Lesson Module Status

- Wall updated and emailed

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

- Lab tested –
- Lab template in depot -
- Youtube Videos uploaded –

- VM (Classroom PC) –
- VMs (VLab) - extra gondor and arnor switches made for each pod

- Headset charged –

- Special – test published/locked
- Bring MikroTik router - done

CCC Confer

[ ] Has the phone bridge been added?

[ ] Is phone being used for voice input?

[ ] Is recording on?

[ ] Share slides, putty (rsimms, simben192), Chrome, vlab192.rdp, wireshark

[ ] Disable spelling on PowerPoint

[ ] Repeat all ?'s for remote students

[ ] Remote student proxy

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/

CIS 192 – Lesson 4

James

Lars

Instructor: **Rich Simms**

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

Passcode: **761867**

Daniel

Elizabeth

Carlos V

Brandon

Chad

Donovan

Leopoldo

Jacob G

Jeff

Timothy

Jacob S

Laura

Gabriel V

Jason

Thomas

Josh

Carlos R

Geoffrey

Ellison

Mark

David

Leandro

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

First Minute Quiz

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

No Quiz Today

(Test instead during last part of class)

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

Routing Continued and Transport Protocols

Objectives

- 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.

Agenda

- 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
- Practice test
- How this class works

Housekeeping

- VLab Pod Reservations – **MUST USE FANG!**

- Lab 3 due midnight
- Five posts due midnight

- Test 1 during the last part of class

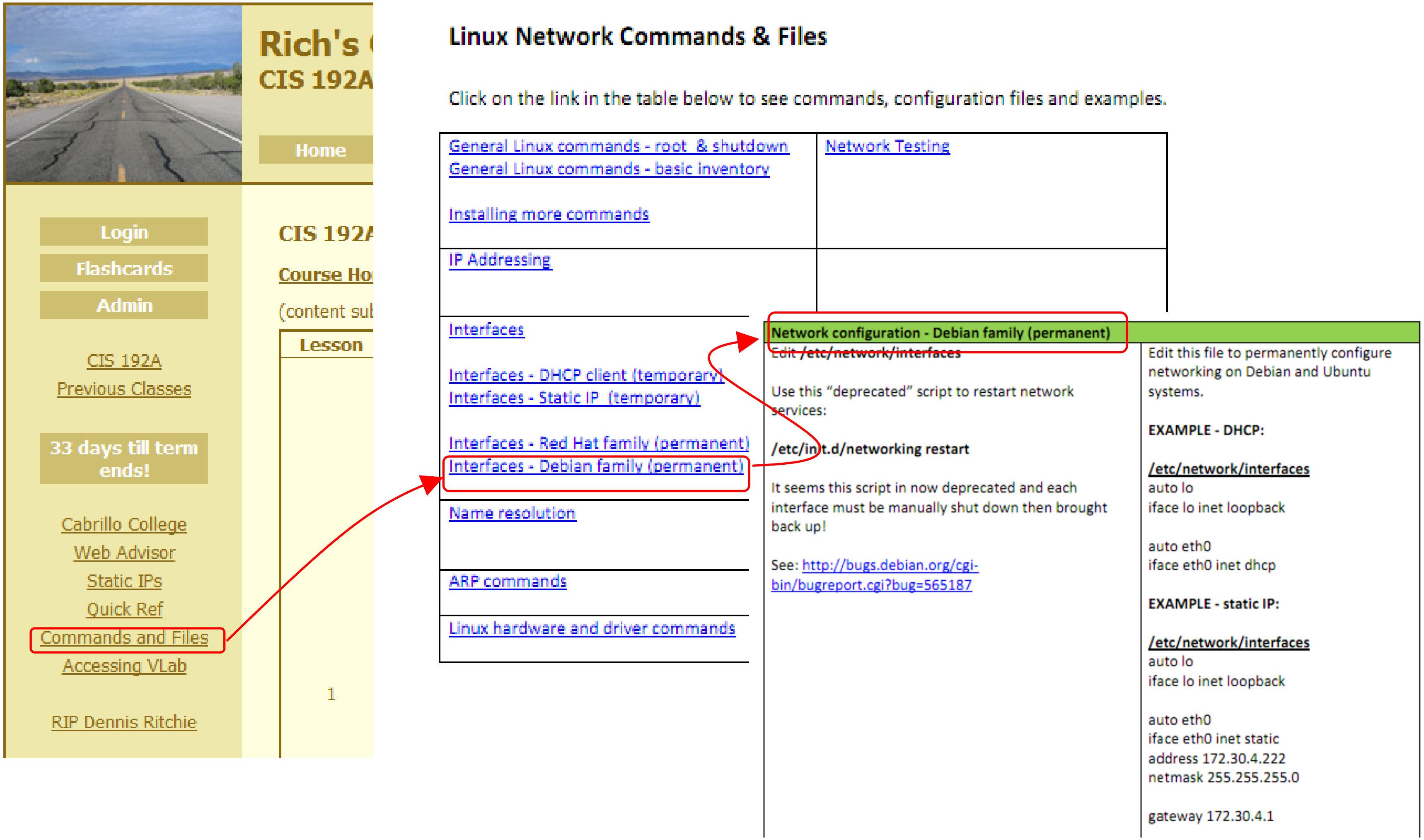
/home/cis192/answers directory on Opus

```
[rsimms@opus ~]$ cat /home/cis192/answers/lab01.simben192
CIS 192A Lab 1
Fall 2011

Name: Benji Simms
Date started: 09/13/2011
Date completed: 09/13/2011
Time spent doing this lab: 2 Hours

Step 1
------
Logged into Opus from: dsl-63-249-103-107.


Step 2
------
CentOS VM:   [root@elrond ~]#
Ubuntu VM:   root@frodo:~#

< snipped >
```

```
[rsimms@opus ~]$ cat /home/cis192/answers/lab02.simben192
CIS 192A Lab 2
Fall 2011

Name: Benji Simms
Date started: 11/07/2011
Date completed: 11/07/2011
Time spent doing this lab: 4 hours
CIS Lab Station or VLab pod: Pod 7


Step 3 - Frodo network information
----------------------------------

eth0 NIC hardware and model: VMware VMXNET3 Ethernet Controller (rev 01)
eth0 NIC driver: vmxnet3
eth0 IPv4 address: 172.30.4.153
Default Gateway: 172.30.4.1
DNS servers: 192.168.0.8 10.240.1.2


Step 4 - How to repair Frodo after its removing NIC driver
----------------------------------------------------------

modprobe vmxnet3
ping google.com

< snipped >
```

*After labs are graded, example lab reports are placed in the **/home/cis192/answers** directory*

Commands and Files
Quick Reference and Examples



Linux Network Commands & Files

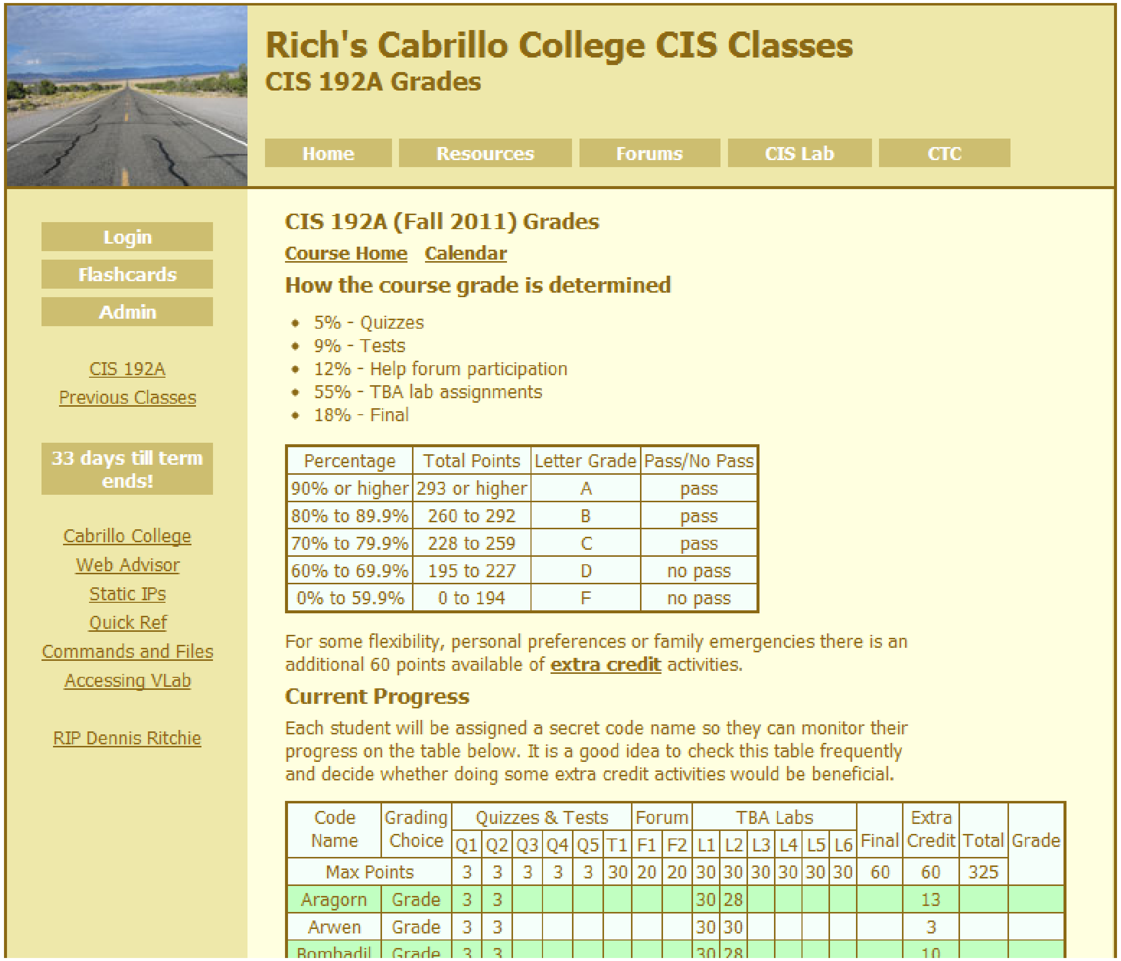
Click on the link in the table below to see commands, configuration files and examples.

General Linux commands - root & shutdown General Linux commands - basic inventory Installing more commands	Network Testing
IP Addressing	
Interfaces Interfaces - DHCP client (temporary) Interfaces - Static IP (temporary) Interfaces - Red Hat family (permanent) Interfaces - Debian family (permanent)	
Name resolution	
ARP commands	
Linux hardware and driver commands	

Network configuration - Debian family (permanent)	
Edit /etc/network/interfaces Use this "deprecated" script to restart network services: **/etc/init.d/networking restart** It seems this script in now deprecated and each interface must be manually shut down then brought back up! See: http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=565187	Edit this file to permanently configure networking on Debian and Ubuntu systems. **EXAMPLE - DHCP:** **/etc/network/interfaces** auto lo iface lo inet loopback auto eth0 iface eth0 inet dhcp **EXAMPLE - static IP:** **/etc/network/interfaces** auto lo iface lo inet loopback auto eth0 iface eth0 inet static address 172.30.4.222 netmask 255.255.255.0 gateway 172.30.4.1

Grades

http://simms-teach.com/cis192Agrades.php



Rich's Cabrillo College CIS Classes

CIS 192A Grades

CIS 192A (Fall 2011) Grades

Course Home   Calendar

How the course grade is determined

- 5% - Quizzes
- 9% - Tests
- 12% - Help forum participation
- 55% - TBA lab assignments
- 18% - Final

Percentage	Total Points	Letter Grade	Pass/No Pass
90% or higher	293 or higher	A	pass
80% to 89.9%	260 to 292	B	pass
70% to 79.9%	228 to 259	C	pass
60% to 69.9%	195 to 227	D	no pass
0% to 59.9%	0 to 194	F	no pass

For some flexibility, personal preferences or family emergencies there is an additional 60 points available of **extra credit** activities.

Current Progress

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

Code Name	Grading Choice	Quizzes & Tests						Forum		TBA Labs						Final	Extra Credit	Total	Grade
		Q1	Q2	Q3	Q4	Q5	T1	F1	F2	L1	L2	L3	L4	L5	L6				
Max Points		3	3	3	3	3	30	20	20	30	30	30	30	30	30	60	60	325	
Aragorn	Grade	3	3							30	28						13		
Arwen	Grade	3	3							30	30						3		
Bombadil	Grade	3	3							30	28						10		

Please review your grading choice and grades for accuracy

*Email me at **risimms@cabrillo.edu** to get your LOTR codename*

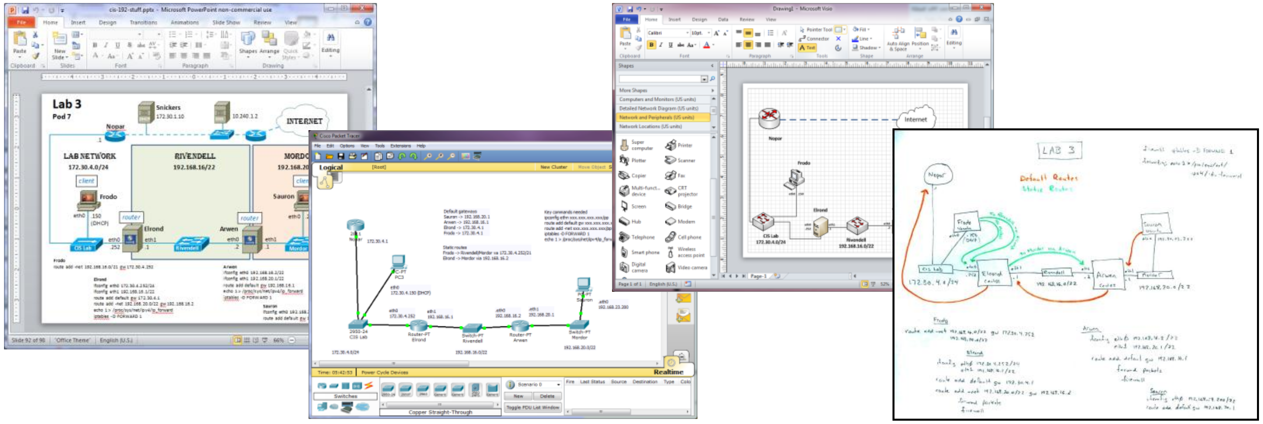
Network Topology Tools

Network Topology Diagrams

Plan & Design > Implement > Troubleshoot/Test

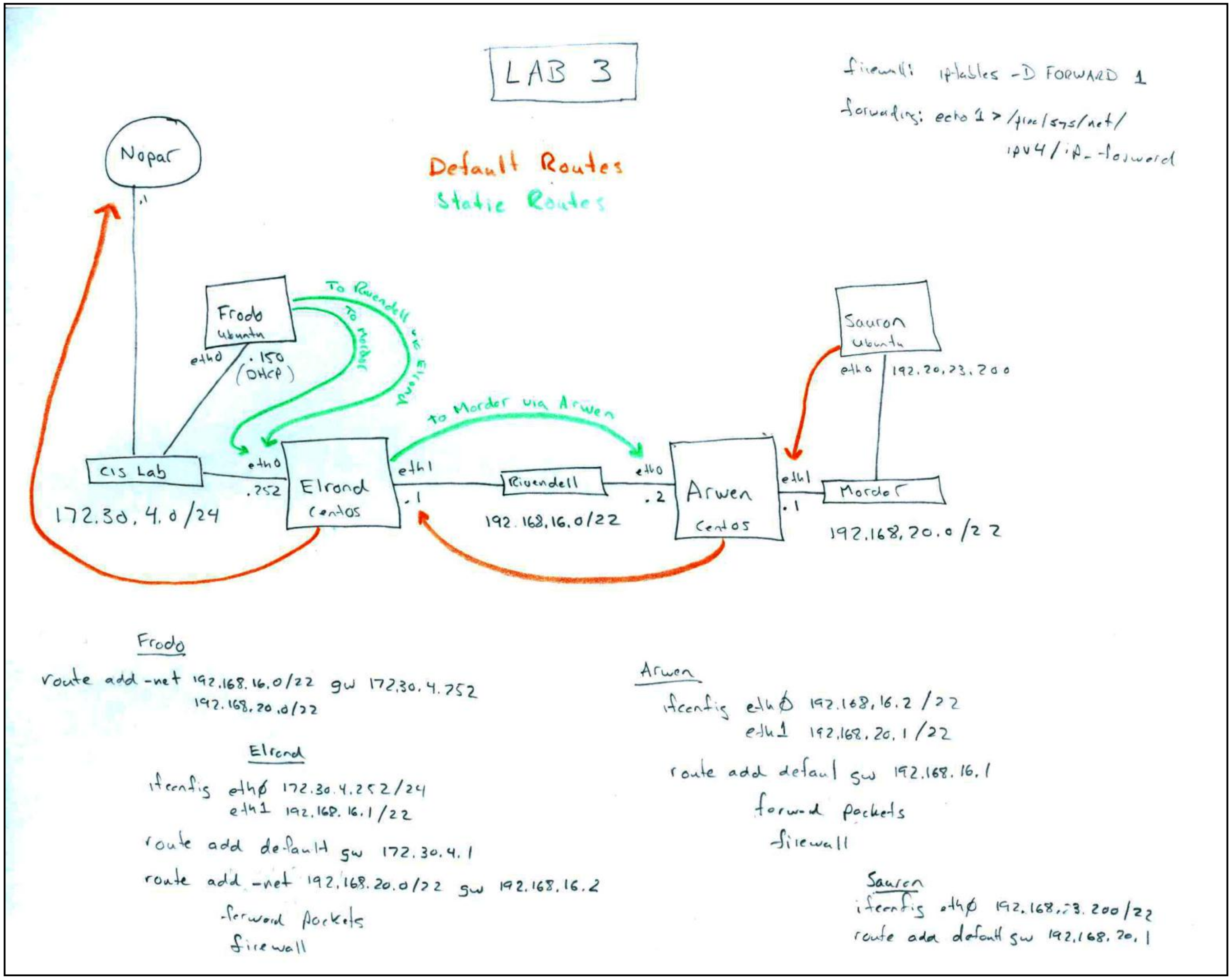
Spending a little time here

... will save you a lot of time here



*TIP: **Save LOTS OF TIME doing CIS 192 labs** by making a network topology diagram **FIRST**. Document the networks, devices, interfaces, addresses, routes and include any key commands/files you will need before implementing the lab.*

