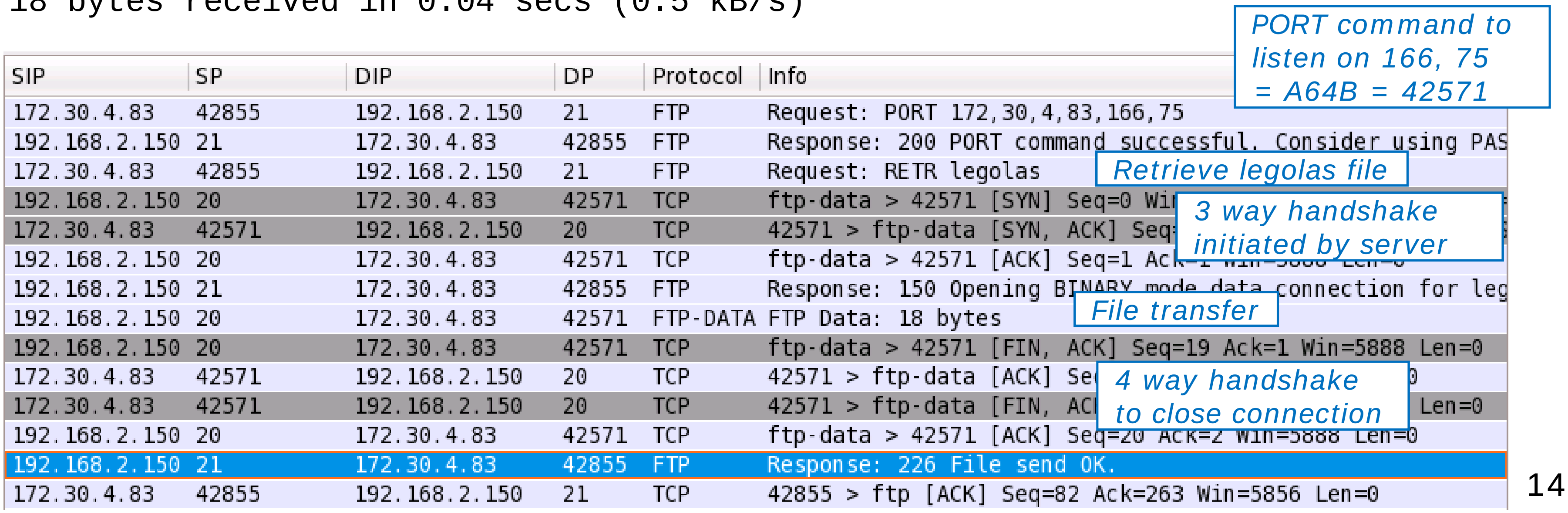
Client	Server
172.30.4.83	192.168.2.150
42855	21

Socket for data transfer

Client	Server
172.30.4.83	192.168.2.150
42571	20

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

```
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 Win=5888 Len=0
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                    0
172.30.4.83    42571  192.168.2.150  20     TCP      42571 > ftp-data [FIN, ACK                   Len=0
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
```