CIS 192 – Lesson 4



firewall: iptables -D FORWARD 1

forwarding: echo 1 > /proc/sys/net/
ipv4/ip_forward

Frodo

route add -net 192.168.16.0/22 gw 172.30.4.252
192.168.20.0/22

Elrond

ifconfig eth0 172.30.4.252/24
eth1 192.168.16.1/22

route add default gw 172.30.4.1

route add -net 192.168.20.0/22 gw 192.168.16.2

forward packets
firewall

Arwen

ifconfig eth0 192.168.16.2/22
eth1 192.168.20.1/22

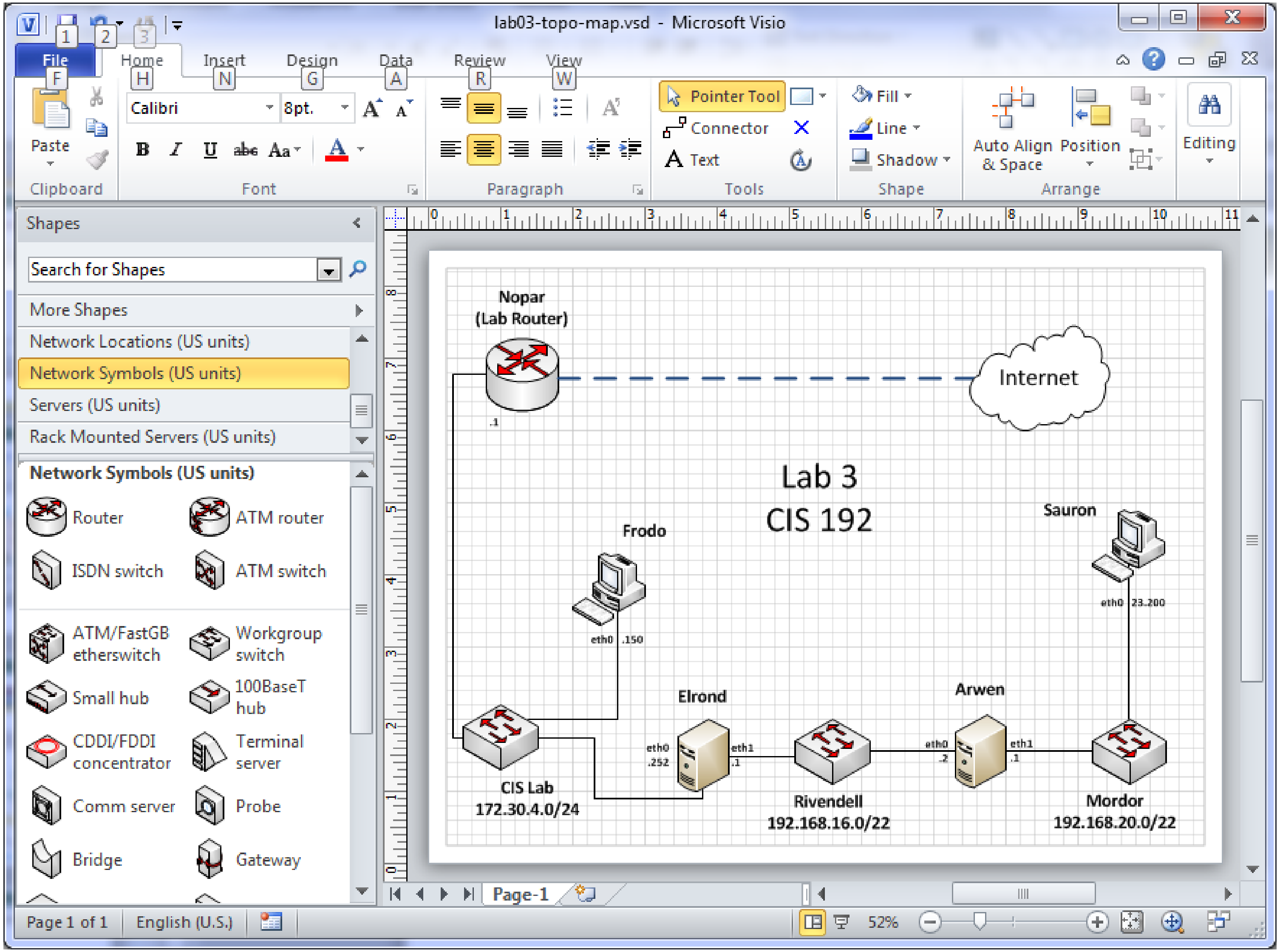
route add defaul gw 192.168.16.1

forward packets
firewall

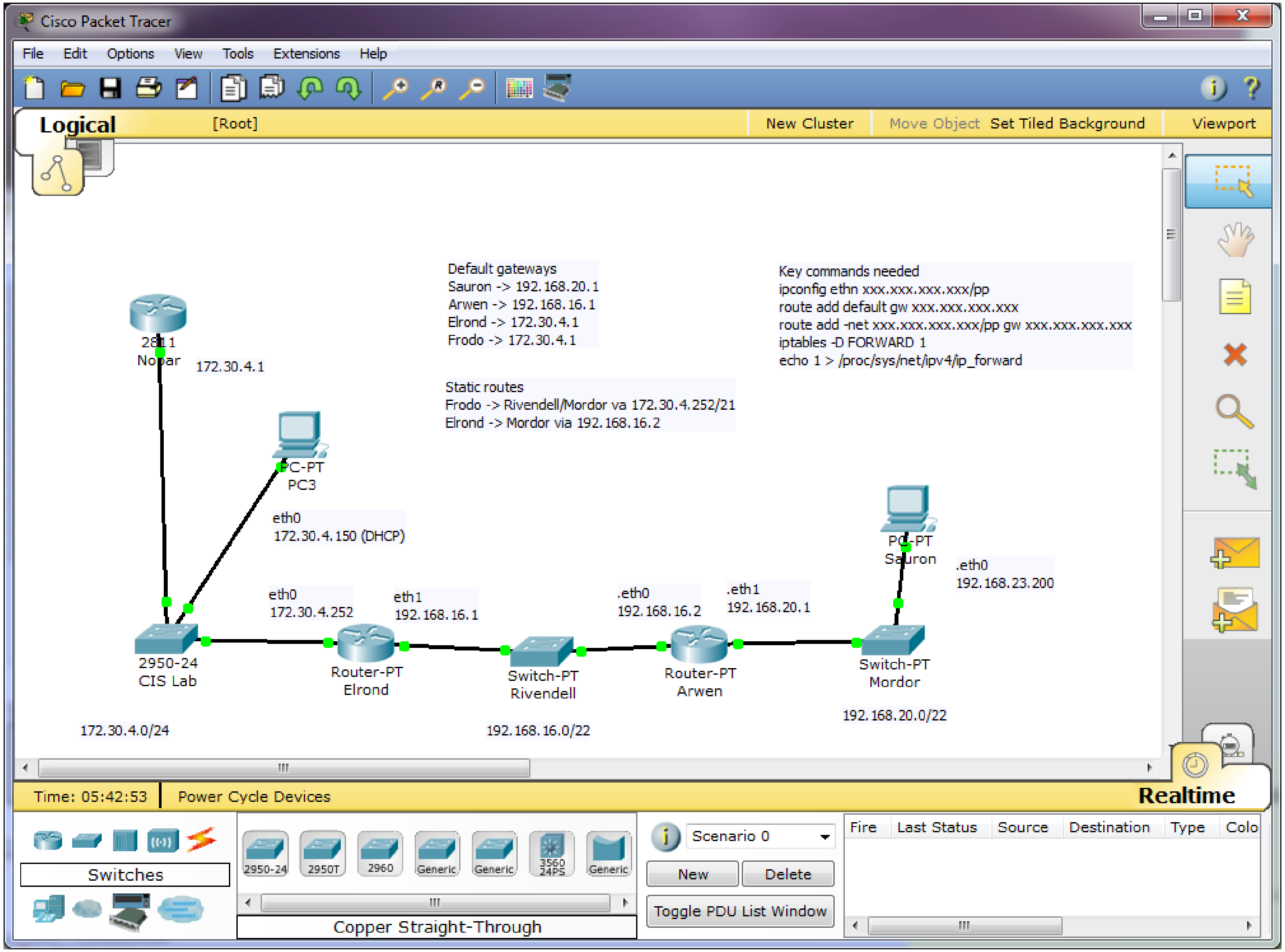
Sauron

ifconfig eth0 192.168.23.200/22
route add default gw 192.168.20.1

Some use pencil or whiteboard



Some use Visio



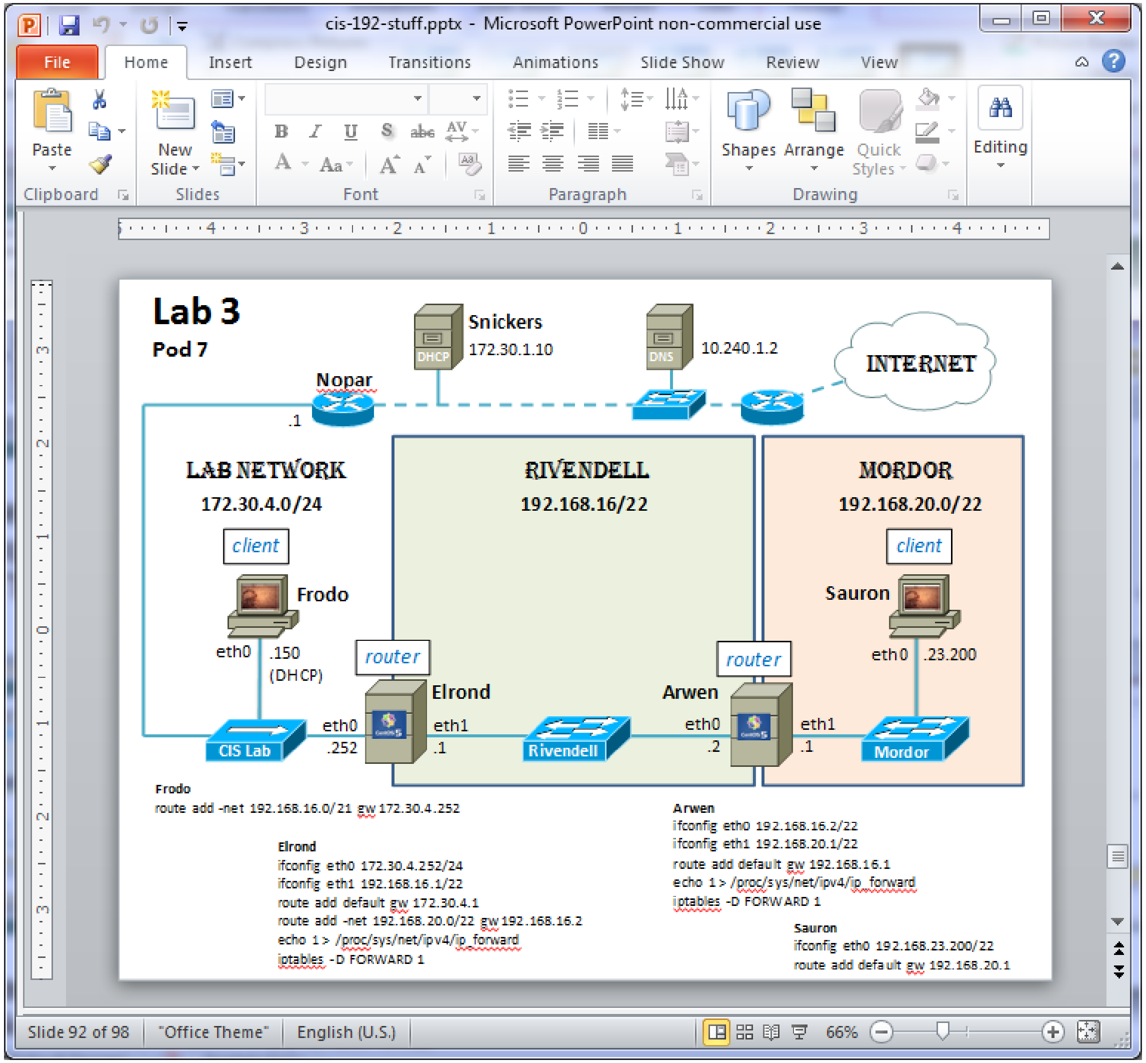
Default gateways
Sauron -> 192.168.20.1
Arwen -> 192.168.16.1
Elrond -> 172.30.4.1
Frodo -> 172.30.4.1

Static routes
Frodo -> Rivendell/Mordor va 172.30.4.252/21
Elrond -> Mordor via 192.168.16.2

Key commands needed

```
ipconfig ethn xxx.xxx.xxx.xxx/pp
route add default gw xxx.xxx.xxx.xxx
route add -net xxx.xxx.xxx.xxx/pp gw xxx.xxx.xxx.xxx
iptables -D FORWARD 1
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Some use Cisco Packet Tracer



Frodo

```
route add -net 192.168.16.0/21 gw 172.30.4.252
```

Elrond

```
ifconfig eth0 172.30.4.252/24
ifconfig eth1 192.168.16.1/22
route add default gw 172.30.4.1
route add -net 192.168.20.0/22 gw 192.168.16.2
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -D FORWARD 1
```

Arwen

```
ifconfig eth0 192.168.16.2/22
ifconfig eth1 192.168.20.1/22
route add default gw 192.168.16.1
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -D FORWARD 1
```

Sauron

```
ifconfig eth0 192.168.23.200/22
route add default gw 192.168.20.1
```

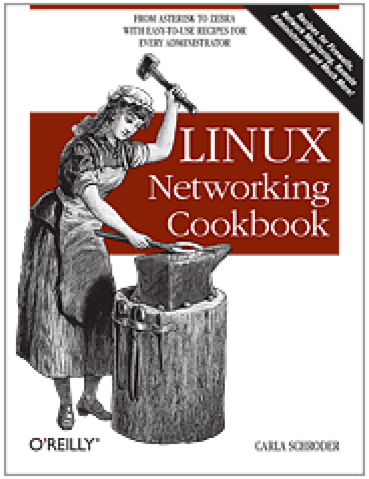
Some use PowerPoint

Network Topology Drawing Tools

Many more … use Google

Some use other solutions

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

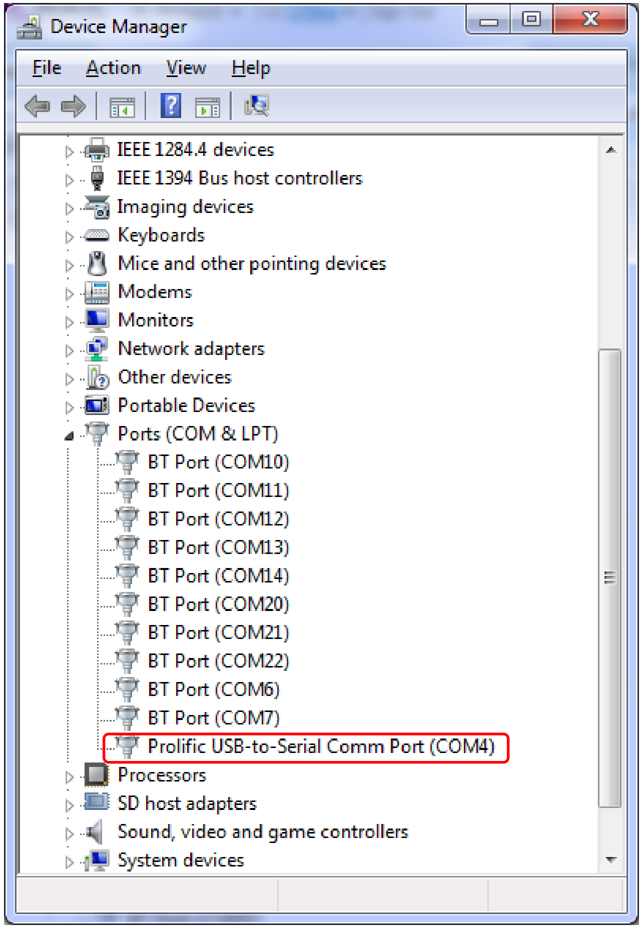
Assemble your own Linux based Router. This one has five Ethernet interfaces and uses 6.4 watts of power.

- *Eth1 is attached to the home LAN.*
- *Eth2 is attached to a 172.30.4.0/24 network.*
- *Eth3 is attached to a 172.30.1.0/24 network.*
- *The serial cable (console) can be attached to a laptop.*

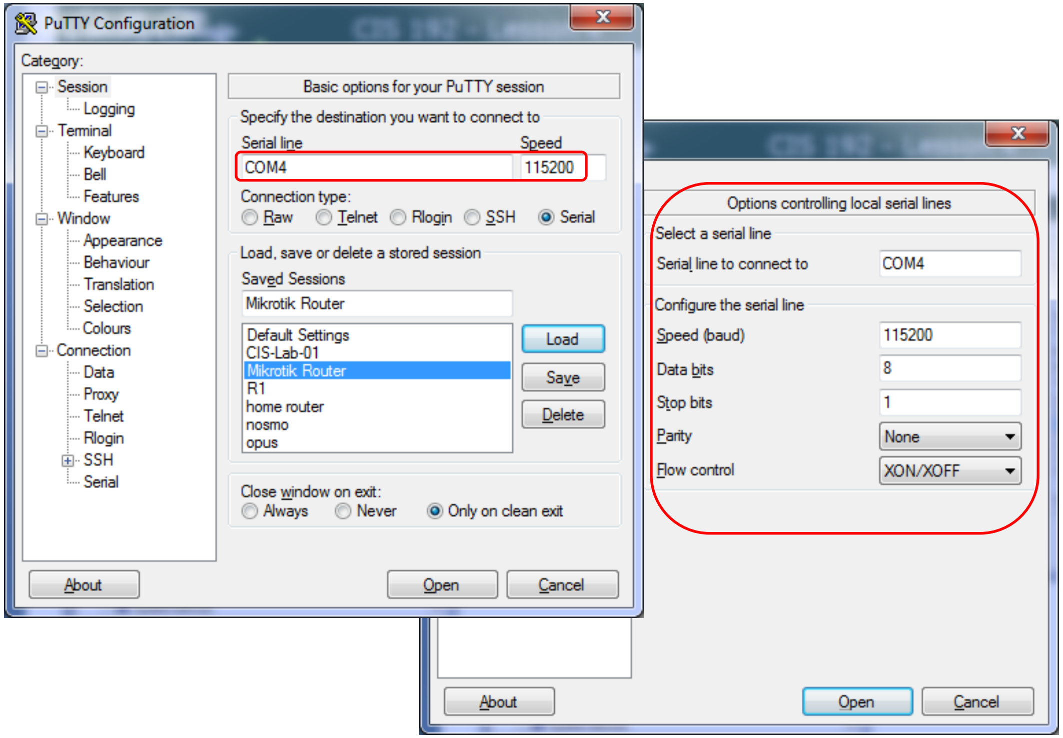
• RB/450 Routerboard	$69
• CA/150 indoor case	$19
• 24HPOW power supply	$18
• SW-1301 USB-to-serial adapter	$12

MikroTik/Routerboard – A Linux based router

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



- 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



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

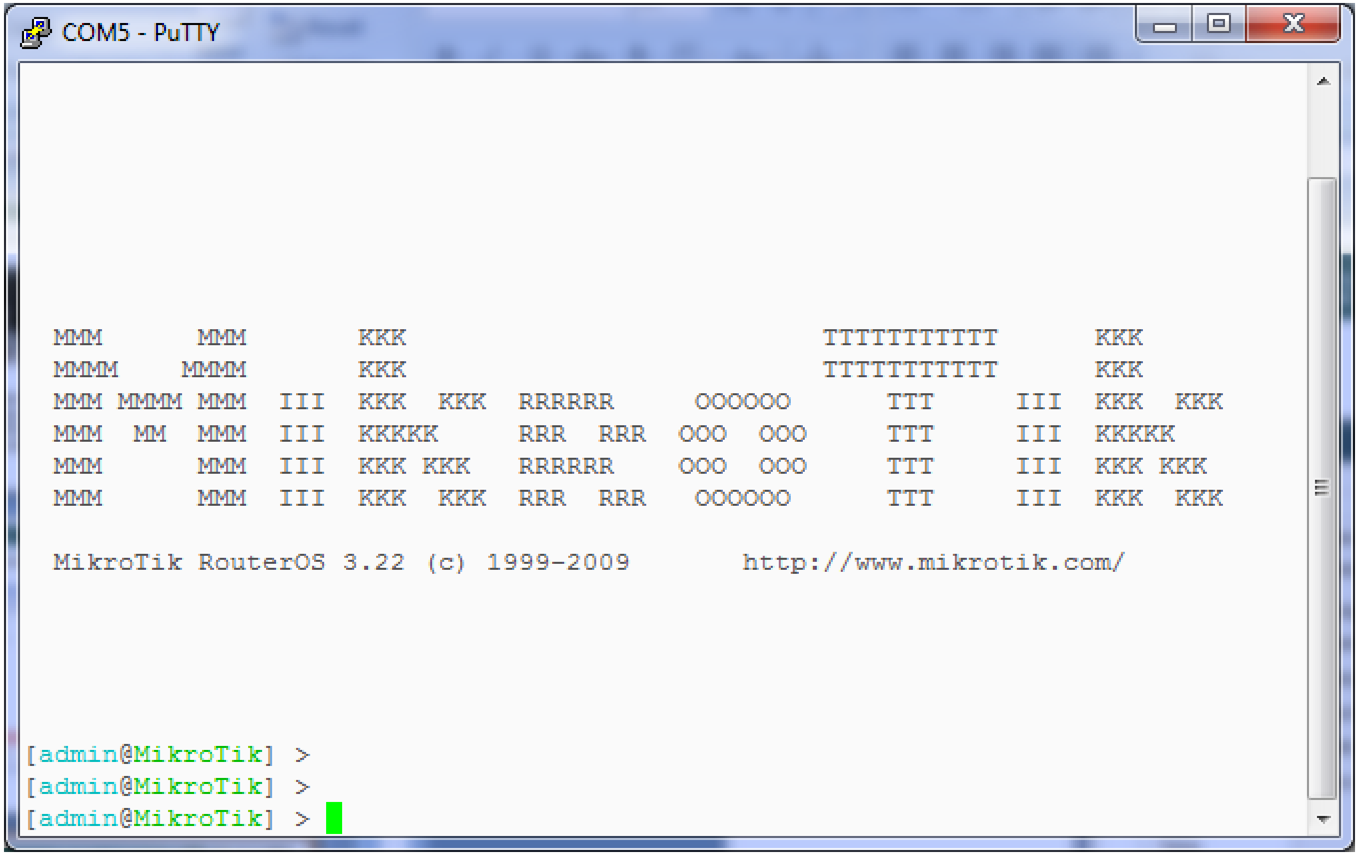
Select a serial line

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

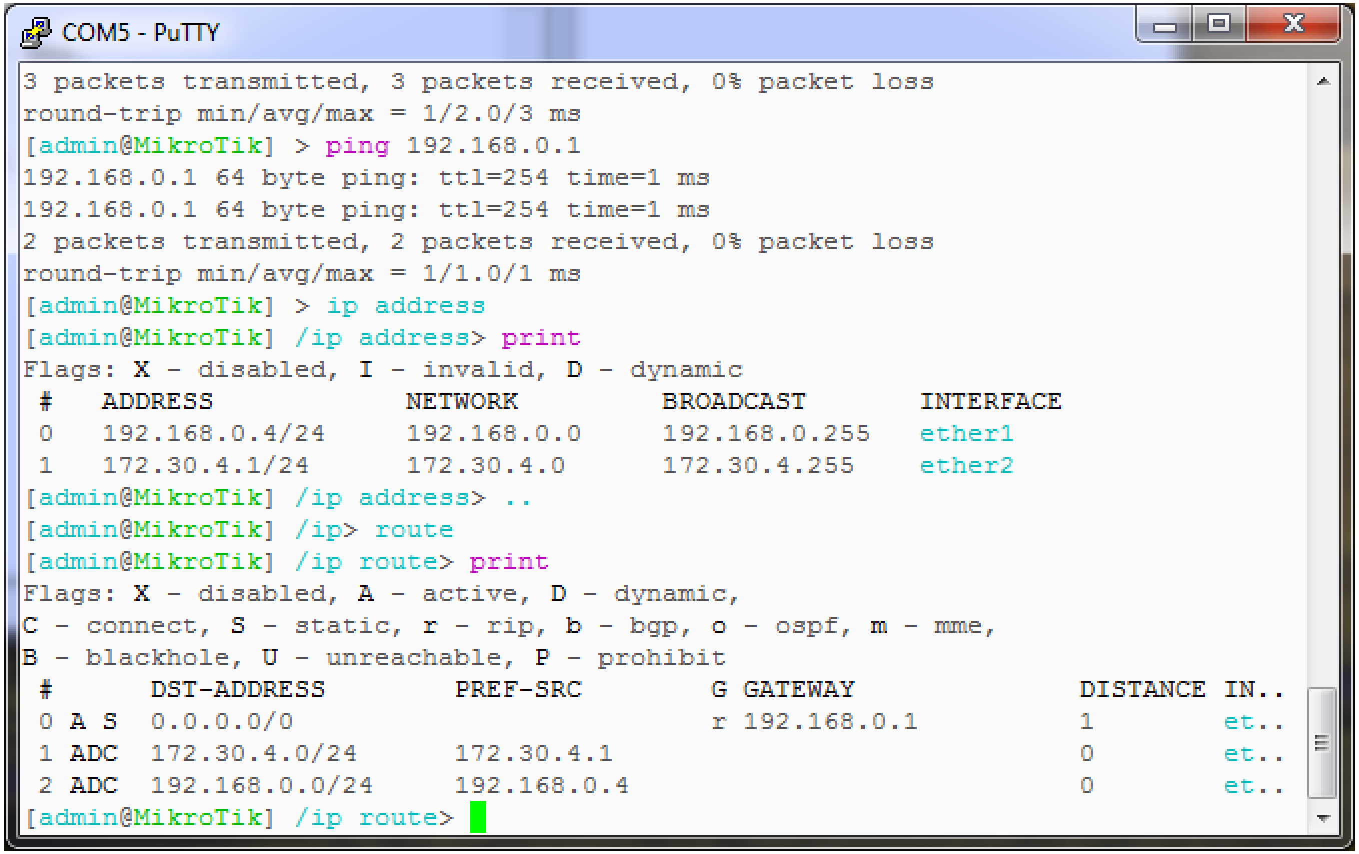
MikroTik/Routerboard – A Linux based router



```
  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.

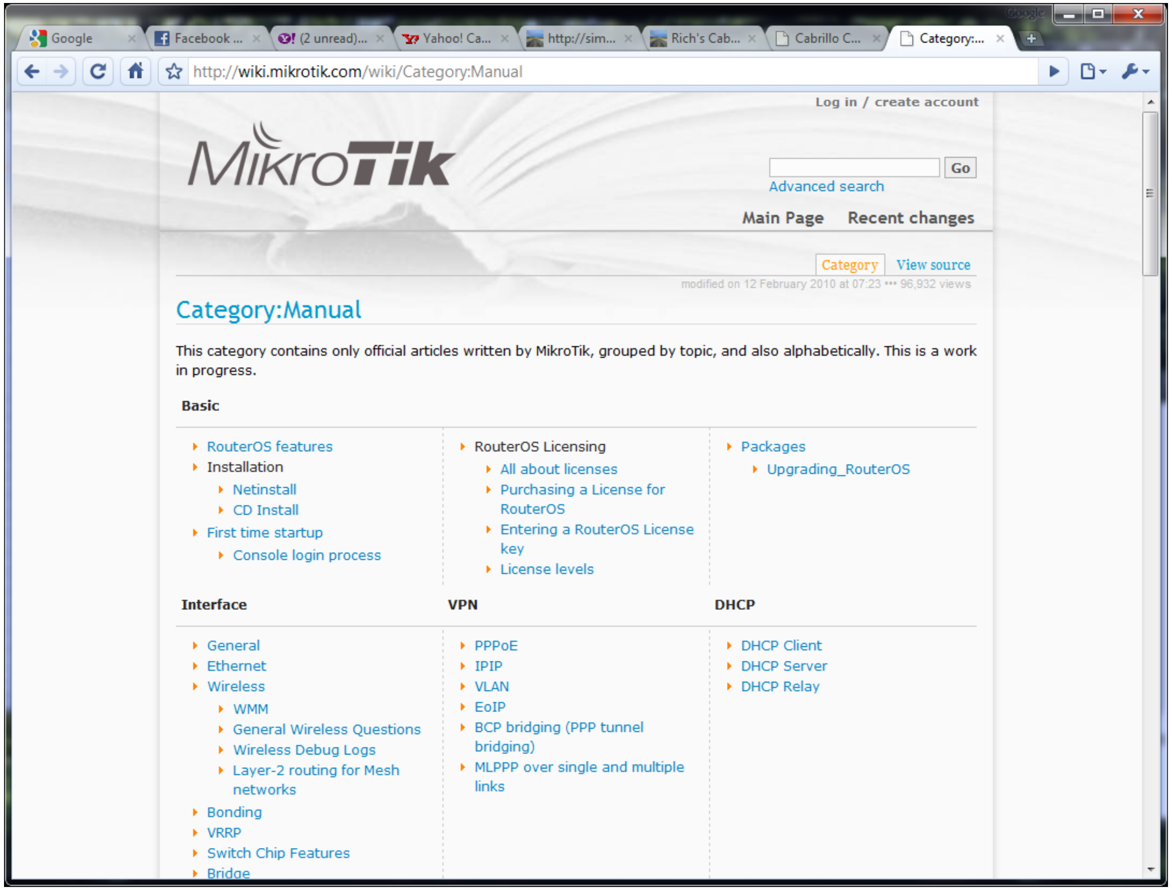
MikroTik/Routerboard – A Linux based router



```
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

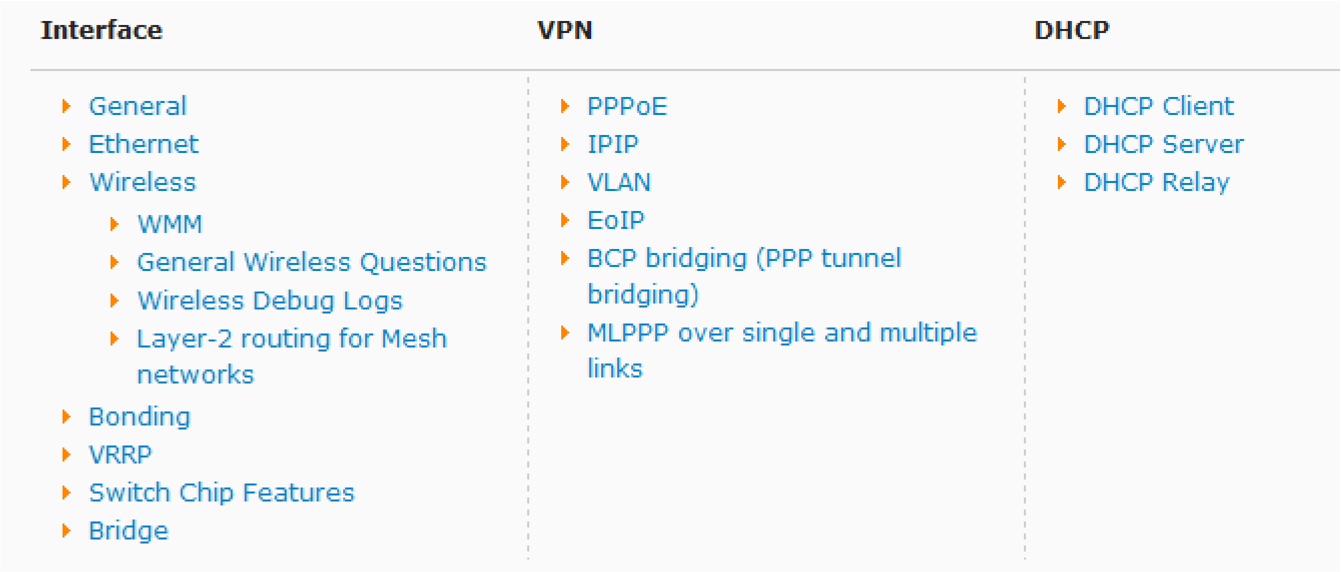
- PPPoE
- IPIP
- VLAN
- EoIP
- BCP bridging (PPP tunnel bridging)
- MLPPP over single and multiple links

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

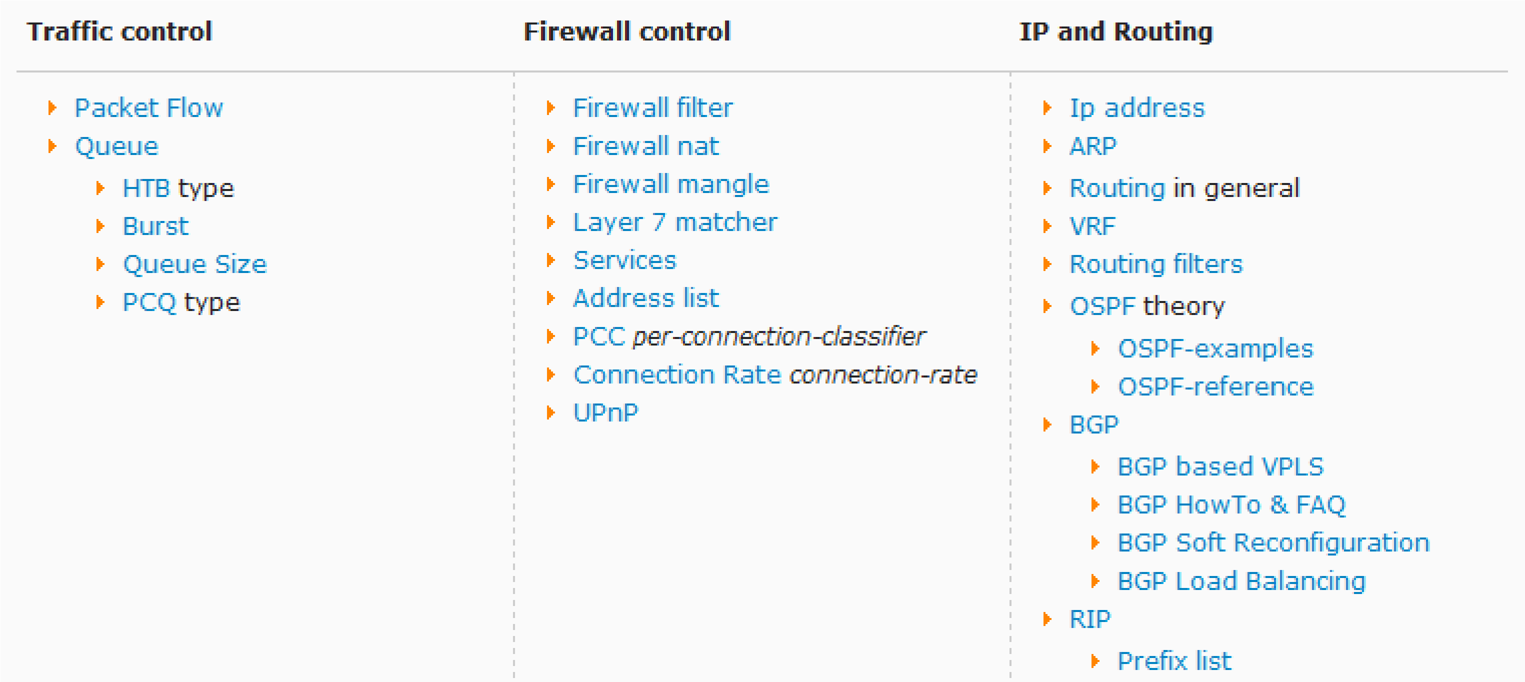
- PPPoE
- IPIP
- VLAN
- EoIP
- BCP bridging (PPP tunnel bridging)
- MLPPP over single and multiple links

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

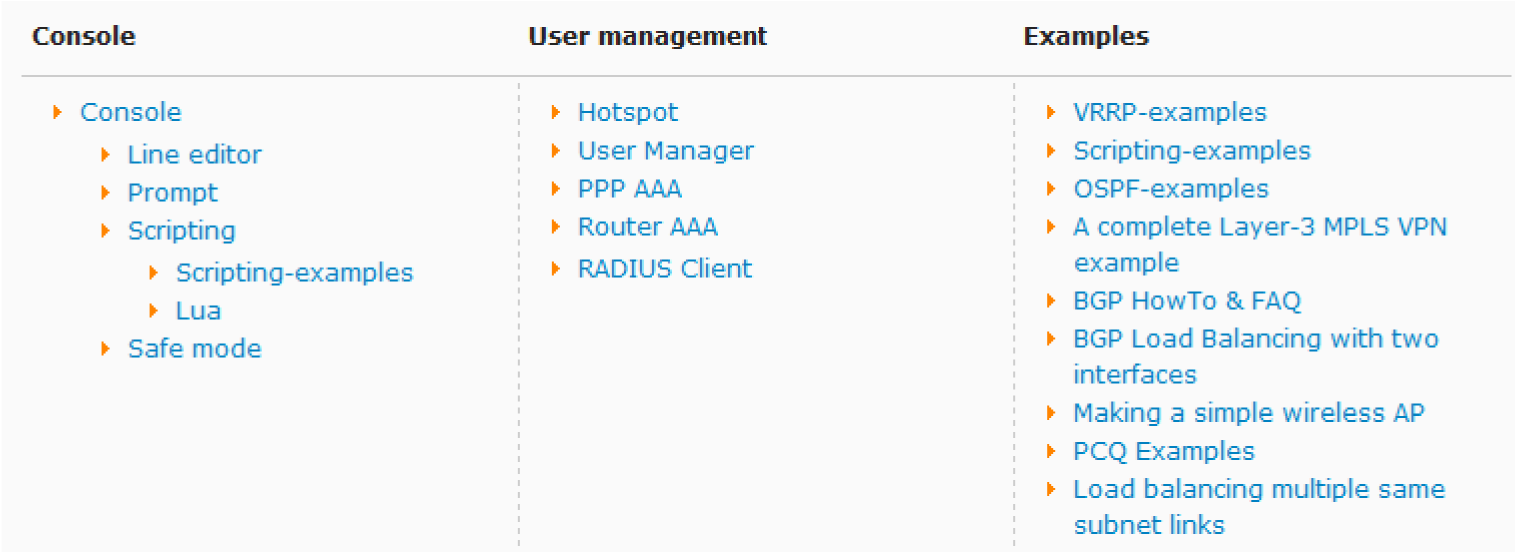
- Firewall filter
- Firewall nat
- Firewall mangle
- Layer 7 matcher
- Services
- Address list
- PCC *per-connection-classifier*
- Connection Rate *connection-rate*
- UPnP

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

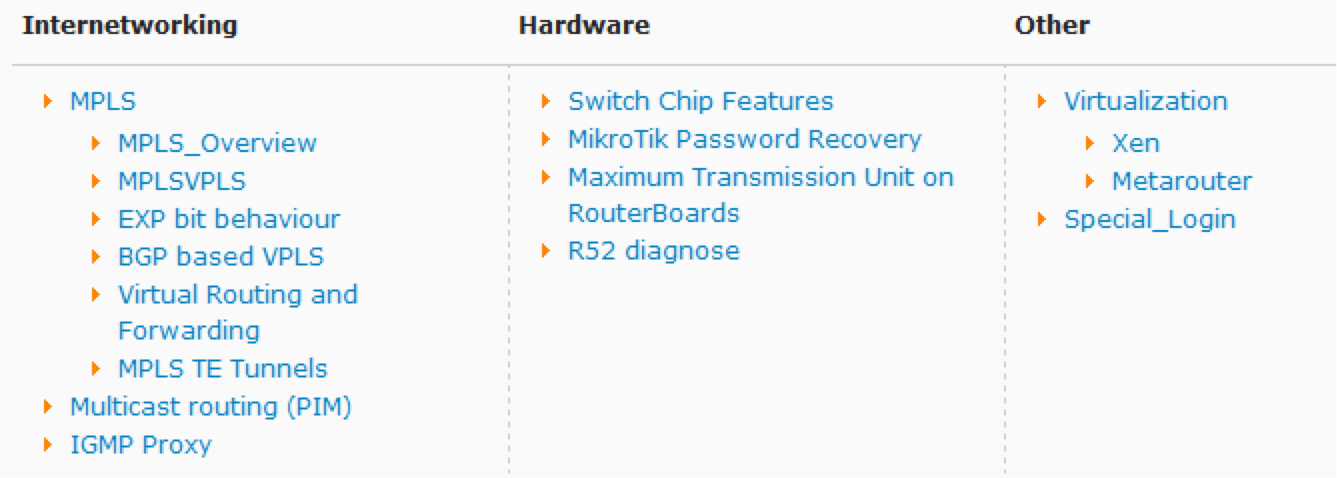
- Hotspot
- User Manager
- PPP AAA
- Router AAA
- RADIUS Client

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

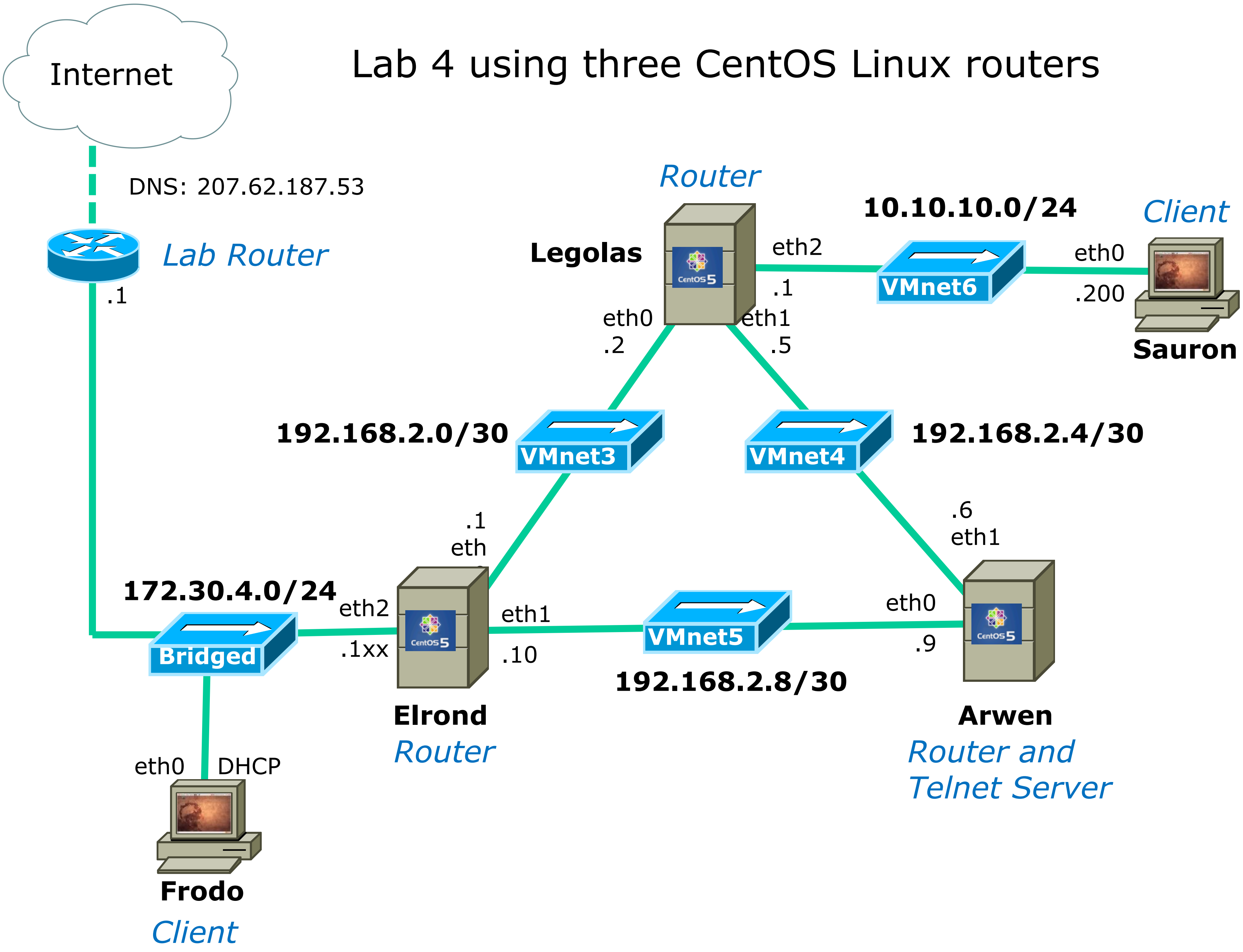
Other

- Virtualization
  - Xen
  - Metarouter
- Special_Login

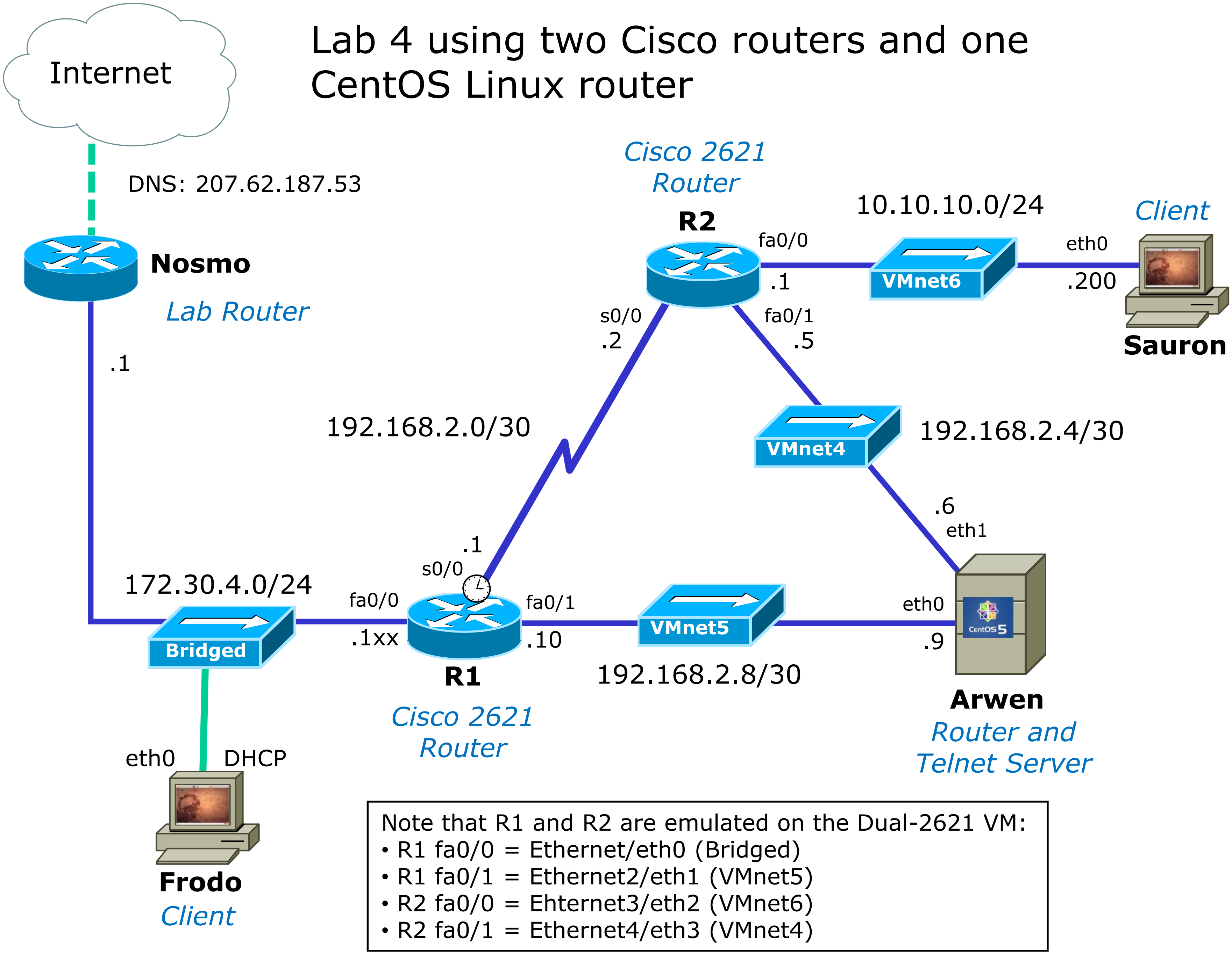
Online wiki documentation areas

Dynamips Dynagen

Lab 4 using three CentOS Linux routers



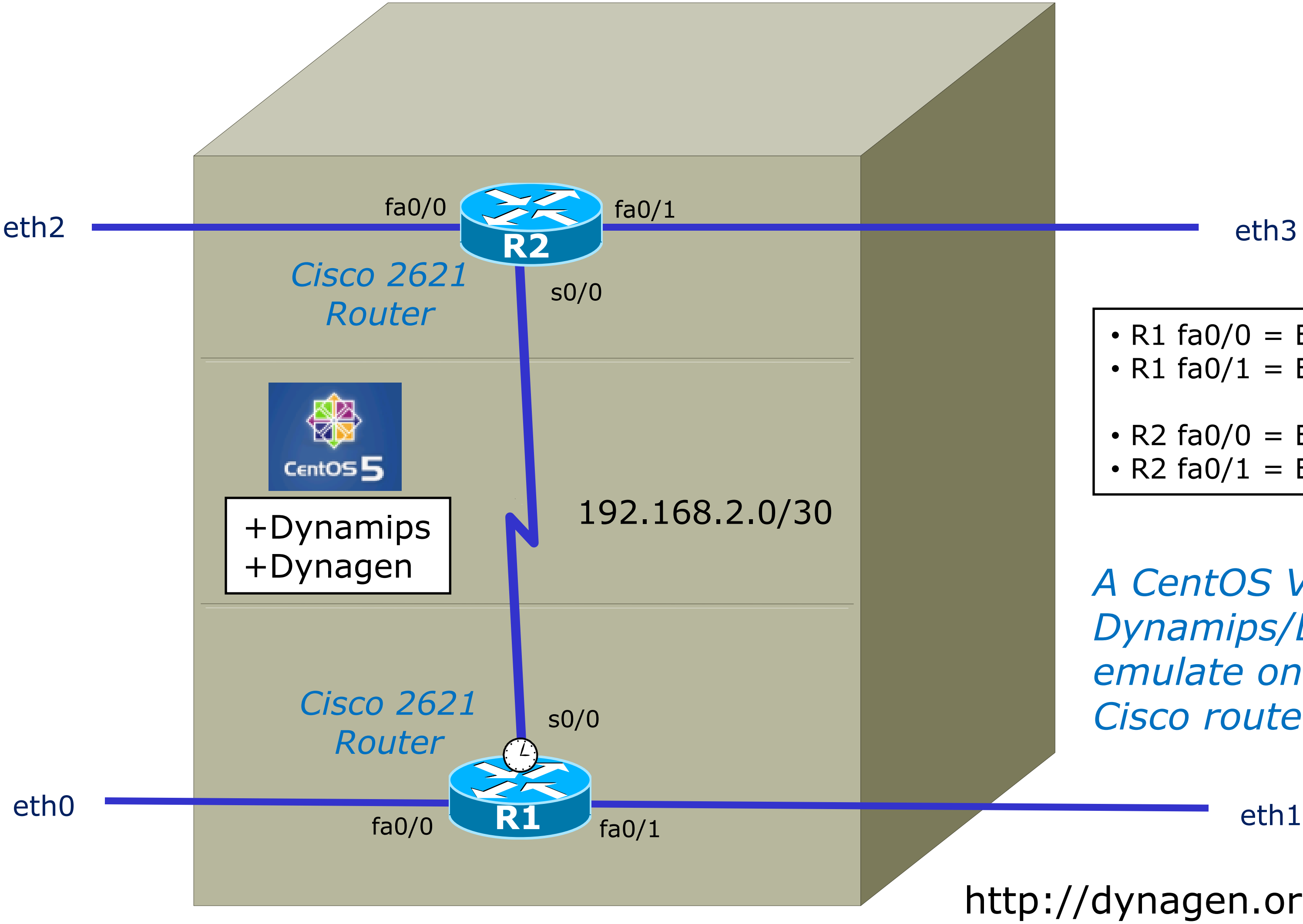
Lab 4 using two Cisco routers and one CentOS Linux router



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



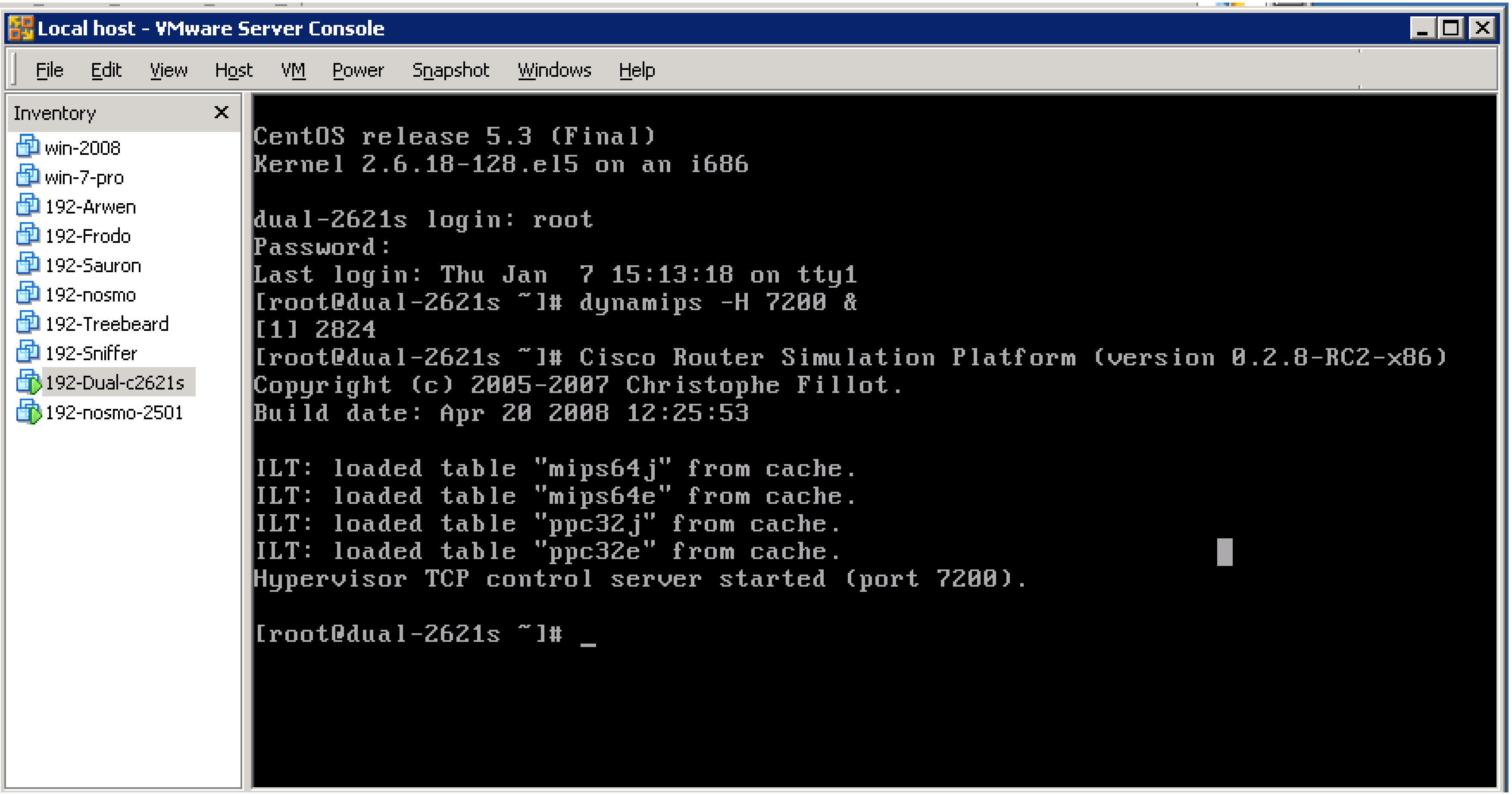
- R1 fa0/0 = Ethernet/eth0
- R1 fa0/1 = Ethernet2/eth1

- 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

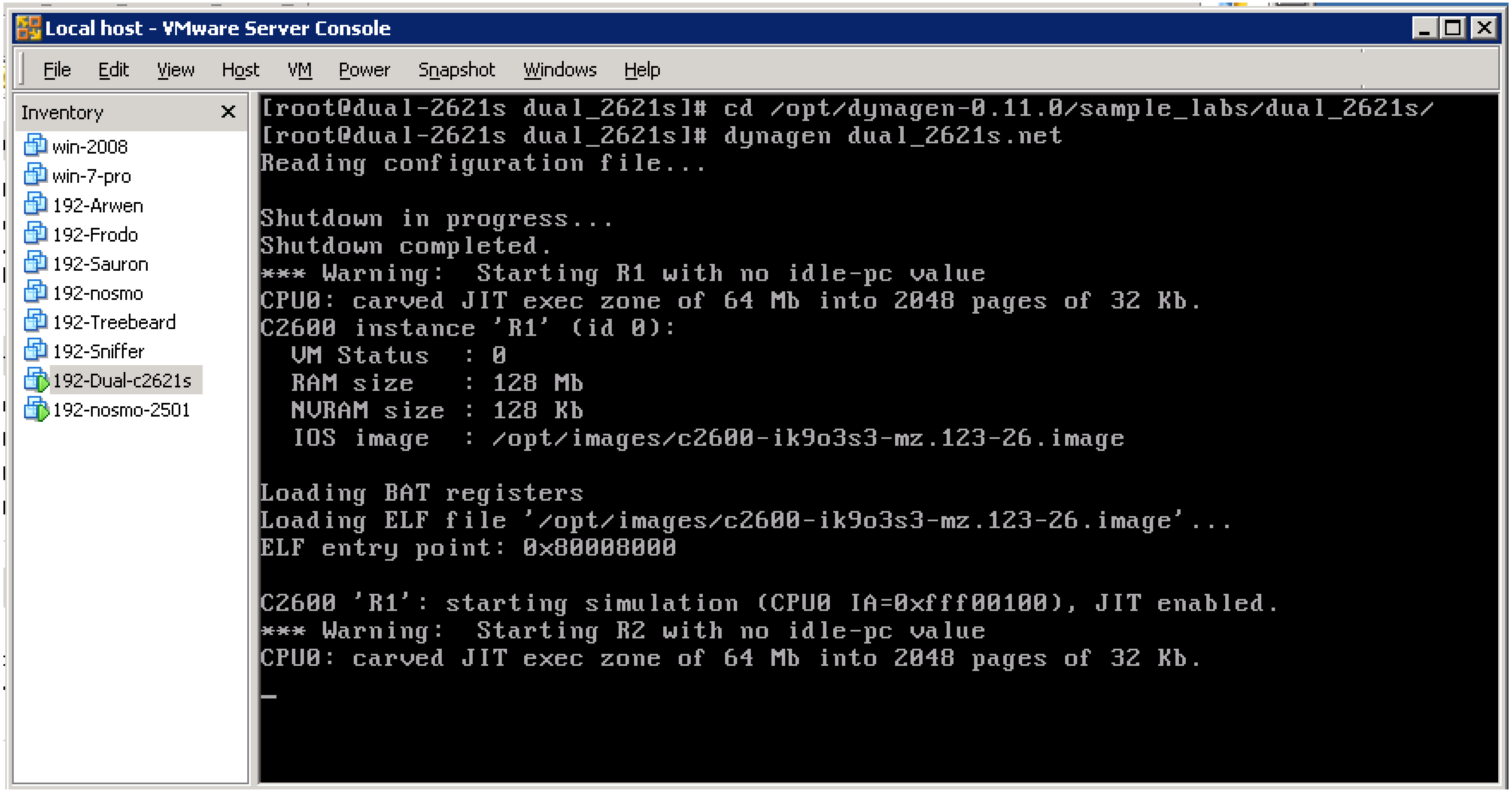
The Dual-c2621s VM