PORT command to listen on 166, 75 = A64B = 42571

Retrieve legolas file

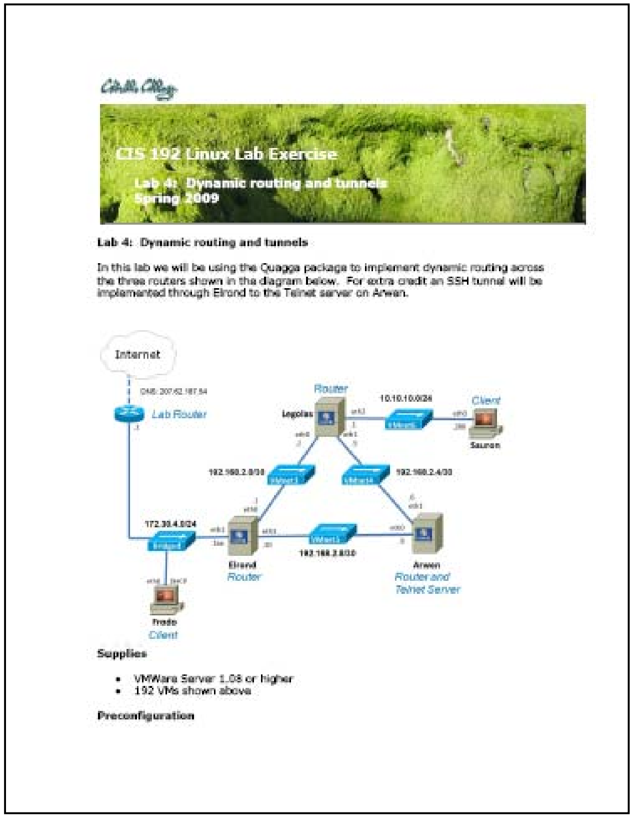
3 way handshake initiated by server

File transfer

4 way handshake to close connection

Lab 4 covers dynamic routing and SSH tunneling. It due in two weeks and the SSH tunneling is extra credit

Lab X1 is a repeat of Lab 3 except the NIC configuration is permanent. This is an extra credit lab.



CIS 192 Linux Lab Exercise

Lab 4: Dynamic routing and tunnels
Spring 2009

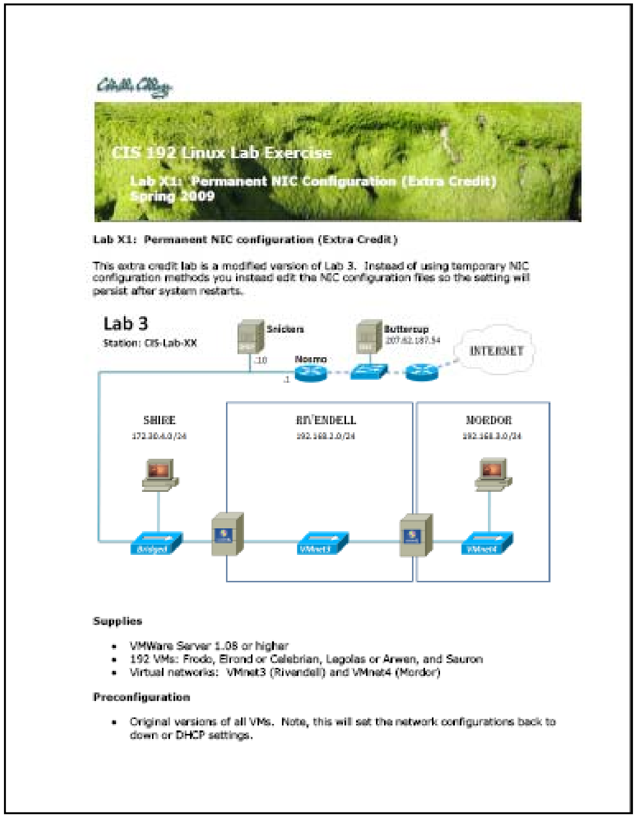
Lab 4: Dynamic routing and tunnels

In this lab we will be using the Quagga package to implement dynamic routing across the three routers shown in the diagram below. For extra credit an SSH tunnel will be implemented through Elrond to the Telnet server on Arwen.

Supplies

- VMWare Server 1.08 or higher
- 192 VMs shown above

Preconfiguration



CIS 192 Linux Lab Exercise

Lab X1: Permanent NIC Configuration (Extra Credit)
Spring 2009

Lab X1: Permanent NIC configuration (Extra Credit)

This extra credit lab is a modified version of Lab 3. Instead of using temporary NIC configuration methods you instead edit the NIC configuration files so the setting will persist after system restarts.

Supplies

- VMWare Server 1.08 or higher
- 192 VMs: Frodo, Elrond or Celebrian, Legolas or Arwen, and Sauron
- Virtual networks: VMnet3 (Rivendell) and VMnet4 (Mordor)

Preconfiguration

- Original versions of all VMs. Note, this will set the network configurations back to down or DHCP settings.