```
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*

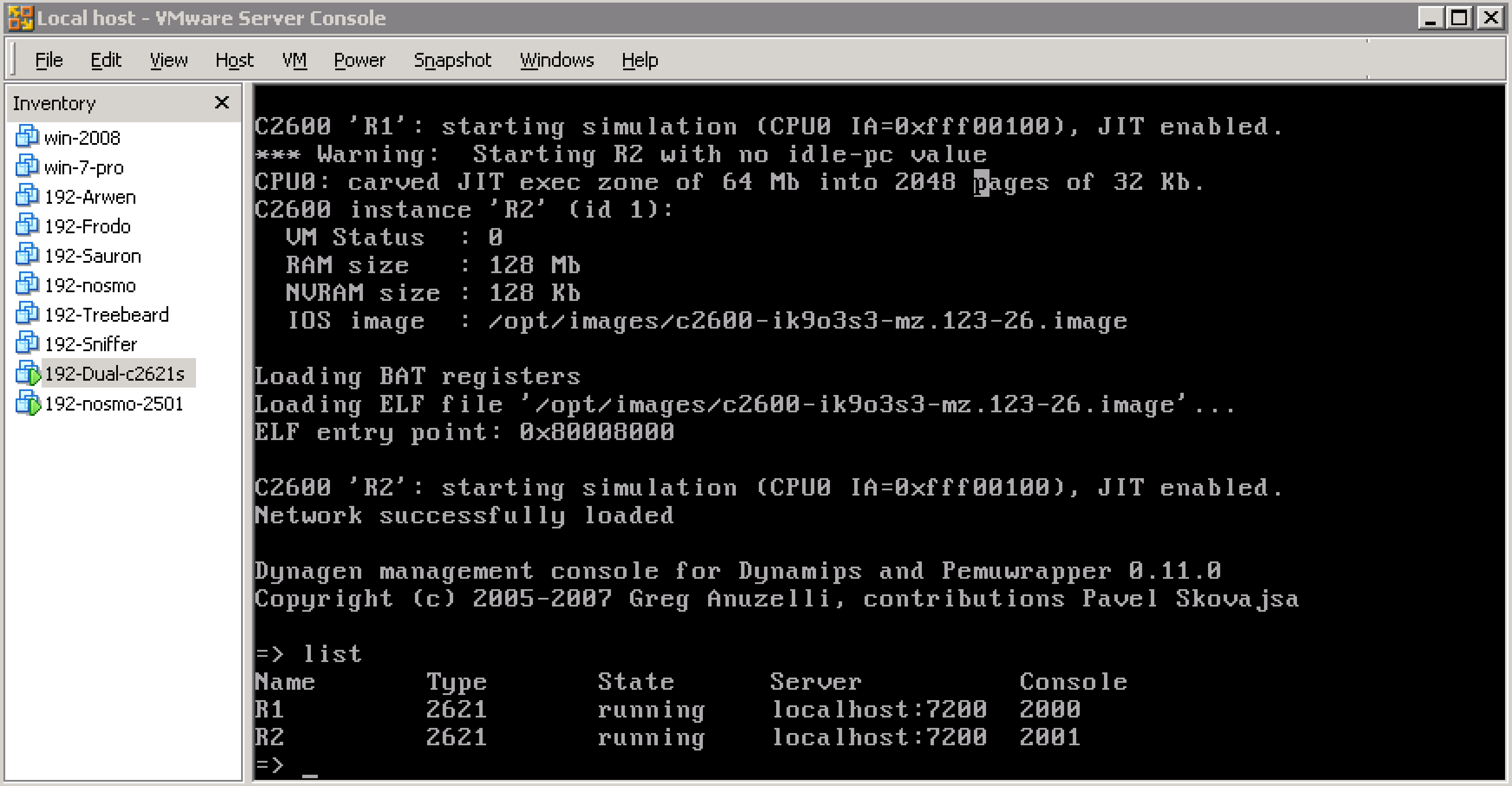
The Dual-c2621s VM



```
[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*

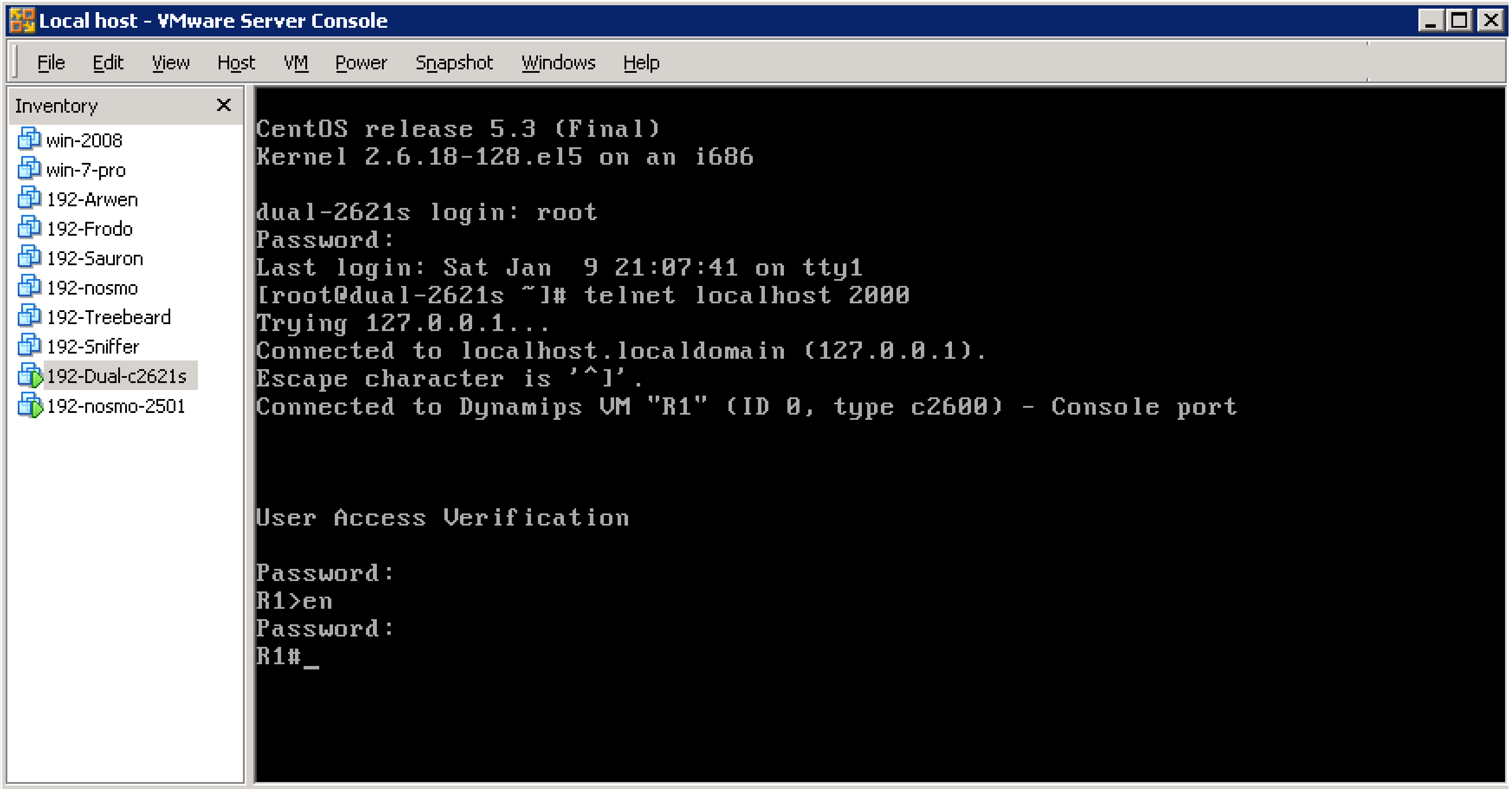
The Dual-c2621s VM



```
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*

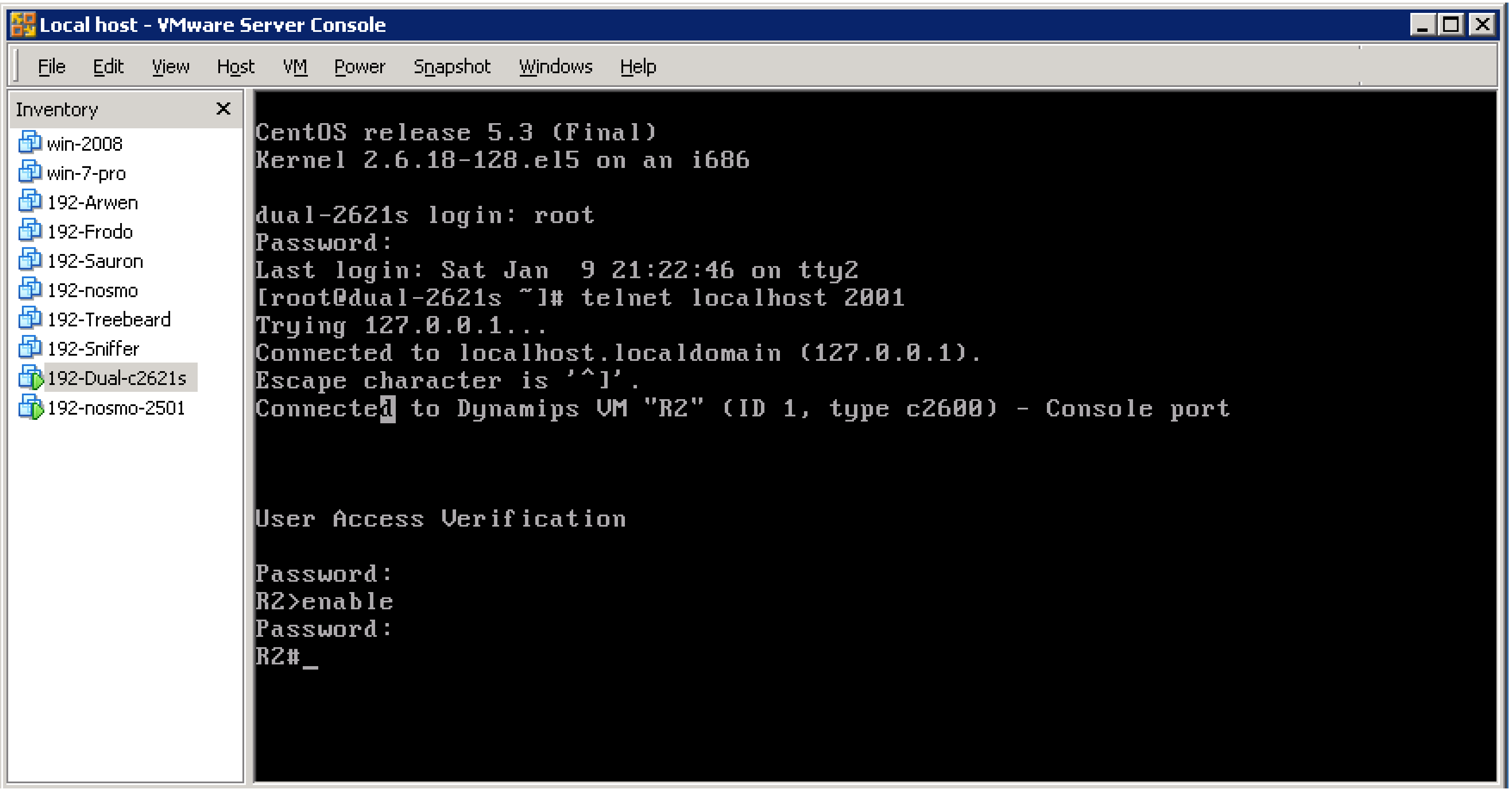
The Dual-c2621s VM



```
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)*

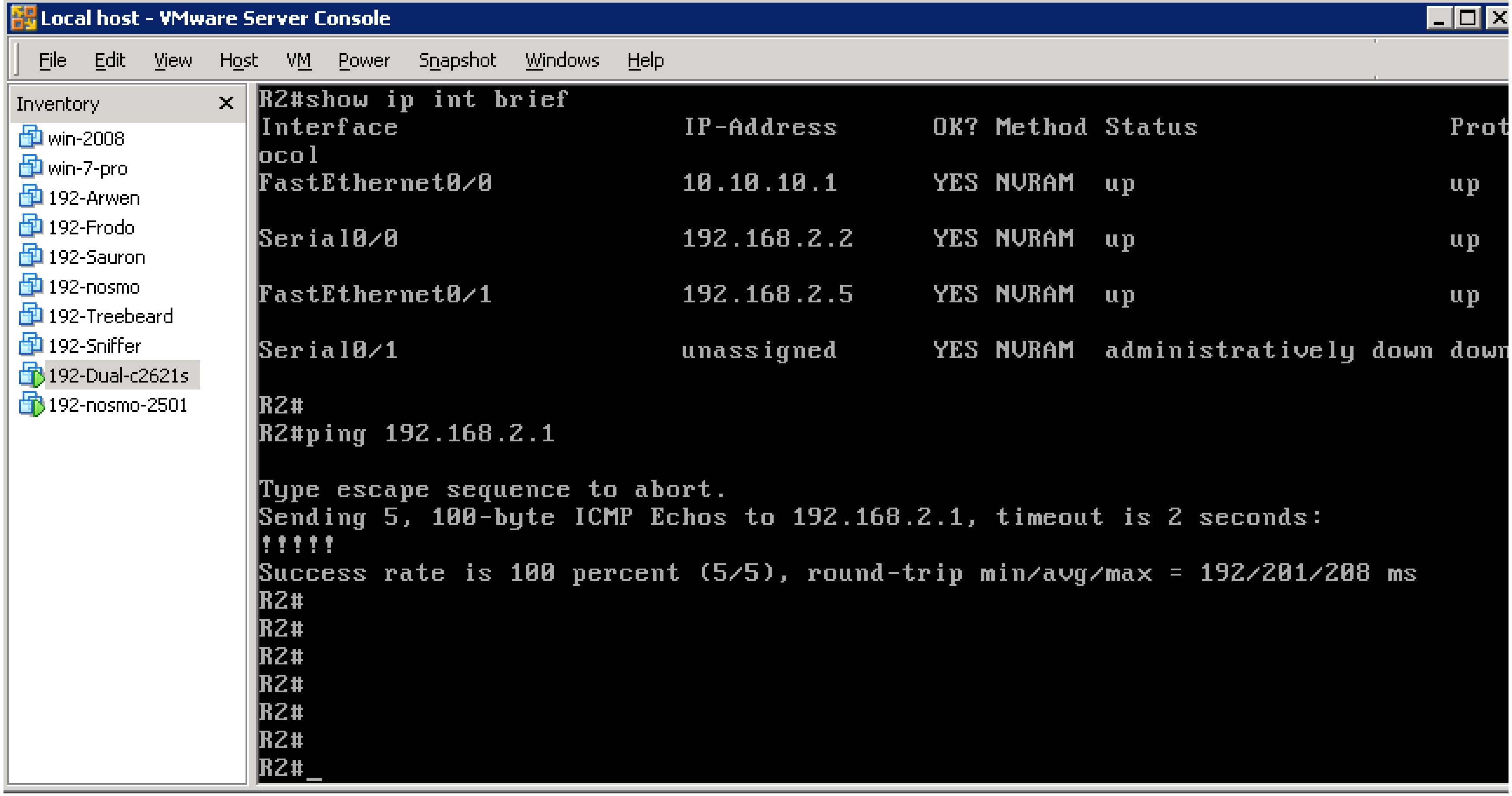
The Dual-c2621s VM



```
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!

Routing Review

Routing Summary

sign post

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

routing table

- Routers operate at **layer 3** and make decisions on where to send a packet.

- Routers use the **routing table** to decide where to forward a packet.

- If there is no route for a packet’s destination, the packet is dropped

Reading the routing table

Routing Table

-n shows IP addresses instead of names (faster)

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

Reading and understanding routing tables is absolutely ***critical***

The Routing Algorithm

(How the decision is made)

Routing Algorithm

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

1. Compute the route destination network address for the destination IP address

   *by applying the **genmask** in the routing table to the destination IP address in the packet*

2. Does the destination network match any routes to a directly attached network?

   *If so, packet has arrived, send it out the **iface** (interface) for that network*

3. Does the destination network match one or more non-directly attached network routes listed in the routing table?

   *If so, packet has not arrived at its destination, send it along via the next hop **gateway** using the appropriate **iface** (interface). If more than one route matches, select the best match (largest **genmask**).*

4. Is there a default route listed in the routing table?

   *If so, use that **gateway***
   Otherwise, drop the packet - "network is unreachable"

The Routing Algorithm

Compute the route destination network address

The destination network is obtained by applying the genmask to the IP destination address in the packet.

Example: Destination IP=192.168.3.200 and genmask=255.255.255.0

- By hand

```
        128 64 32 16  8 4 2 1
  110000 10101000 00000011 11001000  192.168.3.200
  111111 11111111 11111111 00000000  255.255.255.0
  110000 10101000 00000011 00000000  192.168.3.0
```

- With ipcalc

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

*The computed **destination** network address is 192.168.3.0*

Reading the routing table

Routing Table

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

*The router will use a **routing table** like this to decide where or whether to forward a packet.*

The router will take the packet's destination IP address and look for the best route.

The best route has the most genmask network bits that match.

Reading the routing table

Routing Table

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

*These routes are **directly connected networks**. No **gateway,** aka **next hop router**, is necessary to get to those networks.*

Reading the routing table

Routing Table

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

*These routes are **NOT directly connected networks**. Packets must travel via a **gateway,** aka **next hop router,** to get to the destination network.*

Reading the routing table

Routing Table

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

Translation:

- *Packets going to a destination host network of **172.30.4.0/24** have arrived!*
- *Proceed out the door labeled **eth0** and locate the destination host on that directly attached network.*

Reading the routing table

Routing Table

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

Translation:

- *Packets going to a destination host network of* ***192.168.3.0/24*** *have* ***NOT*** *arrived!*
- *Proceed out the door labeled* ***eth1****, locate the next hop router at* ***192.168.2.123*** *and ask for more routing directions there.*

Reading the routing table

Routing Table

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

Translation:

- *Packets going to a destination host network of **192.168.2.0/24** have arrived!*
- *Proceed out the door labeled **eth1** and locate the destination host on that directly attached network.*

Reading the routing table

Routing Table

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

Translation:

- *Packets going to a destination host network of **169.254.0.0/16** have arrived!*
- *Proceed out the door labeled **eth1** and locate the destination host on that directly attached network.*

Reading the routing table

Routing Table

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

Translation:

- *Packets going to a destination for **any other networks** have **NOT** arrived!*
- *Proceed out the door labeled **eth0**, locate the next hop router at **172.30.4.1** and ask for more routing directions there.*

Configuring the Routing Table

- Directly connected networks are automatically added to the routing table.
- APIPA routes are automatically added to the routing table.
- Default gateways can be **manually** added using the route command or added to a configuration file used by the network service. *(Lab 3)*
- Static routes can be **manually** added using the route command or added to a configuration file used by the network service. *(Lab 3)*
- Dynamic routing services that use routing protocols like RIP and OSPF can add **automatically** add routes to the routing table. *(Lab 4)*

The Routing Table
Supernetting

Routing Table

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

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

route command –n option

show route table with names

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

show route table with IP addresses

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

route command for viewing cache

show route table cache with names

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

route command for viewing cache

show route table cache with IP addresses

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

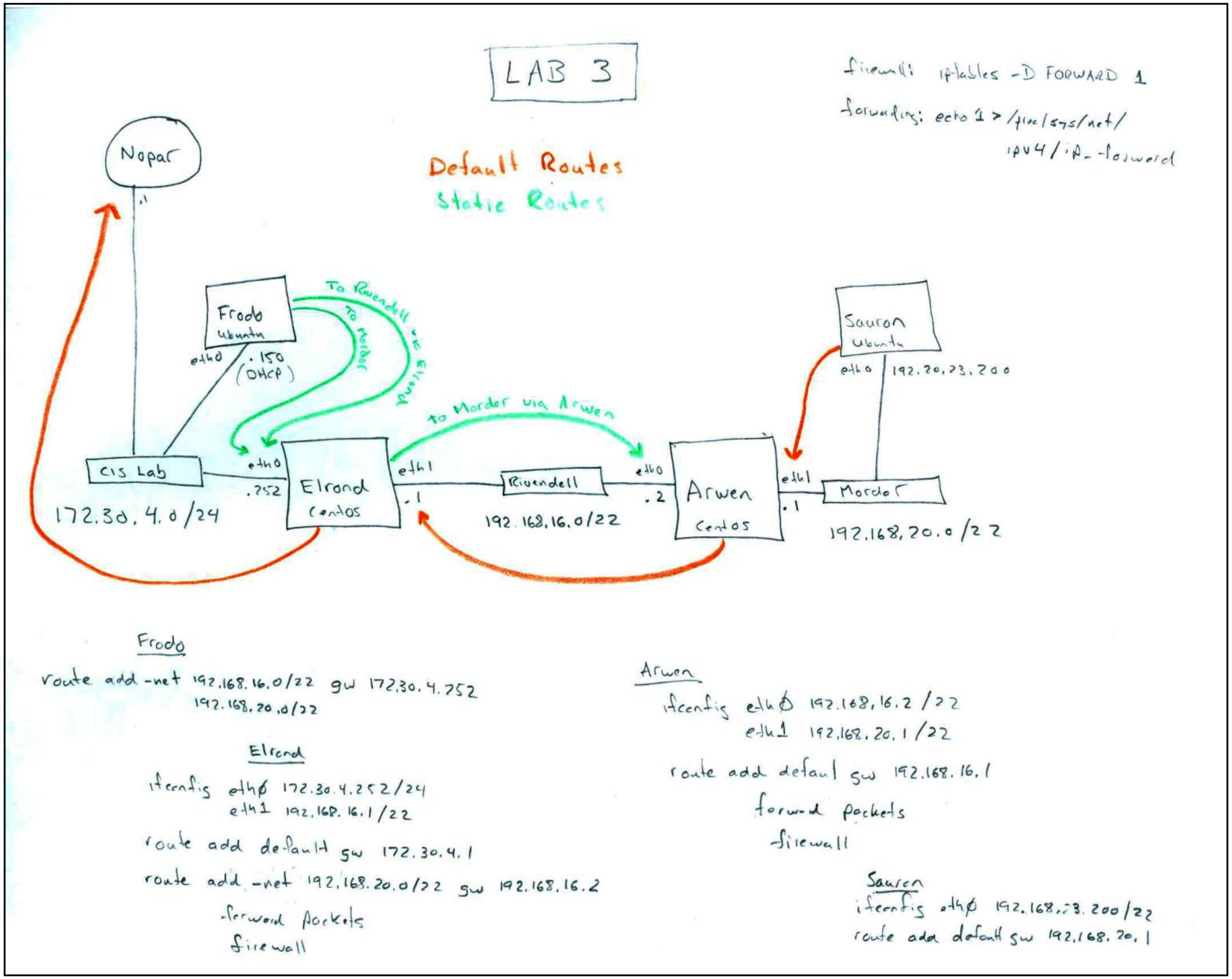
route command

flushing the cache

Flush the route cache

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

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



```
firewall: iptables -D FORWARD 1
forwarding: echo 1 > /proc/sys/net/
                         ipv4/ip_forward
```

Frodo

```
route add -net 192.168.16.0/22 gw 172.30.4.252
               192.168.20.0/22
```

Elrond

```
ifconfig eth0 172.30.4.252/24
         eth1 192.168.16.1/22
route add default gw 172.30.4.1
route add -net 192.168.20.0/22 gw 192.168.16.2
    forward packets
    firewall
```

Arwen

```
ifconfig eth0 192.168.16.2/22
         eth1 192.168.20.1/22
route add defaul gw 192.168.16.1
    forward packets
    firewall
```

Sauron

```
ifconfig eth0 192.168.23.200/22
route add default gw 192.168.20.1
```

In Lab 3 we manually added three default routes to Elrond, Arwen and Sauron.

Three static routes to Mordor and Rivendell were added to Frodo and Elrond.

If this was a larger and more complex network manually adding routes would get very ***tedious*** *and* ***problematic****!*

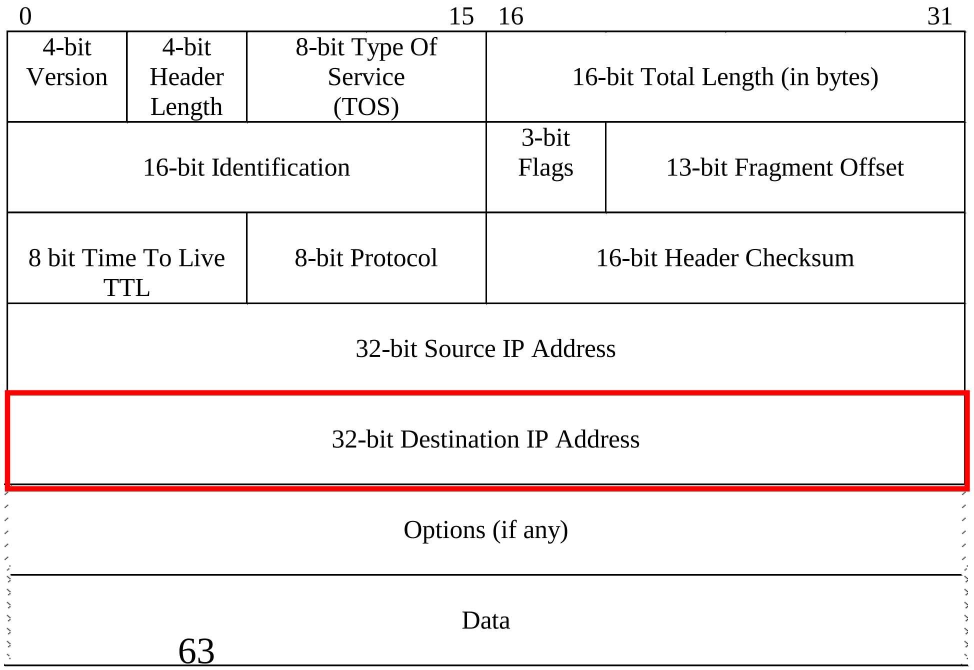
Dynamic Routing Protocols

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.

Note that the subnet mask does not travel with the packet.

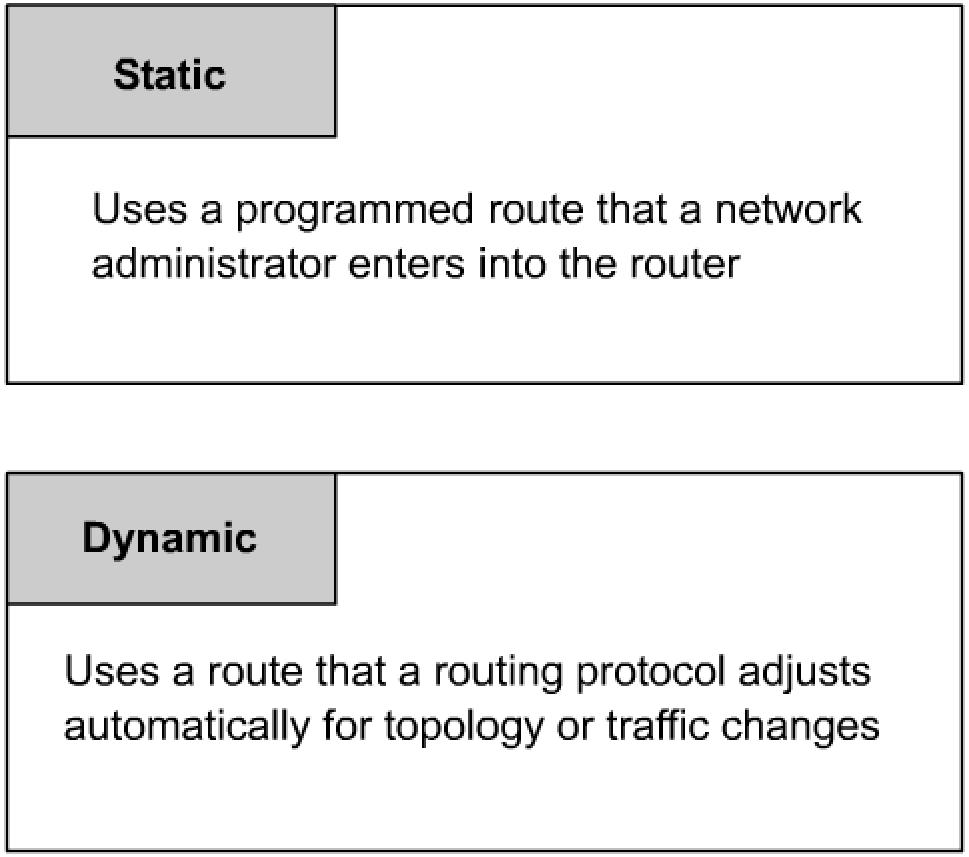
IP Header



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

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 CIS Lab 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 CIS Lab 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

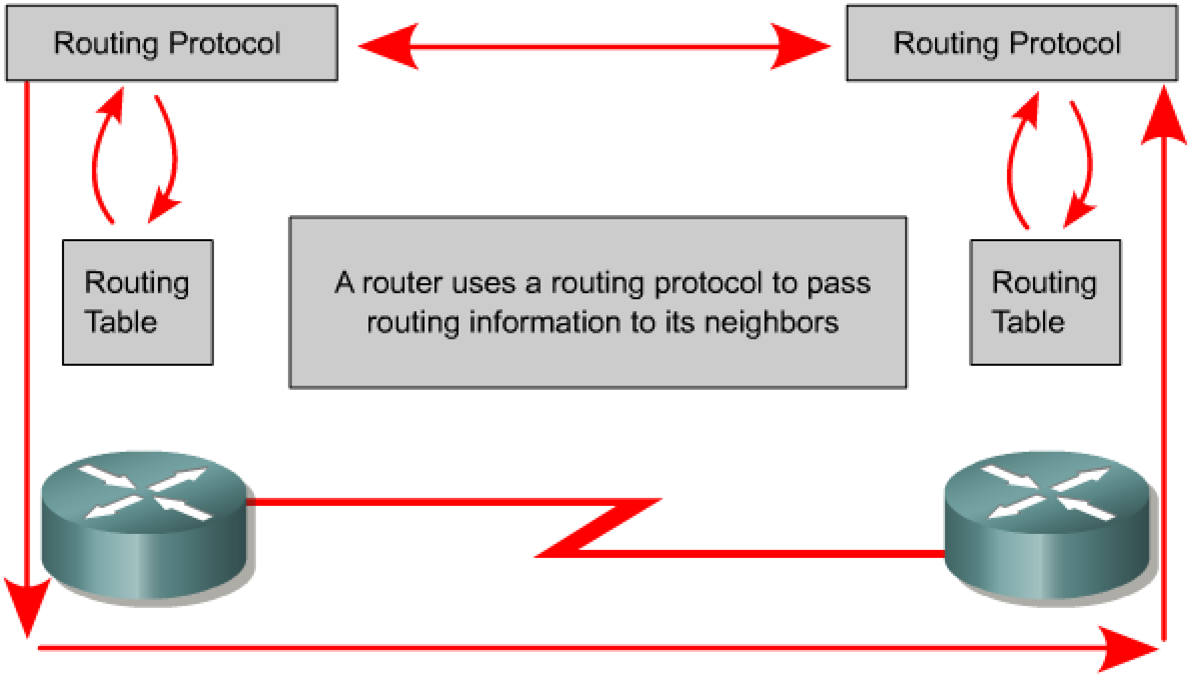
After doing lab 3 can you imagine ***manually*** *setting up and maintaining static routes on dozens or evens hundreds of routers!*

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

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)

Software Installation Tip for Labs

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

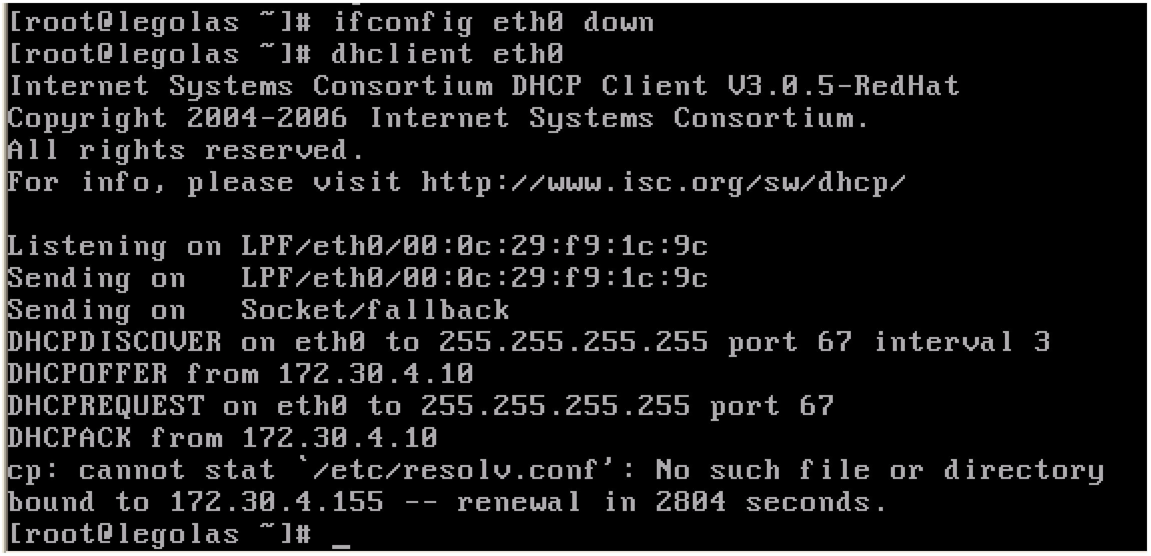
Just cable it temporarily to the CIS Lab network and use dhclient to get an IP address

1. Use **ifconfig eth0 down**
2. Re-cable eth0 from VMnet3 to Bridged/CIS Lab network.
3. Use **dhclient eth0** to join the CIS Lab network[1].
4. Use **yum install** *whatever*
5. Use **dhclient –r** to release DHCP address.
6. Use **ifconfig eth0 down**
7. Re-cable eth0 from Bridged back to the previous network.
8. 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

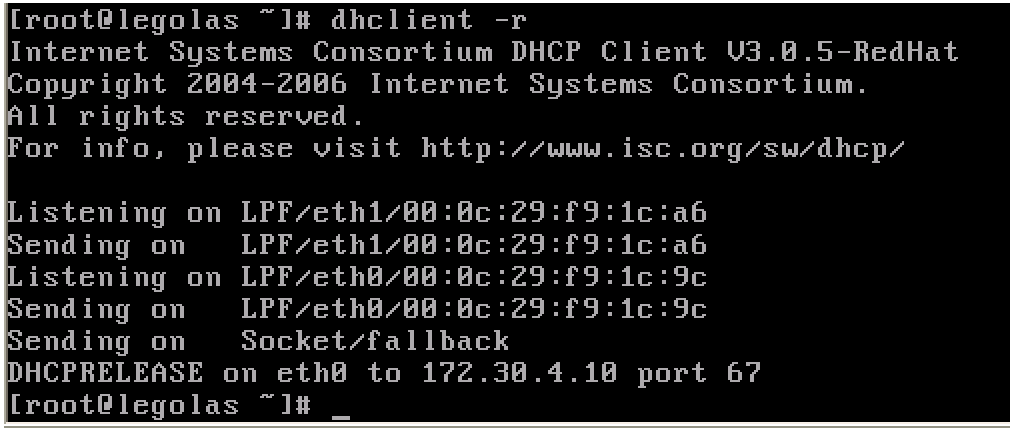
- *Bringing down the currently configured interface*
- *Re-cable the interface to the CIS Lab network*
- *Using DHCP to get an IP address*



```
[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

- *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*

10 Steps for installing Network Service

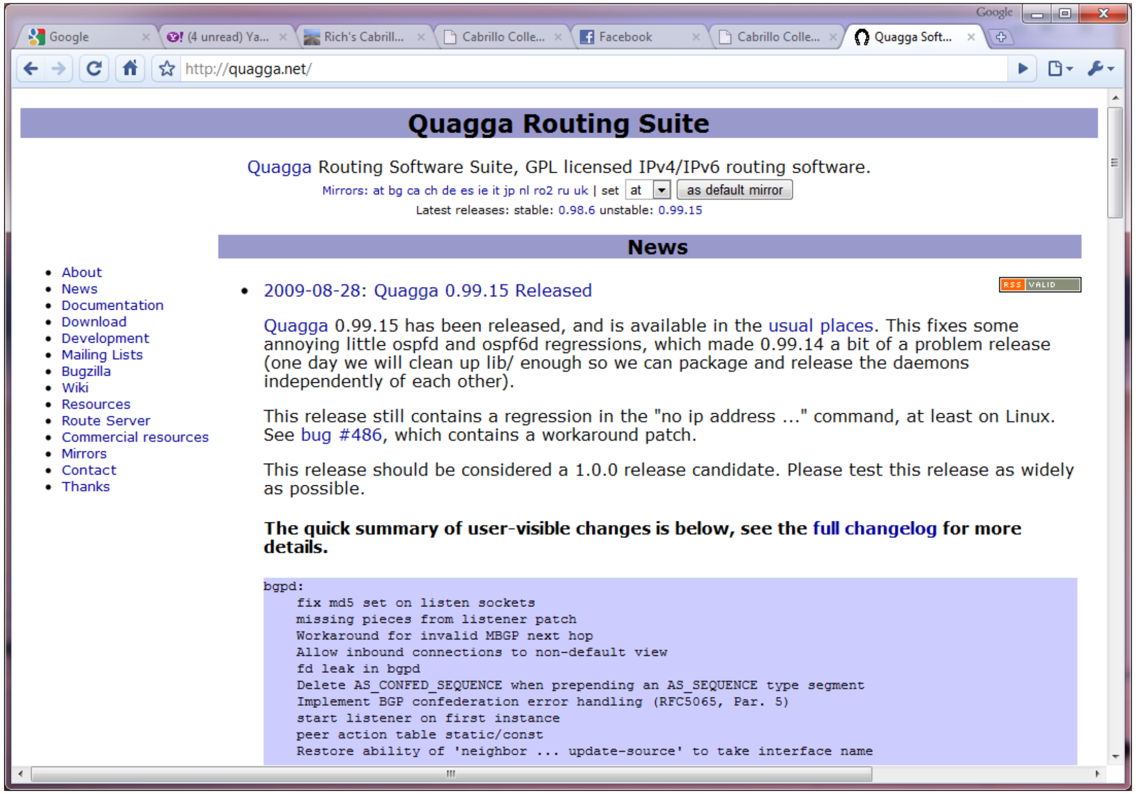
Service Applications

Steps to installing network services

1. Install software package using **yum**, **rpm** or build from source code
2. Customize service's configuration file
3. Modify the firewall to allow access to the service
4. Customize SELinux context settings to allow use
5. Start the service
6. Configure service to automatically start when system boots
7. Monitor and verify service is running
8. Troubleshoot as necessary
9. Monitor log files as appropriate
10. Configure additional security

Installing Quagga

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

- 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.

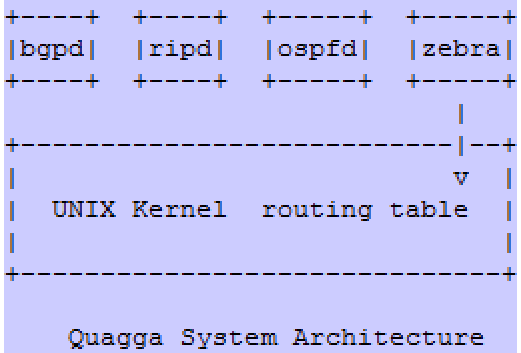
This release should be considered a 1.0.0 release candidate. Please test this release as widely as possible.

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!

Installing Quagga

Step 1 *Install software*

```
[root@celebrian ~]# rpm -qa | grep quagga
[root@celebrian ~]#
```

The server package "quagga" has not yet been installed.

Installing Quagga

Step 1 *Install software with yum*

```
[root@celebrian ~]# yum install quagga
Loaded plugins: fastestmirror
Determining fastest mirrors
 * base: mirrors.versaweb.com
 * extras: mirrors.usc.edu
 * updates: ftp.osuosl.org
base                                                     | 3.7 kB     00:00
extras                                                   | 3.0 kB     00:00
updates                                                  | 3.5 kB     00:00
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package quagga.i686 0:0.99.15-5.el6_0.2 set to be updated
--> Processing Dependency: libnetsnmp.so.20 for package: quagga-0.99.15-5.el6_0.2.i686
--> Processing Dependency: net-snmp for package: quagga-0.99.15-5.el6_0.2.i686
--> Running transaction check
---> Package net-snmp.i686 1:5.5-27.el6_0.1 set to be updated
--> Processing Dependency: libsensors.so.4 for package: 1:net-snmp-5.5-27.el6_0.1.i686
---> Package net-snmp-libs.i686 1:5.5-27.el6_0.1 set to be updated
--> Running transaction check
---> Package lm_sensors-libs.i686 0:3.1.1-10.el6 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved
```

Installing Quagga

```
Dependencies Resolved

================================================================================
 Package                Arch       Version                    Repository     Size
================================================================================
Installing:
 quagga                 i686       0.99.15-5.el6_0.2          updates       1.0 M
Installing for dependencies:
 lm_sensors-libs        i686       3.1.1-10.el6               base           36 k
 net-snmp               i686       1:5.5-27.el6_0.1           updates       297 k
 net-snmp-libs          i686       1:5.5-27.el6_0.1           updates       1.5 M