Start early on this lab … it's a beefy one!

Wrap

New commands, tools and services:

```
init 5
iptables –L -–line-numbers
service ripd restart
service xinetd restart
service zebra restart
startx
telnet localhost 2601
telnet localhost 2602
vtysh
yum install quagga
```

New Files and Directories:

```
/etc/quagga/ripd.conf
/etc/quagga/zebra.conf
/etc/services
/etc/sysconfig/iptables
/etc/xinetd.d/telnet
```

Next Class

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

Test next week on lessons 1 through 4

- Open book, open notes, open VMs, during last hour of class
- 15 questions (2 points each)
- Practice test available
- Doing Lab 4 early would be good practice for test

Students may work together and use the forum to work out the answers on the practice test.

The actual test will be ***almost identical*** *to the practice test.*

For the actual test, students must work individually and neither ask nor give assistance to others.

Backup

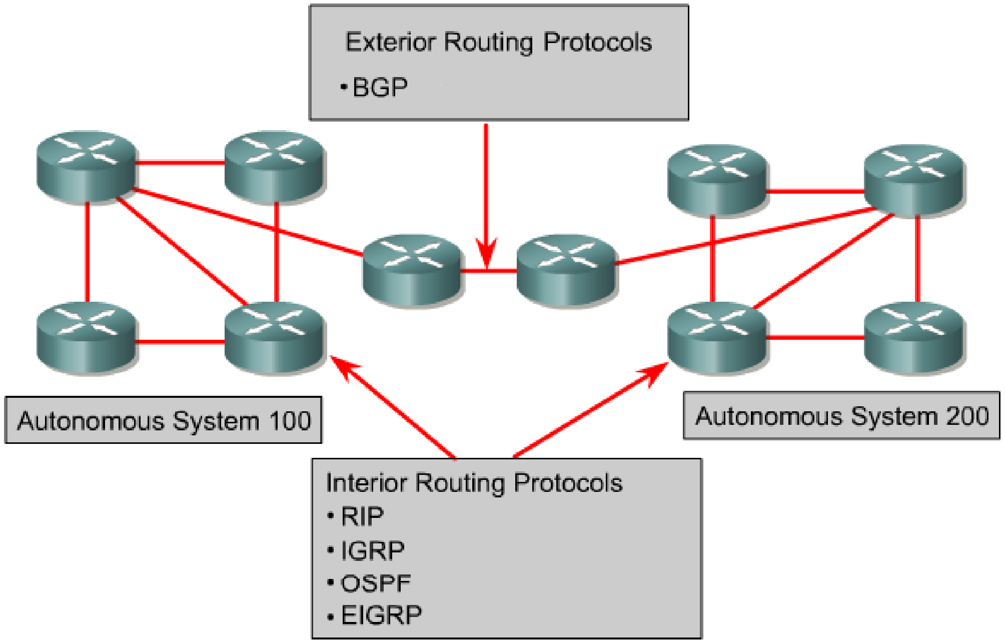
IP addresses for VM's in the classroom

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

Routing Protocols



"An AS is a connected group of one or more IP prefixes run by one or more network operators which has a SINGLE and CLEARLY DEFINED routing policy." (RFC 1930)

ISPs and large organizations are assigned a unique ASN (Autonomous System Number) for use with BGP routing.

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

*Some **Distance** **Vector** routing protocols*

(The Cost) (The Direction)

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

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

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

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

EIGRP is a Cisco proprietary enhanced distance vector routing protocol.