Transaction Summary
================================================================================
Install       4 Package(s)
Upgrade       0 Package(s)

Total download size: 2.8 M
Installed size: 11 M
Is this ok [y/N]: y
```

Installing Quagga

```
Is this ok [y/N]: y
Downloading Packages:
(1/4): lm_sensors-libs-3.1.1-10.el6.i686.rpm             |  36 kB     00:00
(2/4): net-snmp-5.5-27.el6_0.1.i686.rpm                  | 297 kB     00:00
(3/4): net-snmp-libs-5.5-27.el6_0.1.i686.rpm             | 1.5 MB     00:00
(4/4): quagga-0.99.15-5.el6_0.2.i686.rpm                 | 1.0 MB     00:00
--------------------------------------------------------------------------------
Total                                           2.2 MB/s | 2.8 MB     00:01
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing     : lm_sensors-libs-3.1.1-10.el6.i686                         1/4
  Installing     : 1:net-snmp-libs-5.5-27.el6_0.1.i686                       2/4
  Installing     : 1:net-snmp-5.5-27.el6_0.1.i686                            3/4
  Installing     : quagga-0.99.15-5.el6_0.2.i686                             4/4

Installed:
  quagga.i686 0:0.99.15-5.el6_0.2

Dependency Installed:
  lm_sensors-libs.i686 0:3.1.1-10.el6       net-snmp.i686 1:5.5-27.el6_0.1
  net-snmp-libs.i686 1:5.5-27.el6_0.1

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

Installing Quagga

```
[root@celebrian ~]# rpm -qa | grep quagga
quagga-0.99.15-5.el6_0.2.i686
[root@celebrian ~]#
```

Quagga has been installed

Note, you can use **yum** command to only download rpms (and not install them) with the downloadonly option. Useful for doing installations on systems with no Internet access.

yum install yum-downloadonly
yum install quagga --downloadonly

The downloaded rpms will be found in /var/cache/yum//packages*

Installing Quagga

```
[root@celebrian ~]# rpm -qi quagga
Name        : quagga                       Relocations: (not relocatable)
Version     : 0.99.15                           Vendor: CentOS
Release     : 5.el6_0.2                     Build Date: Sat 25 Jun 2011 05:15:32 AM PDT
Install Date: Tue 15 Nov 2011 06:40:56 AM PST      Build Host: c6b5.bsys.dev.centos.org
Group       : System Environment/Daemons    Source RPM: quagga-0.99.15-5.el6_0.2.src.rpm
Size        : 4431645                          License: GPLv2+
Signature   : RSA/8, Tue 05 Jul 2011 06:45:16 PM PDT, Key ID 0946fca2c105b9de
Packager    : CentOS BuildSystem <http://bugs.centos.org>
URL         : http://www.quagga.net
Summary     : Routing daemon
Description :
Quagga is a free software that manages TCP/IP based routing
protocol. It takes multi-server and multi-thread approach to resolve
the current complexity of the Internet.

Quagga supports BGP4, BGP4+, OSPFv2, OSPFv3, RIPv1, RIPv2, and RIPng.

Quagga is intended to be used as a Route Server and a Route Reflector. It is
not a toolkit, it provides full routing power under a new architecture.
Quagga by design has a process for each protocol.

Quagga is a fork of GNU Zebra.
[root@celebrian ~]#
```

The –qi option on rpm gives you a summary of the package

Celebrian

- Check which samba packages have been installed on Elrond with

  rpm –qa | grep quagga

- Install the samba server package with

  yum install quagga

- Check again which samba packages have been installed on Elrond with

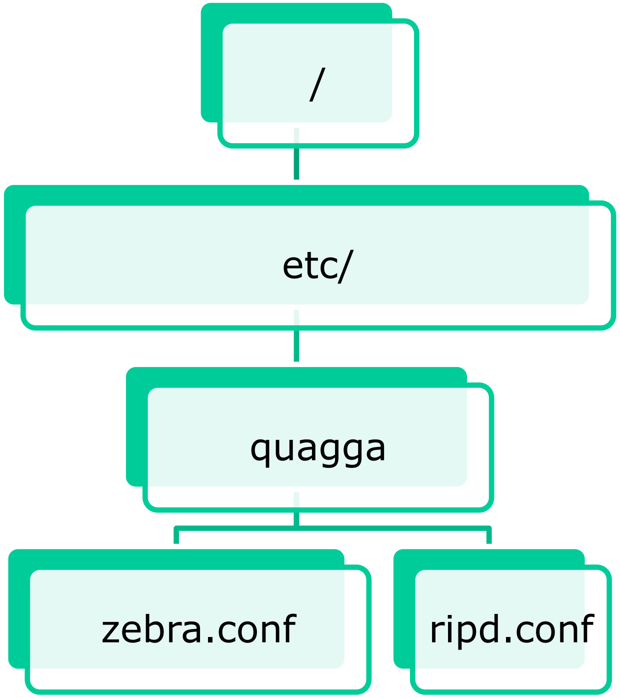
  rpm –qa | grep quagga

- To learn more about a package use

  rpm –qi quagga

/etc/quagga/zebra.conf and /etc/quagga/ripd.conf

Step 2 *Customize the configuration files*



main configuration files for Quagga when implementing RIPv2

/etc/quagga/zebra.conf and /etc/quagga/ripd.conf

```
[root@celebrian ~]# cat /etc/quagga/zebra.conf
hostname arwen.localdomain
!
password quagga
enable password quagga
!
log file /var/log/quagga/zebra.log
```

```
[root@celebrian ~]# cat /etc/quagga/ripd.conf
hostname celebrian.localdomain
log file /var/log/quagga/ripd.log
!
router rip
 network eth0
 network eth1
 redistribute connected
!
line vty
!
```

/etc/quagga/zebra.conf and /etc/quagga/ripd.conf

Set ownership of configuration files

```
[root@celebrian ~]# cd /etc/quagga
[root@celebrian quagga]# chown quagga:quagga ripd.conf zebra.conf

[root@celebrian quagga]# ls -l ripd.conf zebra.conf
-rw-r--r--. 1 quagga quagga 144 Nov 15 07:48 ripd.conf
-rw-r-----. 1 quagga quagga 106 Nov 15 07:47 zebra.conf
```

Quagga and the Firewall

Step 3 *Modify the firewall*

Firewall ports used for implementing Quagga RIPv2

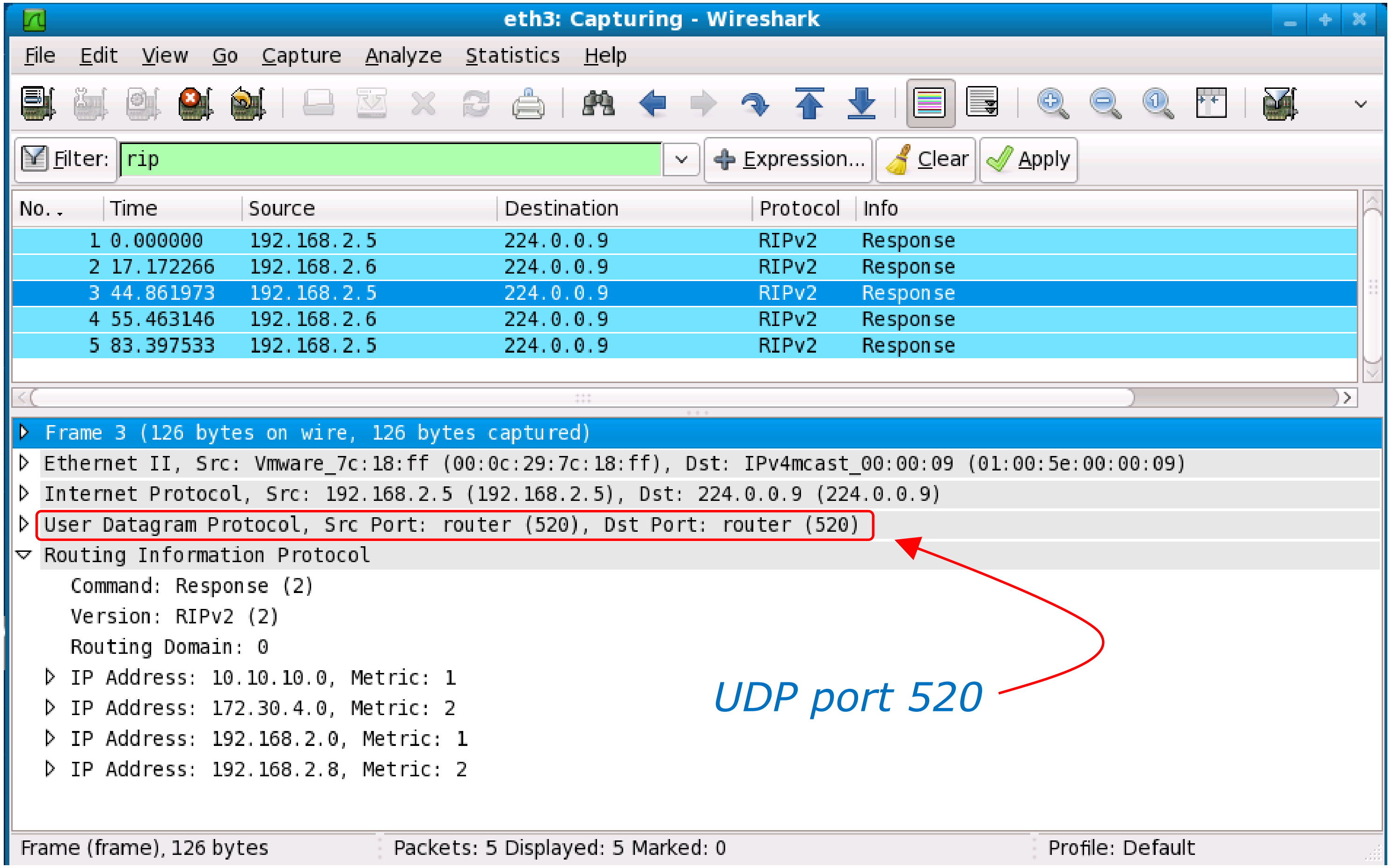
UDP 520	*RIP advertisements*

Other Firewall changes needed for Quagga RIPv2

Routers should be forwarding packets and not filtering them out. In particular the UDP RIP packets must be allowed to pass through the router so they can get to the other routers.

Quagga and the Firewall

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
```

UDP port 520

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

Quagga and the Firewall

Default firewall (in memory)

```
[root@celebrian ~]# iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
2    ACCEPT     icmp --  anywhere             anywhere
3    ACCEPT     all  --  anywhere             anywhere
4    ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
5    REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
[root@celebrian ~]#
```

- *There is no rule on the INPUT chain to accept incoming RIP packets (UDP port 520) so they will be rejected.*
- *All packets going through the FORWARD chain get rejected.*

Quagga and the Firewall

Default firewall (in configuration file)

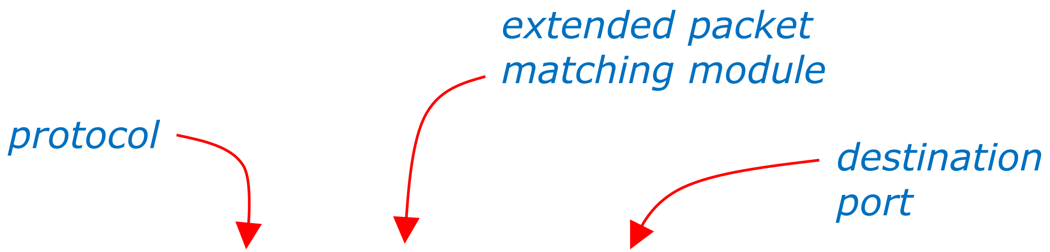
```
[root@celebrian ~]# cat /etc/sysconfig/iptables
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
[root@celebrian ~]#
```

- *There is no rule on the INPUT chain to accept incoming RIP packets (UDP port 520) so they will be rejected.*
- *All packets going through the FORWARD chain get rejected.*

Quagga and the Firewall

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

Delete the first rule on the FORWARD chain allowing all packets to be forwarded



extended packet matching module

protocol

destination port

```
[root@celebrian ~]# iptables -I INPUT 4 -p udp -m udp --dport 520 -j ACCEPT
```

Insert a rule above rule 4 on the INPUT chain to accept incoming packets to UDP port 520

```
[root@celebrian ~]# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[  OK  ]
```

Save the rules in memory to the configuration file

Modifying the Firewall
(Centos)

Modified firewall (in memory)

RIP port open

```
[root@celebrian ~]# iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
2    ACCEPT     udp  --  anywhere             anywhere            udp dpt:router
3    ACCEPT     icmp --  anywhere             anywhere
4    ACCEPT     all  --  anywhere             anywhere
5    ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
6    REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited

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

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
[root@celebrian ~]#
```

No filtering now on forwarded packets

Quagga and the Firewall

Modified firewall (in configuration file)

```
[root@celebrian ~]# cat /etc/sysconfig/iptables
# Generated by iptables-save v1.4.7 on Tue Nov 15 00:41:40 2011
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [11:1740]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p udp -m udp --dport 520 -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Tue Nov 15 00:41:40 2011
[root@celebrian ~]#
```

No filtering now on forwarded packets

RIP port open

Activity

Modify firewall

```
iptables –L

cat /etc/sysconfig/iptables


iptables -I INPUT 4 -p udp -m udp --dport 520 -j ACCEPT

iptables -D FORWARD 1

service iptables save


iptables –L

cat /etc/sysconfig/iptables
```

SELinux

Step 4 *Configure SELinux*

Overview

SELinux is like an internal firewall where you can define what subjects (users, programs) can access which objects (files, devices)

- Originally created by the NSA (National Security Agency)
- Based on the MAC (Mandatory Access Control) concept where administrators control all interactions between programs.
- Programs and users start with no rights. Any rights must be granted by the administrator as part of the security policy for the system.
- Standard UNIX permissions are checked first then SELinux rules are applied if necessary.

SELinux

Security Contexts

Security context have three components: a **user identity**, a **role**, and a **type** (also known as a domain).

```
[root@celebrian quagga]# ls -lZ /etc/quagga/[rz]*.conf
-rw-r--r--. quagga quagga unconfined_u:object_r:zebra_conf_t:s0
  /etc/quagga/ripd.conf
-rw-r-----. quagga quagga unconfined_u:object_r:zebra_conf_t:s0
  /etc/quagga/zebra.conf
```

This context type above is already correct for quagga configuration files, if you did need to reset it use:

```
cd /etc/quagga
chcon -v --type=zebra_conf_t ripd.conf zebra.conf
```

Managing Quagga Services (CentOS)

Step 5 *Start the service*

```
[root@celebrian ~]# service zebra start
Starting zebra:                                            [  OK  ]
[root@celebrian ~]# service ripd start
Starting ripd:                                             [  OK  ]
[  OK  ]
```

Step 6 *Start the service automatically during system startup*

```
[root@celebrian ~]# chkconfig zebra on
[root@celebrian ~]# chkconfig ripd on
```

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

Managing Quagga Services (CentOS)

Step 7 *Monitor and verify service is running*

```
[root@celebrian ~]# service zebra status
zebra (pid 6823) is running...

[root@celebrian ~]# service ripd status
ripd (pid 6836) is running...
```

```
[root@celebrian ~]# ps -ef | grep quagga
quagga    6823     1  0 08:19 ?        00:00:00 zebra -d -A 127.0.0.1 -f /etc/quagga/zebra.conf
quagga    6836     1  0 08:19 ?        00:00:00 ripd -d -A 127.0.0.1 -f /etc/quagga/ripd.conf
root      6862  1856  0 08:20 pts/0    00:00:00 grep quagga
```

Before quagga services were started

```
[root@celebrian ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.2.8     0.0.0.0         255.255.255.252 U     0      0        0 eth0
192.168.2.4     0.0.0.0         255.255.255.252 U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1003   0        0 eth1
[root@celebrian ~]#
```

After quagga services were started

```
[root@celebrian ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.2.8     0.0.0.0         255.255.255.252 U     0      0        0 eth0
192.168.2.0     192.168.2.5     255.255.255.252 UG    2      0        0 eth1
192.168.2.4     0.0.0.0         255.255.255.252 U     0      0        0 eth1
172.30.4.0      192.168.2.10    255.255.255.0   UG    2      0        0 eth0
10.10.10.0      192.168.2.5     255.255.255.0   UG    2      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1003   0        0 eth1
0.0.0.0         192.168.2.10    0.0.0.0         UG    2      0        0 eth0
[root@celebrian ~]#
```

Quagga

Step 7 *Monitor and verify service is running*

```
[root@celebrian ~]# netstat -uln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
udp        0      0 0.0.0.0:520                 0.0.0.0:*
```

UDP port 520 is used for RIP advertisements

Quagga

Step 7 *Monitor and verify service is running*

```
[root@celebrian ~]# netstat -tlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address              Foreign Address      State      PID/Program name
tcp        0      0 localhost:discp-client     *:*                  LISTEN     6823/zebra
tcp        0      0 localhost:discp-server     *:*                  LISTEN     6836/ripd
tcp        0      0 *:ssh                      *:*                  LISTEN     1327/sshd
tcp        0      0 localhost:smtp             *:*                  LISTEN     1403/master
tcp        0      0 *:ssh                      *:*                  LISTEN     1327/sshd
tcp        0      0 localhost:smtp             *:*                  LISTEN     1403/master
[root@celebrian ~]#
```

```
[root@celebrian ~]# netstat -tlnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address              Foreign Address      State      PID/Program name
tcp        0      0 127.0.0.1:2601             0.0.0.0:*            LISTEN     6823/zebra
tcp        0      0 127.0.0.1:2602             0.0.0.0:*            LISTEN     6836/ripd
tcp        0      0 0.0.0.0:22                 0.0.0.0:*            LISTEN     1327/sshd
tcp        0      0 127.0.0.1:25               0.0.0.0:*            LISTEN     1403/master
tcp        0      0 :::22                      :::*                 LISTEN     1327/sshd
tcp        0      0 ::1:25                     :::*                 LISTEN     1403/master
[root@celebrian ~]#
```

zebra and ripd daemons are using TCP ports 2601 and 2602

Quagga

Step 8 *Troubleshoot*

If the Quagga shell write command fails in updating the configuration files:

1. Check config files are owned by quagga

2. Check SELinux context type is zebra_conf_t

Step 8 *Troubleshoot*

Quagga

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

Enable debugging to log RIP events in 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

Quagga

Step 9 *Monitor log files*

```
[root@celebrian ~]# tail /var/log/quagga/zebra.log
2011/11/15 08:19:13 ZEBRA: Zebra 0.99.15 starting: vty@2601
[root@celebrian ~]#
```

```
[root@celebrian ~]# tail /var/log/quagga/ripd.log
2011/11/15 08:38:24 RIP: update timer fire!
2011/11/15 08:38:24 RIP: SEND UPDATE to eth0 ifindex 2
2011/11/15 08:38:24 RIP: multicast announce on eth0
2011/11/15 08:38:24 RIP: update routes on interface eth0 ifindex 2
2011/11/15 08:38:24 RIP: SEND to  224.0.0.9.520
2011/11/15 08:38:24 RIP: SEND UPDATE to eth1 ifindex 3
2011/11/15 08:38:24 RIP: multicast announce on eth1
2011/11/15 08:38:24 RIP: update routes on interface eth1 ifindex 3
2011/11/15 08:38:24 RIP: SEND to  224.0.0.9.520
2011/11/15 08:38:30 RIP: RECV packet from 192.168.2.10 port 520 on eth0
[root@celebrian ~]#
```

Quagga

Step 10 *Configure additional security*

Using Quagga

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

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

Quagga shell

```
celebrian.localdomain# sh run
Building configuration...

Current configuration:
!
hostname arwen.localdomain
log file /var/log/quagga/zebra.log
hostname celebrian.localdomain
log file /var/log/quagga/ripd.log
!
debug rip events
debug rip zebra
!
password quagga
enable password quagga
!
interface eth0
 ipv6 nd suppress-ra
!
interface eth1
 ipv6 nd suppress-ra
!
interface lo
!
interface sit0
 ipv6 nd suppress-ra
end
celebrian.localdomain#
```

Linux shell

```
[root@celebrian ~]# cat /etc/quagga/ripd.conf
!
! Zebra configuration saved from vty
!   2011/11/15 08:52:47
!
hostname celebrian.localdomain
log file /var/log/quagga/ripd.log
!
debug rip events
debug rip zebra
!
router rip
 redistribute connected
 network eth0
 network eth1
!
line vty
!
[root@celebrian ~]#
```

Quagga – A fork of GNU Zebra

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

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

```
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)#
```

Quagga – A fork of GNU Zebra