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

Link-state routing protocols – each node knows the entire network topology and can compute the shortest paths

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

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

Intermediate System to Intermediate System (IS-IS)

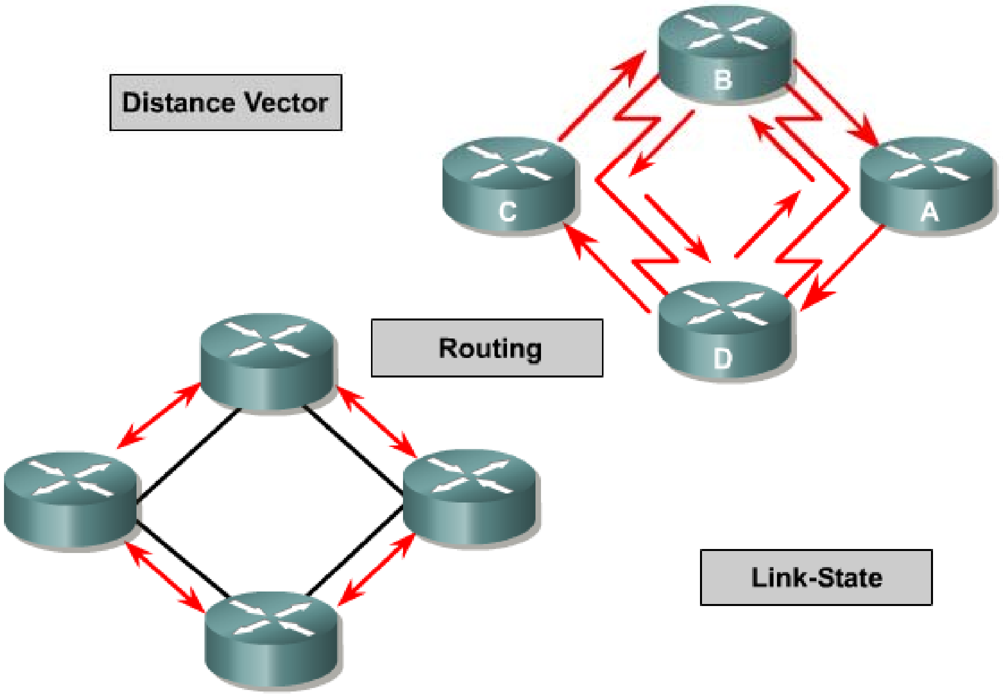
- IS-IS is an Open System Interconnection (OSI) routing protocol originally specified by International Organization for Standardization (ISO) 10589.
- It is a **link-state** routing protocol.

Exterior routing protocols – used between autonomous systems

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

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

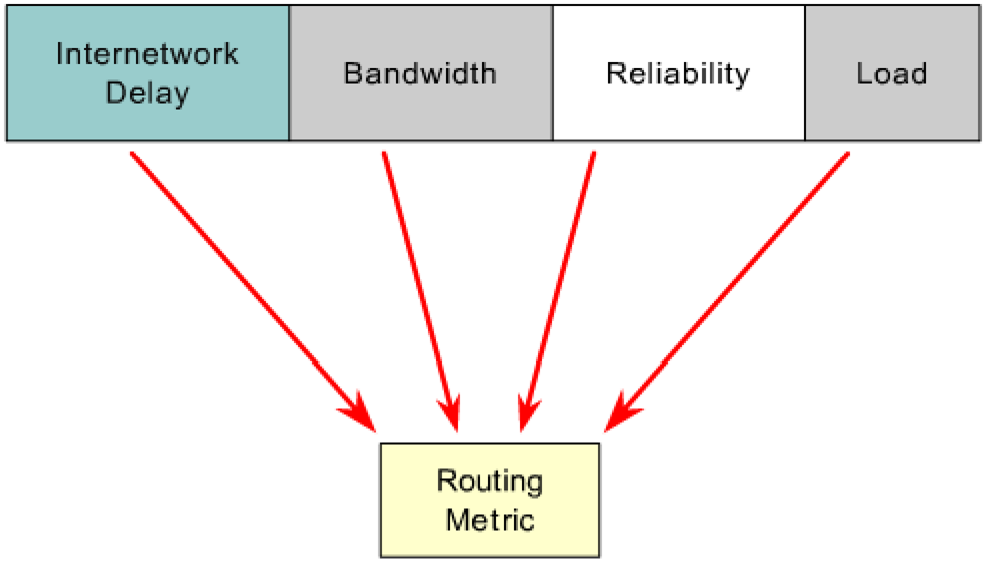
Types of Routing Protocols



- Distance Vector: RIP, IGRP, EIGRP
- Link State: OSPF, IS-IS
- Path Vector: BGP
- Note: IGRP and EIGRP are Cisco Proprietary

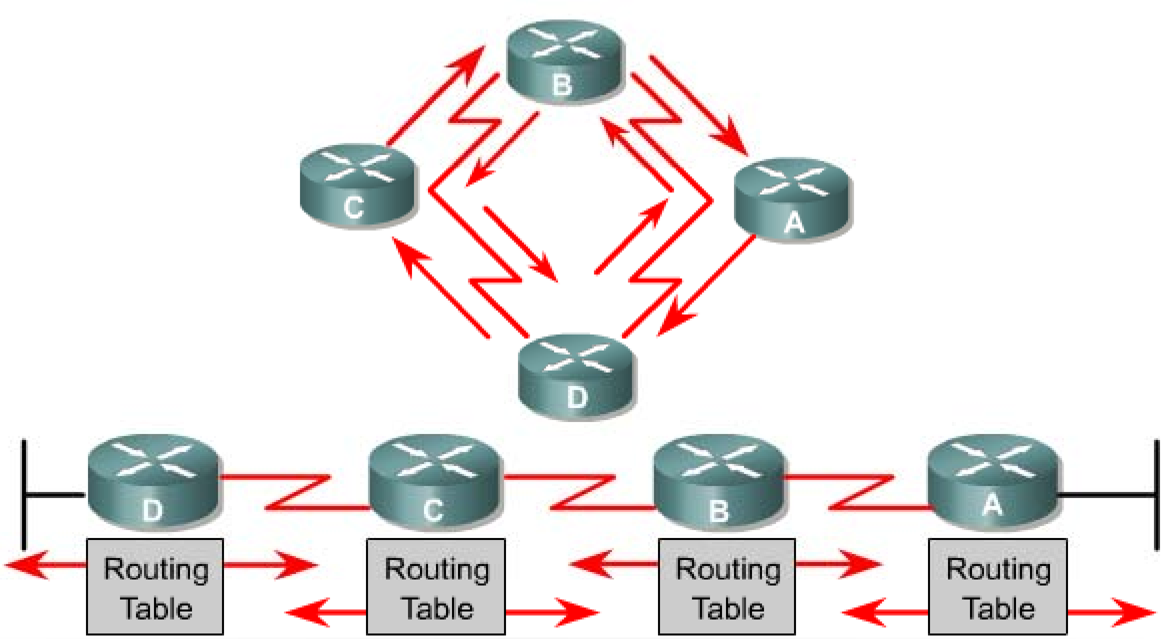
Path vector protocols (like BGP) are a class of distance vector protocols and not a link-state protocol

Routing Protocol Metrics (costs)



- RIP – Hop Count
- IGRP and EIGRP – Bandwidth, Delay, Reliability, Load
- Cisco's OSPF – Bandwidth
- IS-IS – Cost
- BGP – Number of AS or policy

Distance Vector Routing Protocols



Pass periodic copies of a routing table to neighbor routers and accumulate distance vectors.

Router B receives information from Router A.