```
[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)#
```

ripd service

Using the ripd shell to check RIP information

Show routing table

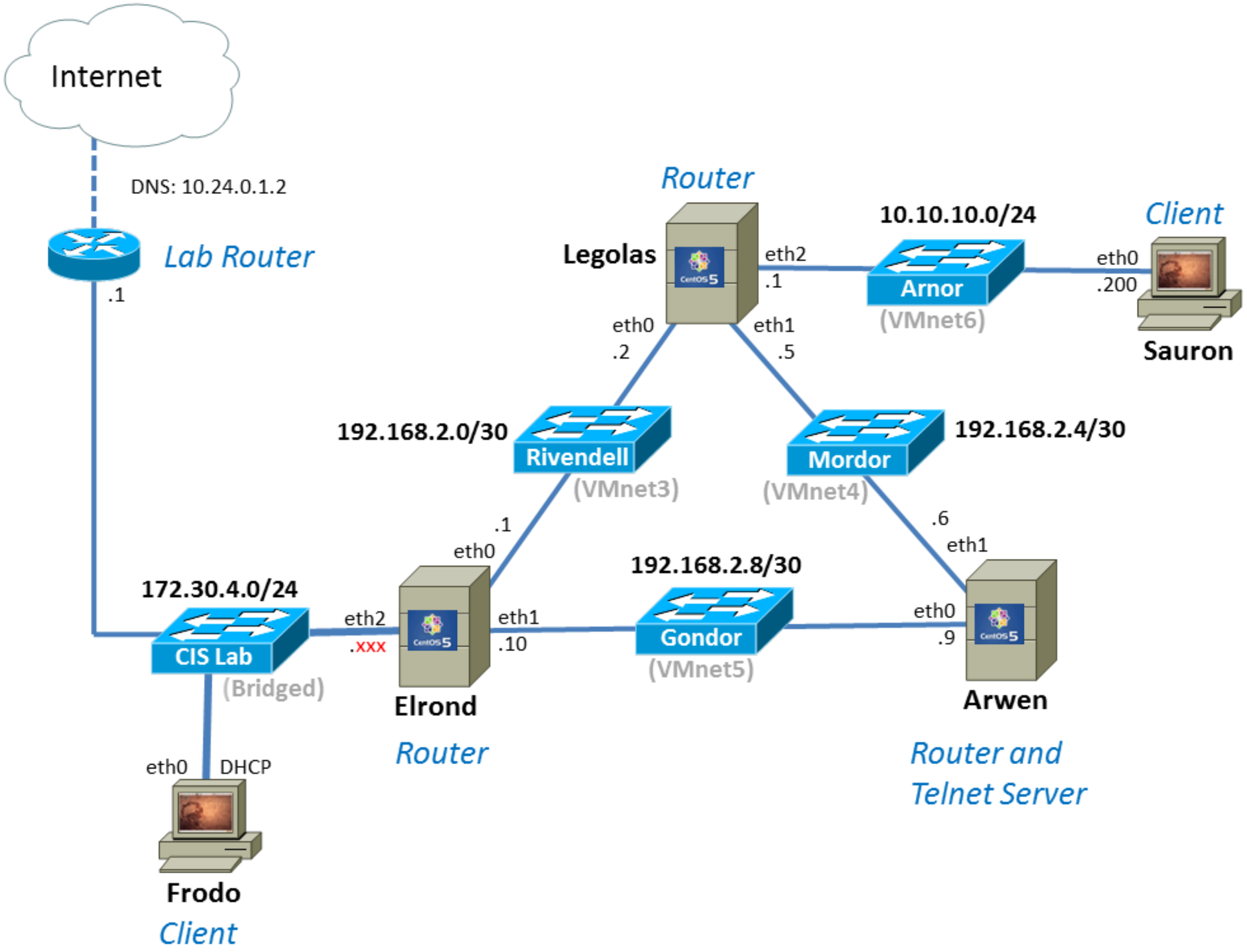
Seeing RIP routes indicates RIP is working between routers

Lab 4 Skills

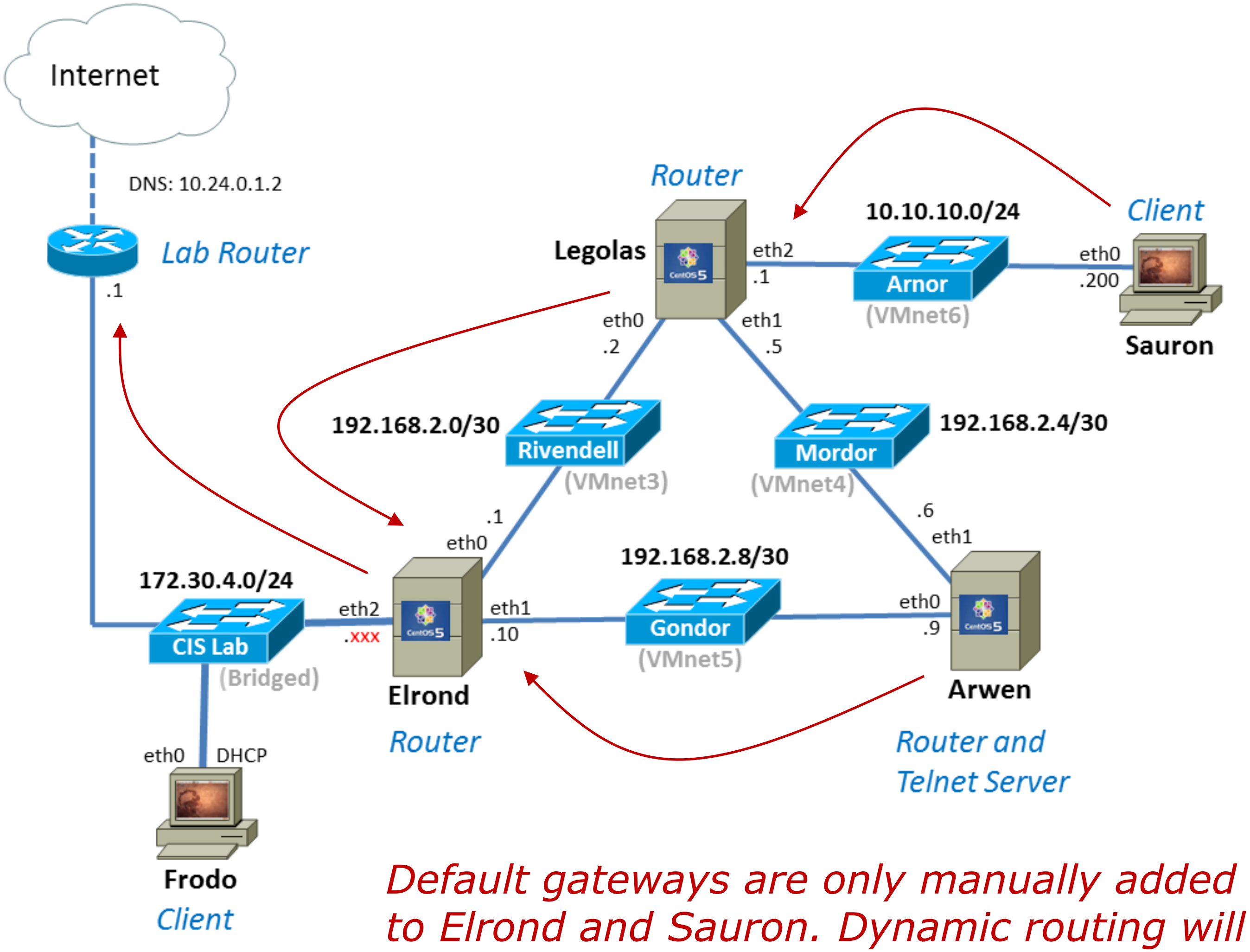
Skills needed for Lab 4!

- 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

The network used for Lab 4



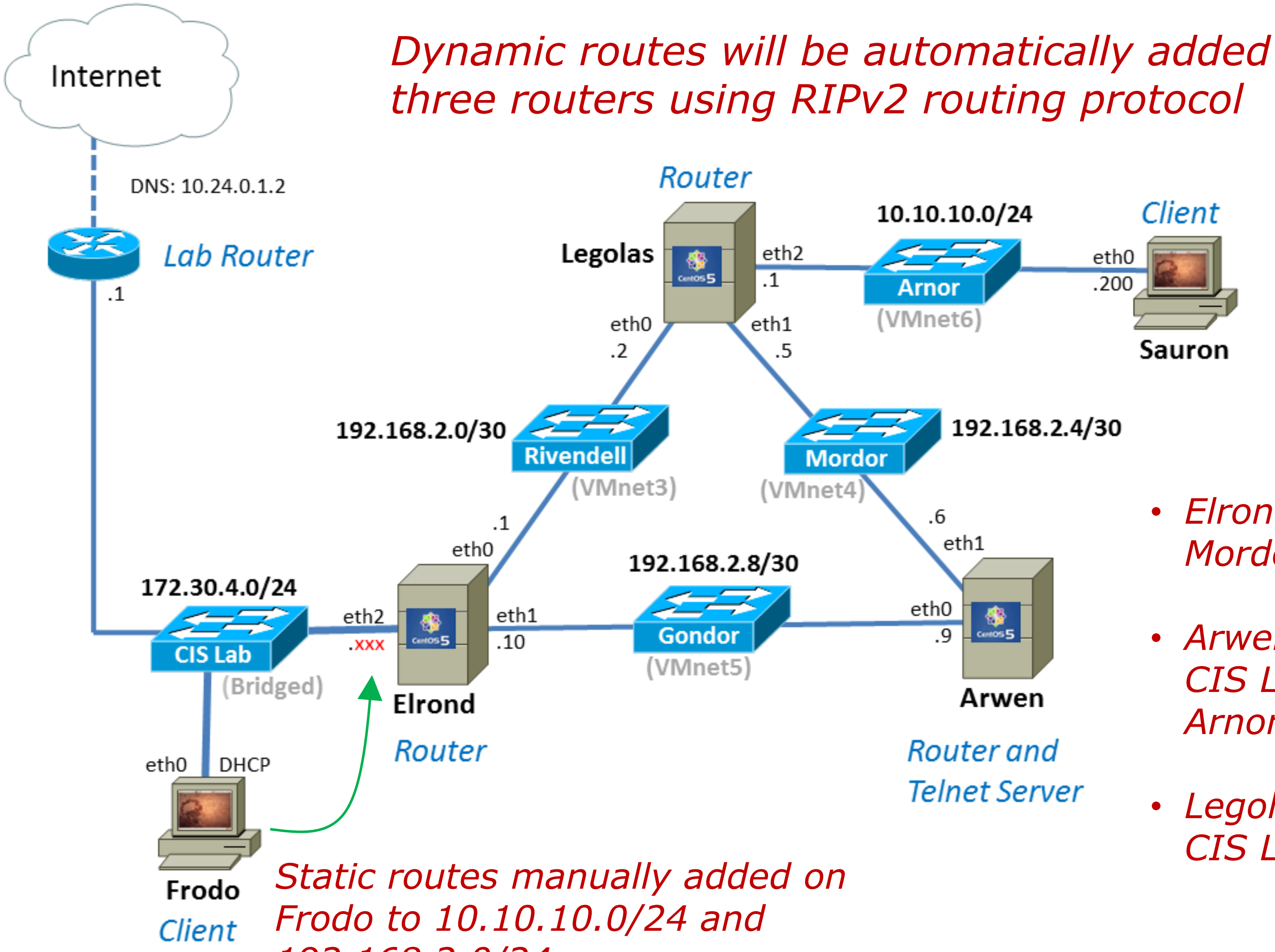
The network used for Lab 4



Default gateways are only manually added to Elrond and Sauron. Dynamic routing will handle adding them to Arwen and Logolas.

The network used for Lab 4

Dynamic routes will be automatically added on all three routers using RIPv2 routing protocol



Static routes manually added on Frodo to 10.10.10.0/24 and 192.168.2.0/24

- *Elrond needs routes to Mordor and Arnor*
- *Arwen needs routes to CIS Lab, Rivendell and Arnor*
- *Legolas need routes to CIS Lab and Gondor*

Adding Hardware

Adding another NIC
(Without going to Fry's)

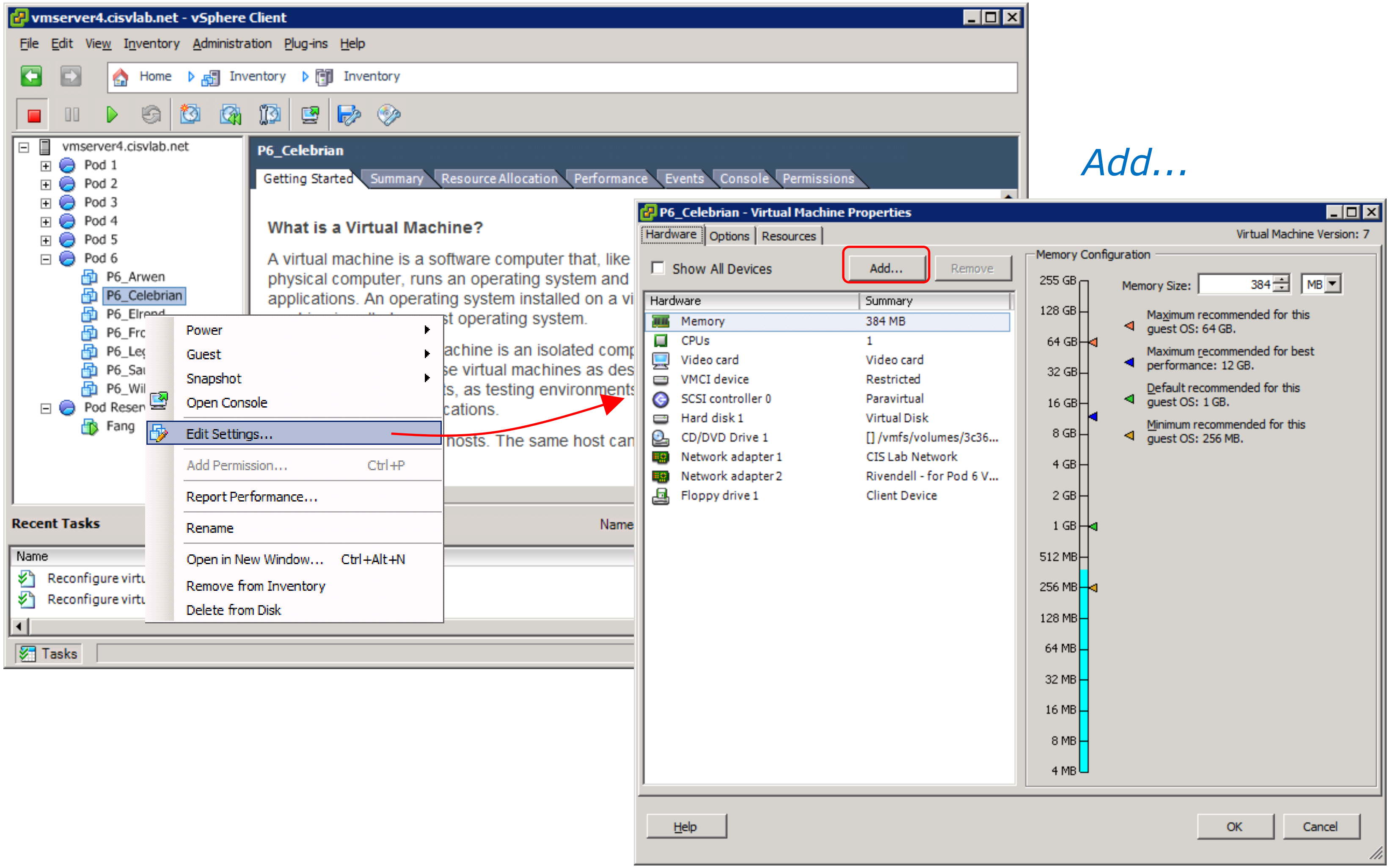
- The VM needs to be powered off
- Start with **(Edit) Settings...**
- Click Add... button to get to the **Add Hardware (Wizard)**
- Add a **(Ethernet) Network Adapter** and keep hitting Next button till added.

Adding another NIC

(VMware ESXi/vSphere)

Right click > Edit Settings...

Add...



What is a Virtual Machine?

A virtual machine is a software computer that, like a physical computer, runs an operating system and applications. An operating system installed on a vi ... st operating system.

... achine is an isolated comp ... se virtual machines as des ... s, as testing environments ... cations.

... hosts. The same host can

Hardware	Summary
Memory	384 MB
CPUs	1
Video card	Video card
VMCI device	Restricted
SCSI controller 0	Paravirtual
Hard disk 1	Virtual Disk
CD/DVD Drive 1	[] /vmfs/volumes/3c36...
Network adapter 1	CIS Lab Network
Network adapter 2	Rivendell - for Pod 6 V...
Floppy drive 1	Client Device

Memory Size: 384 MB

Maximum recommended for this guest OS: 64 GB.

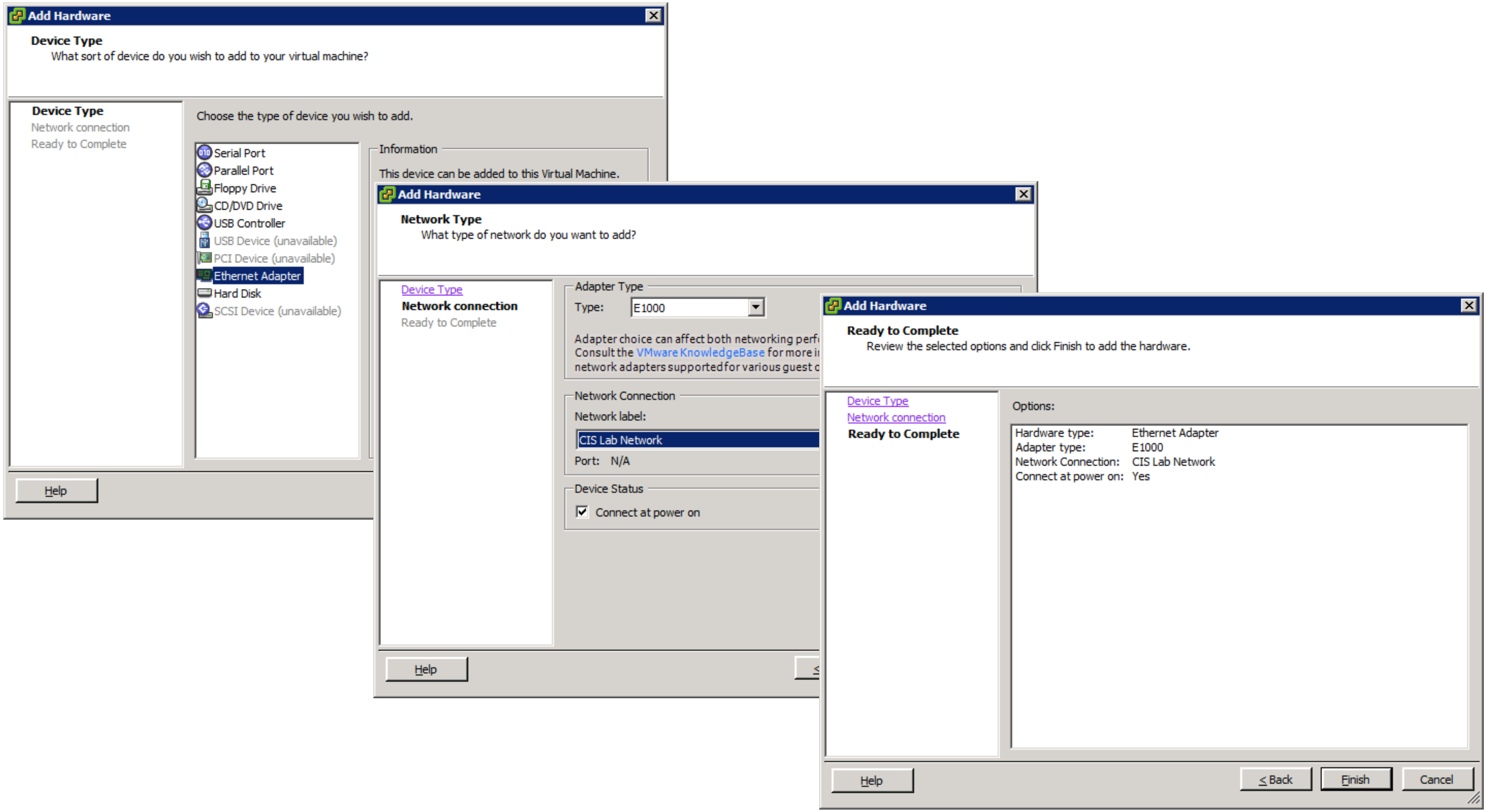
Maximum recommended for best performance: 12 GB.

Default recommended for this guest OS: 1 GB.

Minimum recommended for this guest OS: 256 MB.

Adding another NIC
(VMware ESXi/vSphere)

Add hardware wizard



Add Hardware

Device Type
What sort of device do you wish to add to your virtual machine?

Device Type
Network connection
Ready to Complete

Choose the type of device you wish to add.

- Serial Port
- Parallel Port
- Floppy Drive
- CD/DVD Drive
- USB Controller
- USB Device (unavailable)
- PCI Device (unavailable)
- Ethernet Adapter
- Hard Disk
- SCSI Device (unavailable)

Information
This device can be added to this Virtual Machine.

Add Hardware

Network Type
What type of network do you want to add?

Device Type
Network connection
Ready to Complete

Adapter Type
Type: E1000

Adapter choice can affect both networking perf
Consult the VMware KnowledgeBase for more i
network adapters supported for various guest o

Network Connection
Network label:
CIS Lab Network
Port: N/A

Device Status
☑ Connect at power on

Add Hardware

Ready to Complete
Review the selected options and click Finish to add the hardware.

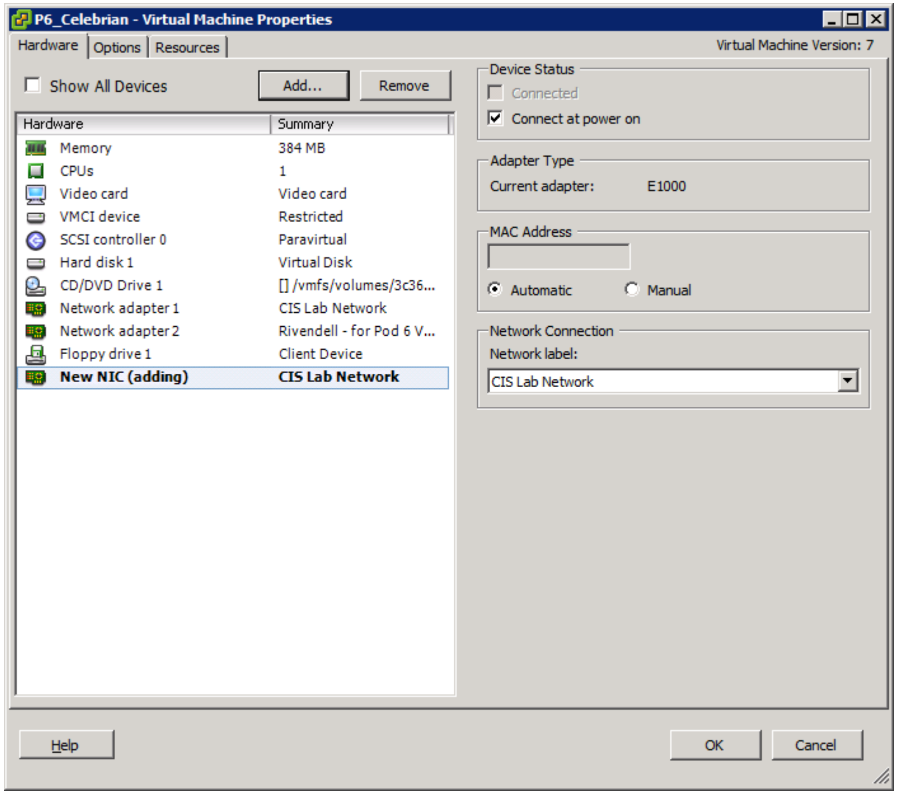
Device Type
Network connection
Ready to Complete

Options:

Hardware type:	Ethernet Adapter
Adapter type:	E1000
Network Connection:	CIS Lab Network
Connect at power on:	Yes

Adding another NIC
(VMware ESXi/vSphere)

Can still be cancelled



P6_Celebrian - Virtual Machine Properties

Hardware | Options | Resources — Virtual Machine Version: 7

☐ Show All Devices — Add... | Remove

Hardware	Summary
Memory	384 MB
CPUs	1
Video card	Video card
VMCI device	Restricted
SCSI controller 0	Paravirtual
Hard disk 1	Virtual Disk
CD/DVD Drive 1	[] /vmfs/volumes/3c36...
Network adapter 1	CIS Lab Network
Network adapter 2	Rivendell - for Pod 6 V...
Floppy drive 1	Client Device
New NIC (adding)	**CIS Lab Network**

Device Status
☐ Connected
☑ Connect at power on

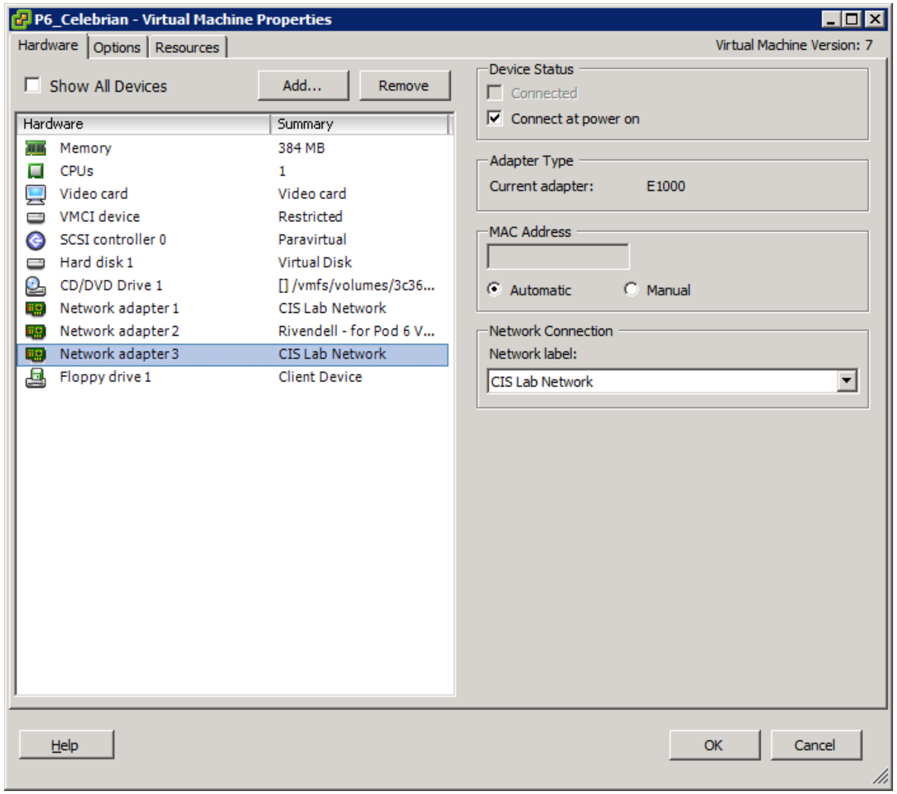
Adapter Type
Current adapter: E1000

MAC Address
● Automatic ○ Manual

Network Connection
Network label: CIS Lab Network

Help | OK | Cancel

New NIC added



P6_Celebrian - Virtual Machine Properties

Hardware | Options | Resources — Virtual Machine Version: 7