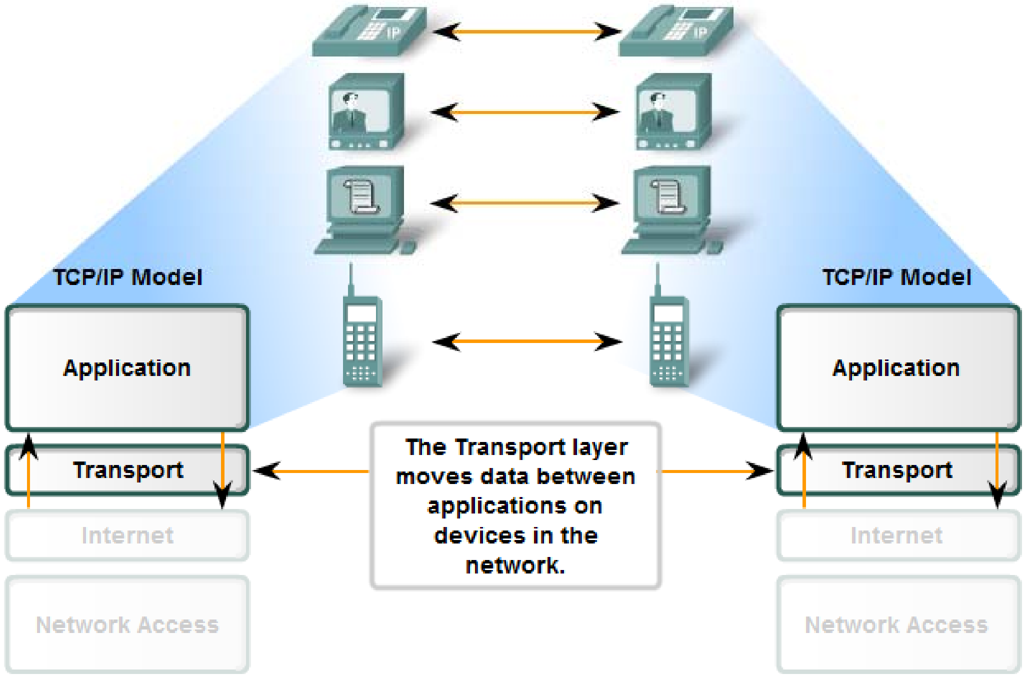
Router B adds a distance vector number (such as a number of hops), which increases the distance vector.

Then Router B passes this new routing table to its other neighbor, Router C.

This same step-by-step process occurs in all directions between neighbor routers.

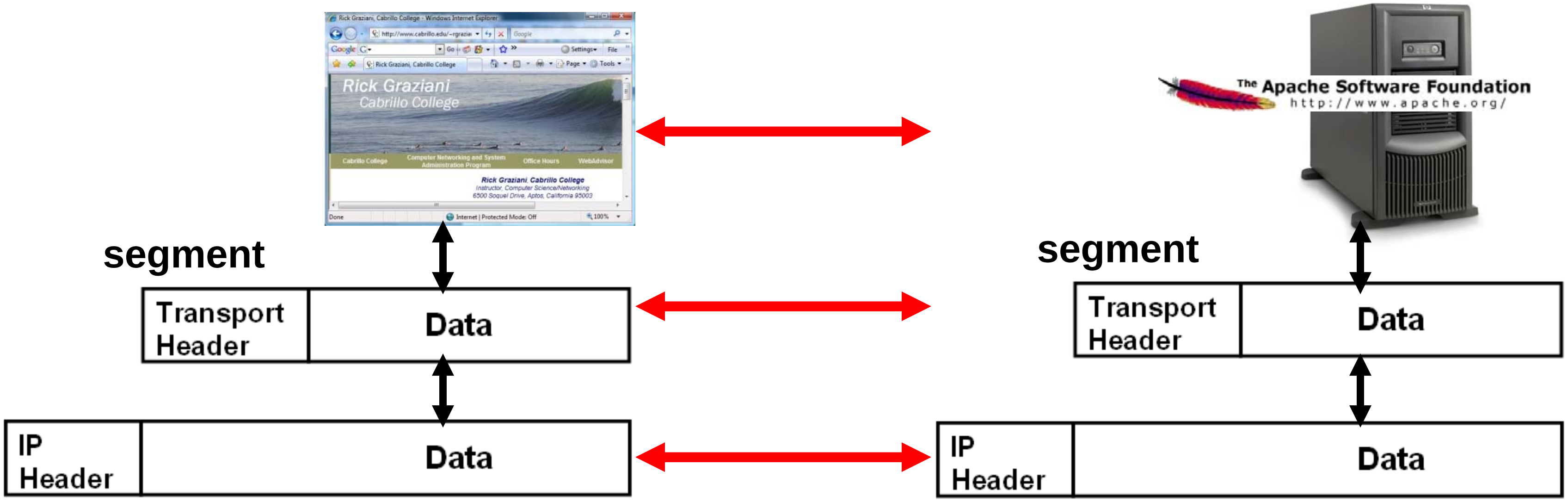
- “Routing by rumor”
- Each router receives a routing table from its directly connected neighbor routers.

Transport Layer



The Transport layer moves data between applications on devices in the network.

- Primary responsibilities:
  - Tracking the individual communication between applications
  - Segmenting data
  - Managing each segment
  - Reassembling the segments
  - Identifying the different applications



segment

Transport Header | Data

IP Header | Data

segment

Transport Header | Data

IP Header | Data

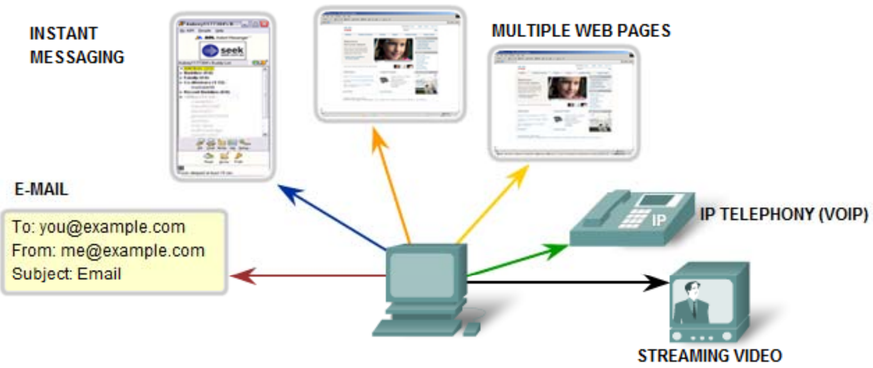
Transport Layer

- Protocols:
  - **TCP**
  - **UDP**
- **IP** is a best-effort delivery service
  - No guarantees
  - Best-effort service
  - “Unreliable service”
- TCP/UDP is responsible for extending IP’s delivery service between two end systems.
  - Known as transport layer **multiplexing** and **demultiplexing**.

Breaking up into little pieces and reassembling at the end

TCP vs. UDP

Transport Layer Services



To: you@example.com
From: me@example.com
Subject: Email

Establishing a Session ensures the application is ready to receive the data.

Reliable delivery means lost segments are resent so the data is received complete.

Same order delivery ensures data is delivered sequentially as it was sent.

Flow Control manages data delivery if there is congestion on the host.

TCP provides:

Reliable delivery
Error checking
Flow control
Congestion control
Ordered delivery
(Connection establishment)
Applications:
HTTP
FTP
Telnet
MSN messenger

UDP provides:

- Unreliable delivery
- No error checking
- No flow control
- No congestion control
- No ordered delivery
- (No connection establishment)
- Applications
  - DNS (usually)
  - SMTP
  - DHCP
  - RTP (Real-Time Protocol)
  - VoIP

and SNMP “fire and forget” traps, RIP updates

Transmission Control Protocol

Transport Layer

The Transmission Control Protocol

More on this later…

We want to be able to identify the start, flow and end of TCP connections as we start exploring network services.

Some quick preview examples for now

Initial Connection

Three-Way Handshake

1. SYN
2. SYN-ACK
3. ACK

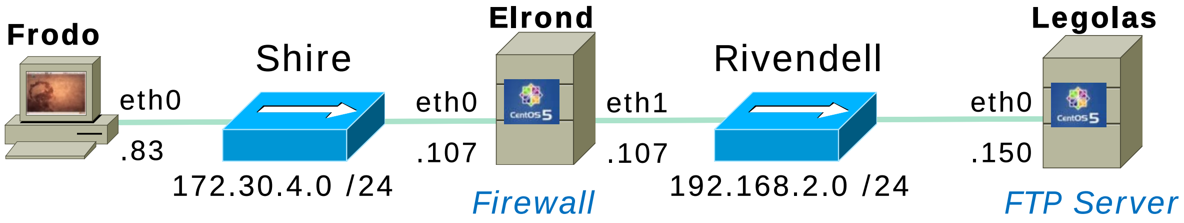
Continuing Communications

- o The Sliding Window
- o Flow Control (cumulative acknowledgment)
- o SACK
- o The RST Flag

Closing a Connection

Four-Way Handshake

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



Socket for commands

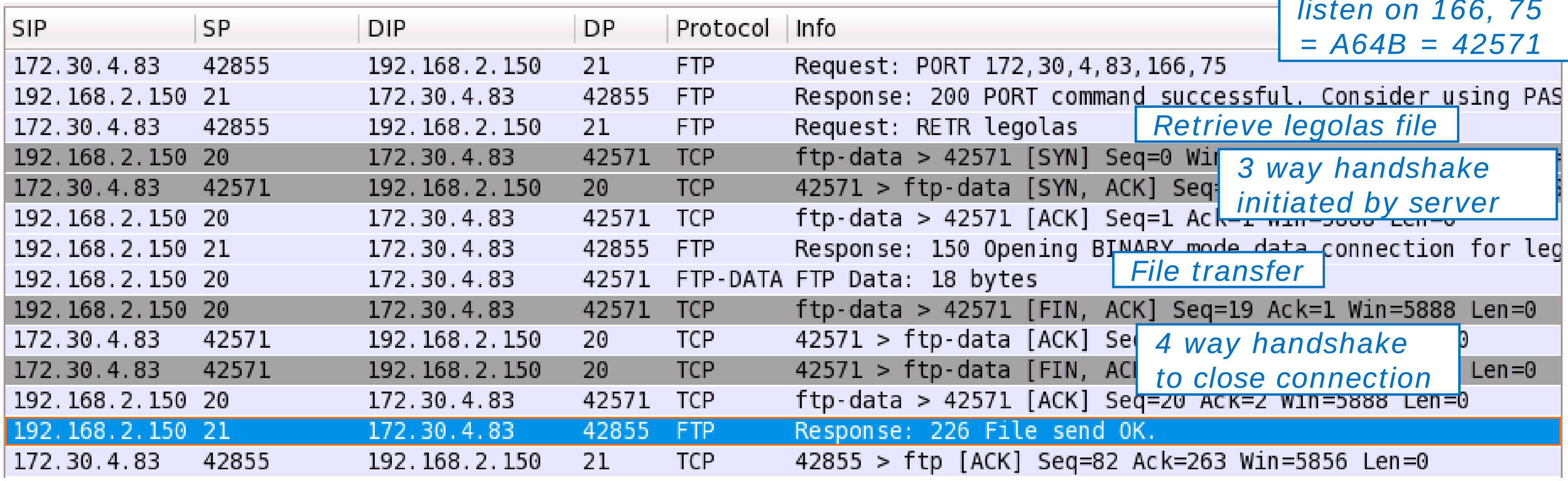
Client	Server
172.30.4.83	192.168.2.150
42855	21

Socket for data transfer

Client	Server
172.30.4.83	192.168.2.150
42571	20

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

```
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 Win=5888 Len=0
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 0
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [FIN, ACK Len=0
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

PORT command to listen on 166, 75 = A64B = 42571

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
tcp_keepalive_time
tcp_sack
tcp_timestamps
tcp_window_scaling
tcp_retries1
tcp_retries2
tcp_syn_retries

Security Issues

Transport Layer

Security Issues

Resource: *www.securityfocus.org*

- SYN Flooding
- Falsifying TCP Communications
- Hijacking connections