☐ Show All Devices — Add... | Remove

Hardware	Summary
Memory	384 MB
CPUs	1
Video card	Video card
VMCI device	Restricted
SCSI controller 0	Paravirtual
Hard disk 1	Virtual Disk
CD/DVD Drive 1	[] /vmfs/volumes/3c36...
Network adapter 1	CIS Lab Network
Network adapter 2	Rivendell - for Pod 6 V...
Network adapter 3	CIS Lab Network
Floppy drive 1	Client Device

Device Status
☐ Connected
☑ Connect at power on

Adapter Type
Current adapter: E1000

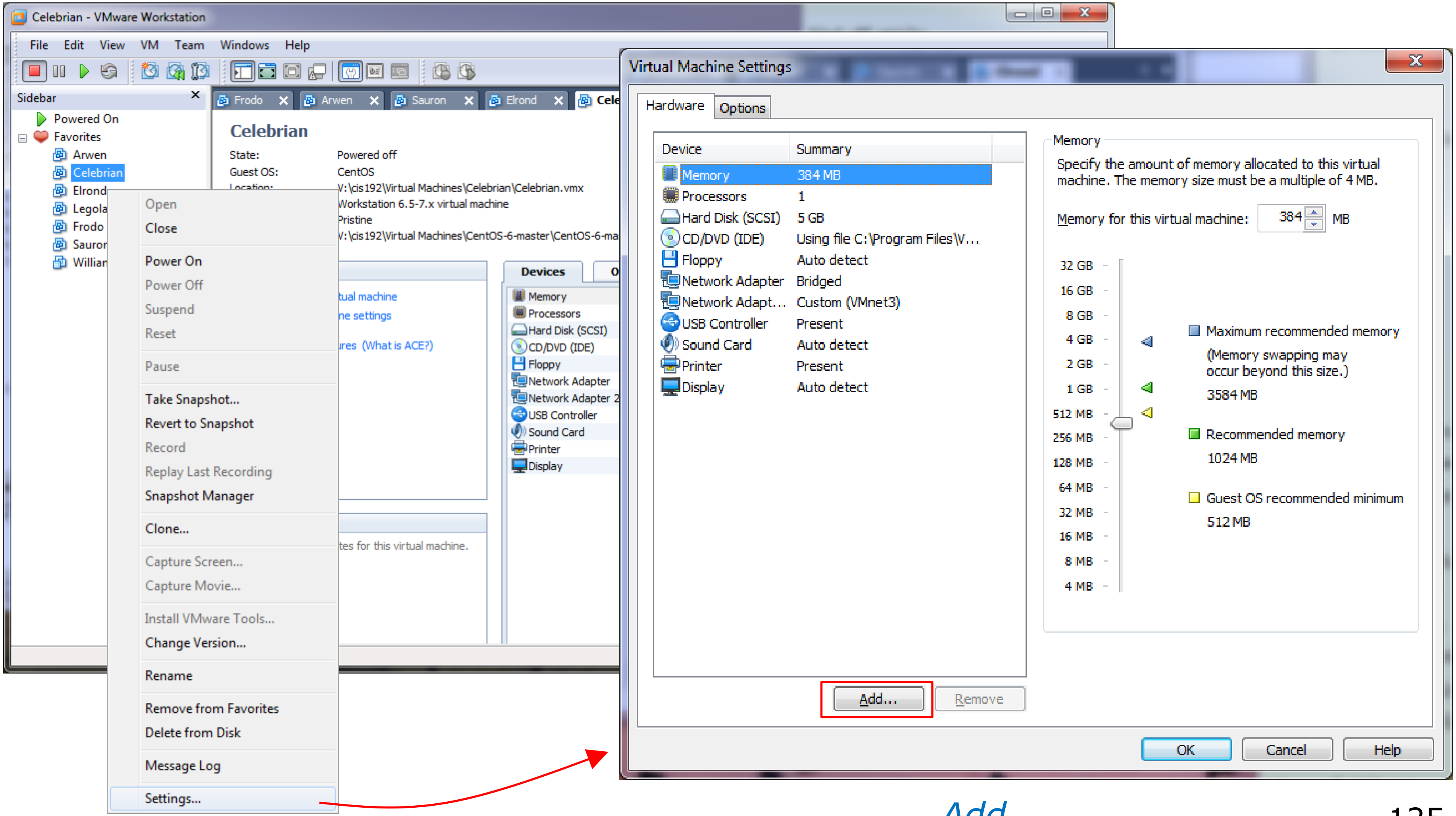
MAC Address
● Automatic ○ Manual

Network Connection
Network label: CIS Lab Network

Help | OK | Cancel

Adding another NIC
(VMware Workstation)

Right click > Settings...



Celebrian

State: Powered off
Guest OS: CentOS
Location: V:\cis192\Virtual Machines\Celebrian\Celebrian.vmx
Workstation 6.5-7.x virtual machine
Pristine
V:\cis192\Virtual Machines\CentOS-6-master\CentOS-6-ma

Device	Summary
Memory	384 MB
Processors	1
Hard Disk (SCSI)	5 GB
CD/DVD (IDE)	Using file C:\Program Files\V...
Floppy	Auto detect
Network Adapter	Bridged
Network Adapt...	Custom (VMnet3)
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

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: 384 MB

🟦 Maximum recommended memory
(Memory swapping may occur beyond this size.)
3584 MB

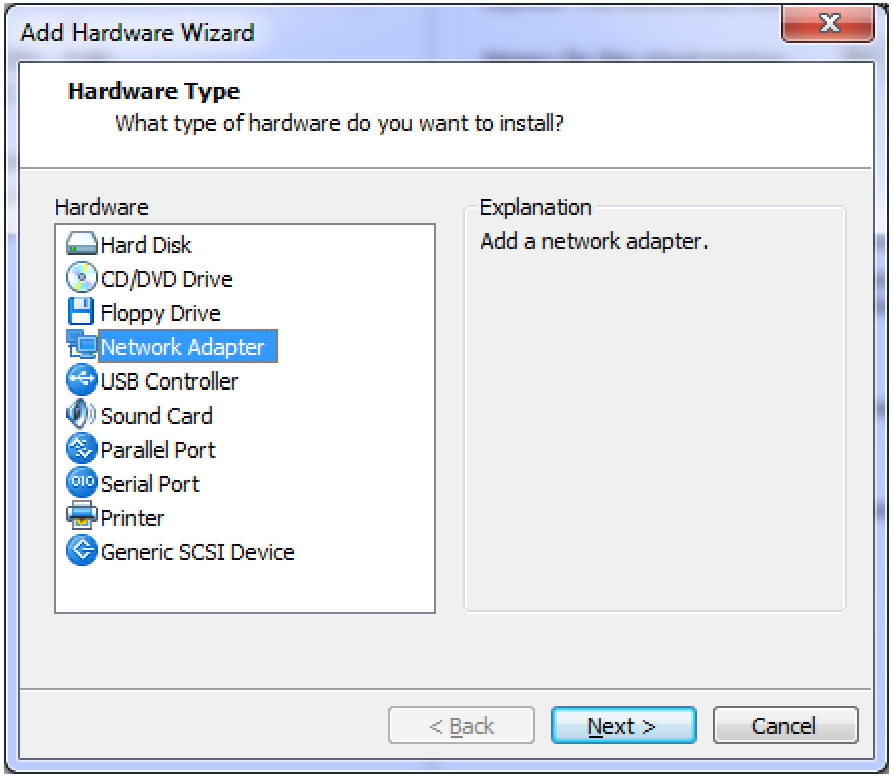
🟩 Recommended memory
1024 MB

🟨 Guest OS recommended minimum
512 MB

Add...

Adding another NIC
(VMware Workstation)

Add hardware wizard

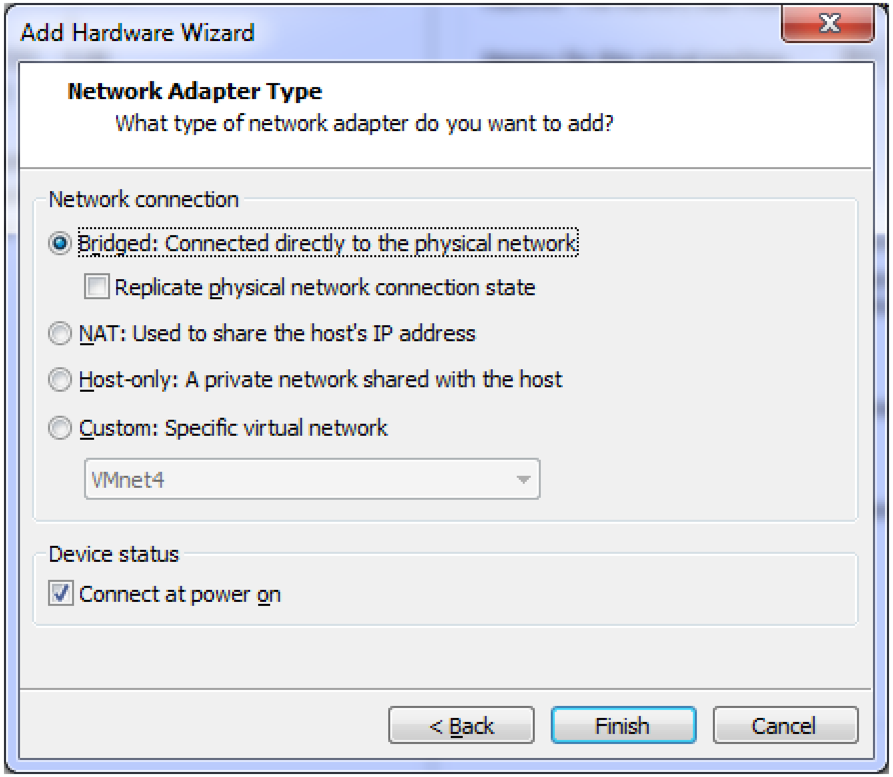


Add Hardware Wizard

Hardware Type
What type of hardware do you want to install?

Hardware
- Hard Disk
- CD/DVD Drive
- Floppy Drive
- Network Adapter
- USB Controller
- Sound Card
- Parallel Port
- Serial Port
- Printer
- Generic SCSI Device

Explanation
Add a network adapter.



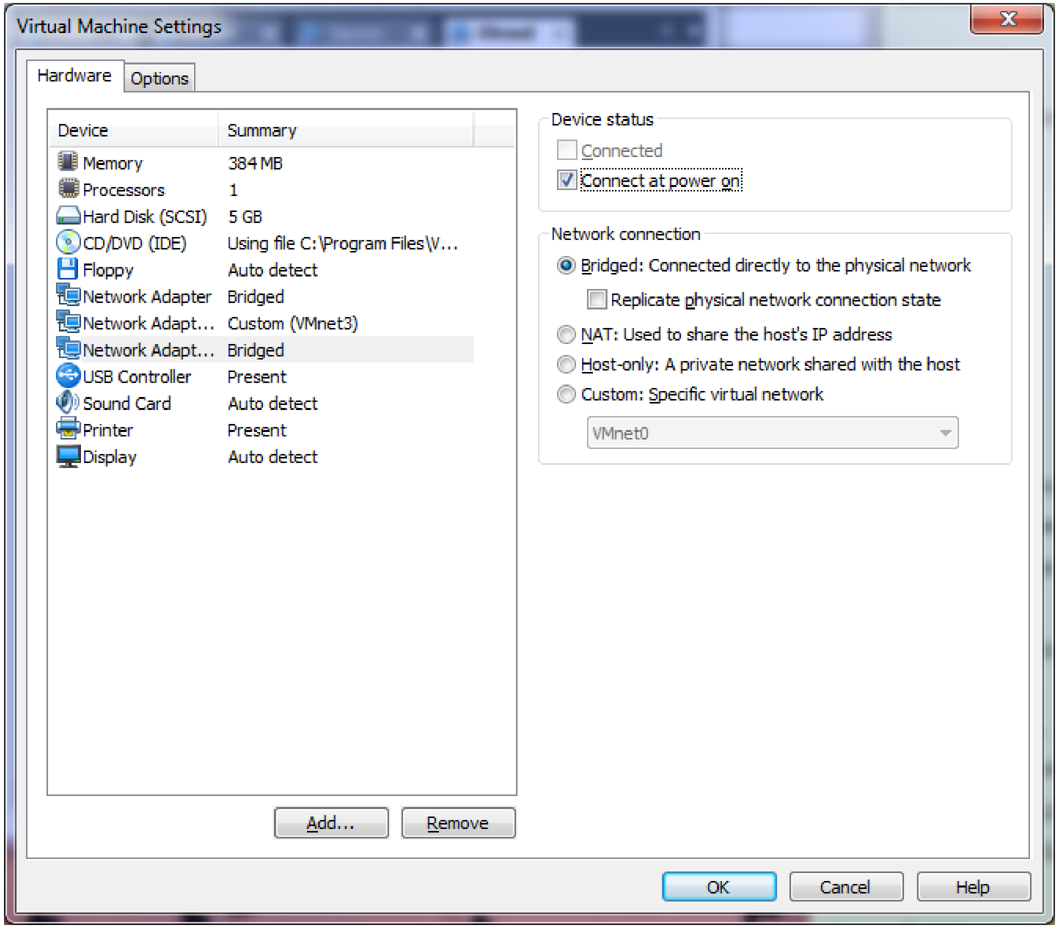
Add Hardware Wizard

Network Adapter Type
What type of network adapter do you want to add?

Network connection
- ● Bridged: Connected directly to the physical network
  - □ Replicate physical network connection state
- ○ NAT: Used to share the host's IP address
- ○ Host-only: A private network shared with the host
- ○ Custom: Specific virtual network
  - VMnet4

Device status
☑ Connect at power on

Adding another NIC
(VMware Workstation)



Virtual Machine Settings

Hardware | Options

Device	Summary
Memory	384 MB
Processors	1
Hard Disk (SCSI)	5 GB
CD/DVD (IDE)	Using file C:\Program Files\V…
Floppy	Auto detect
Network Adapter	Bridged
Network Adapt…	Custom (VMnet3)
Network Adapt…	Bridged
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Device status

□ Connected

☑ Connect at power on

Network connection

● Bridged: Connected directly to the physical network

□ Replicate physical network connection state

○ NAT: Used to share the host's IP address

○ Host-only: A private network shared with the host

○ Custom: Specific virtual network

VMnet0

New NIC added

Activity
(Adding new hardware)

Live Demo

Cabling NICs

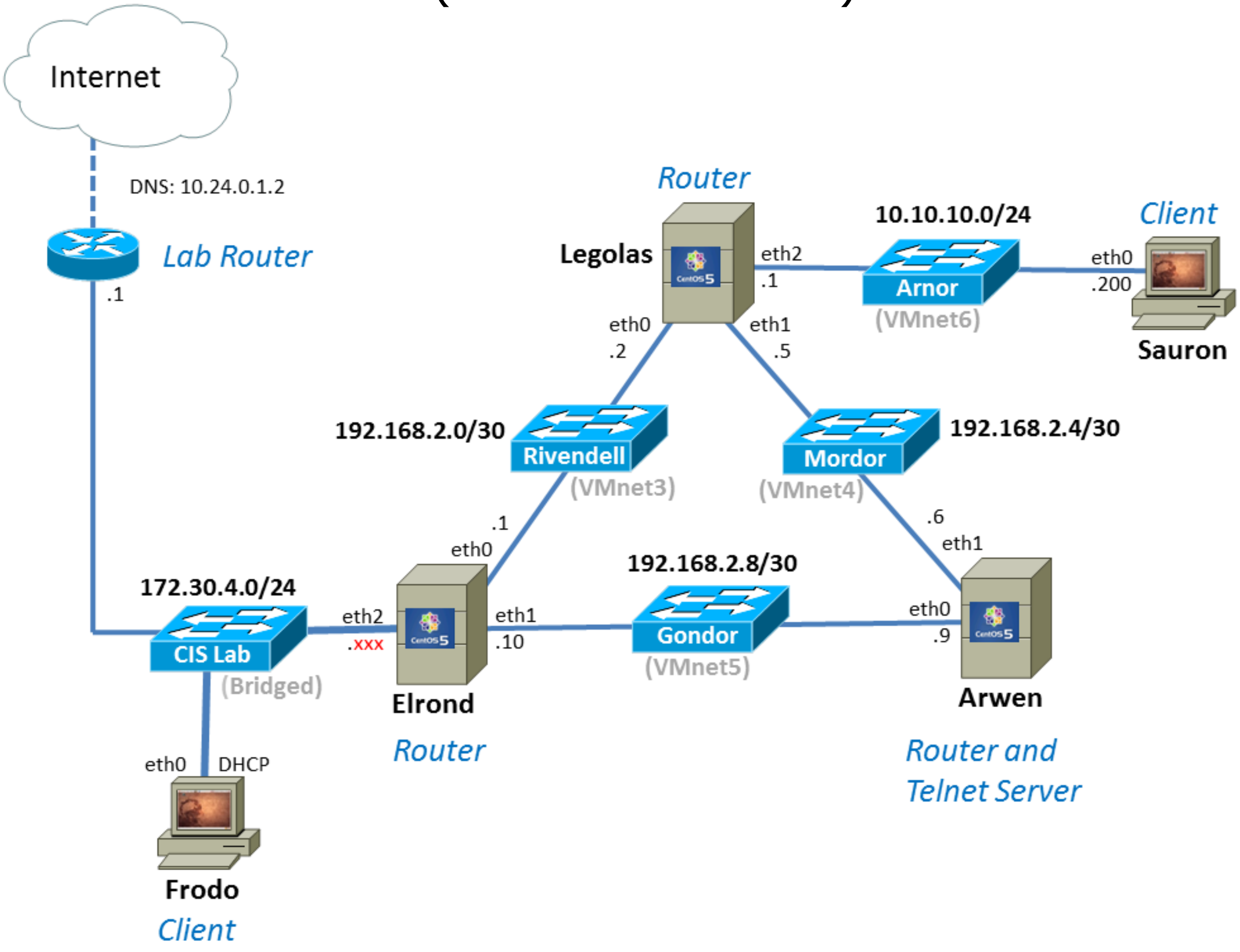
Cabling NICs

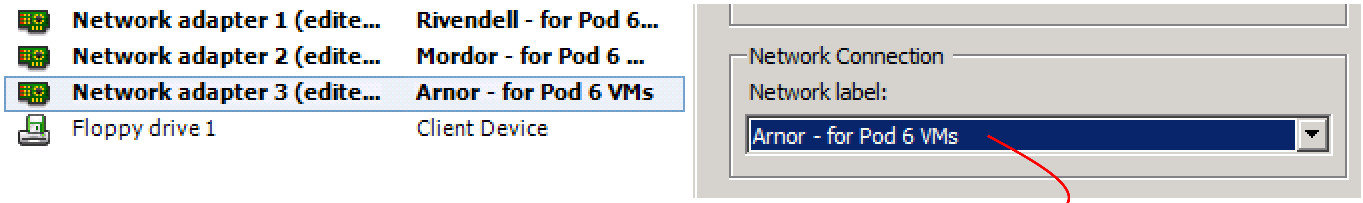
(A must for Lab 4)

- 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 virtual networks.

Cabling NICs
(A must for Lab 4)



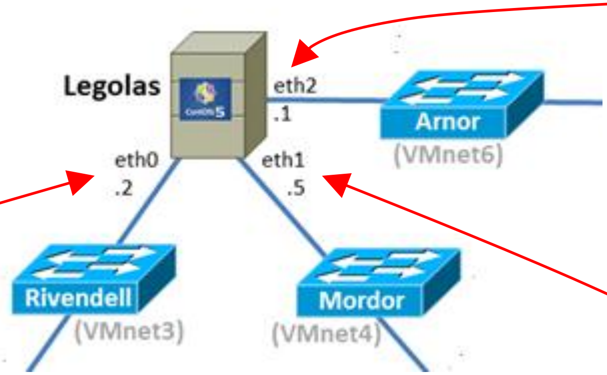


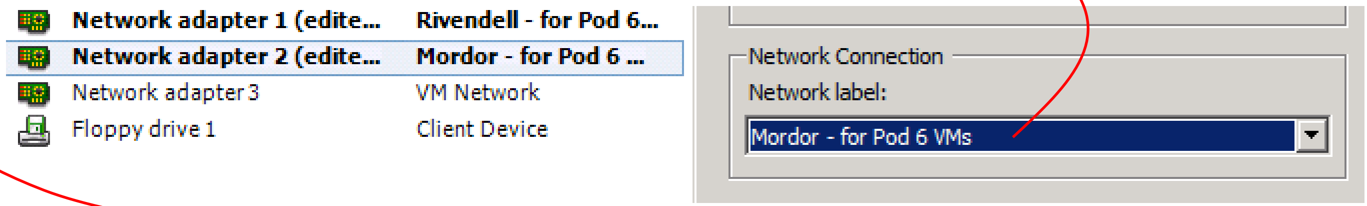
Network adapter 1 (edite...	Rivendell - for Pod 6...
Network adapter 2 (edite...	Mordor - for Pod 6 ...
Network adapter 3 (edite...	Arnor - for Pod 6 VMs
Floppy drive 1	Client Device

Network Connection

Network label: Arnor - for Pod 6 VMs

On VLab Pod 6

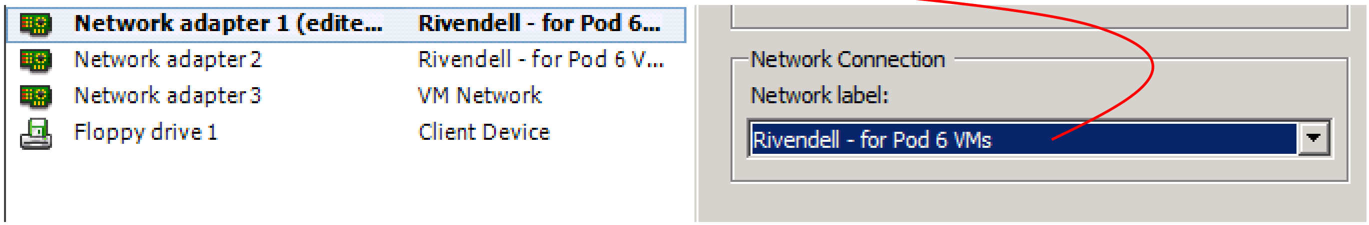




Network adapter 1 (edite...	Rivendell - for Pod 6...
Network adapter 2 (edite...	Mordor - for Pod 6 ...
Network adapter 3	VM Network
Floppy drive 1	Client Device

Network Connection

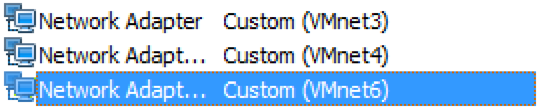
Network label: Mordor - for Pod 6 VMs

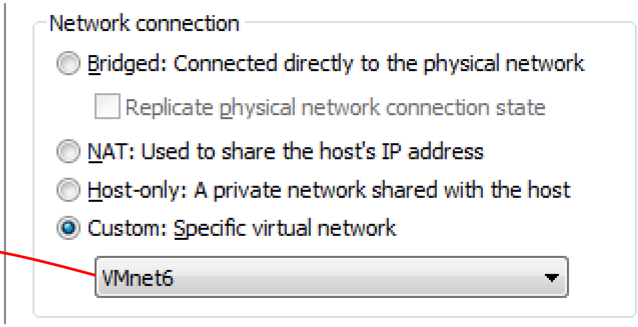


Network adapter 1 (edite...	Rivendell - for Pod 6...
Network adapter 2	Rivendell - for Pod 6 V...
Network adapter 3	VM Network
Floppy drive 1	Client Device

Network Connection

Network label: Rivendell - for Pod 6 VMs


Network Adapter Custom (VMnet3)
Network Adapt… Custom (VMnet4)
Network Adapt… Custom (VMnet6)


Network connection

○ Bridged: Connected directly to the physical network

□ Replicate physical network connection state

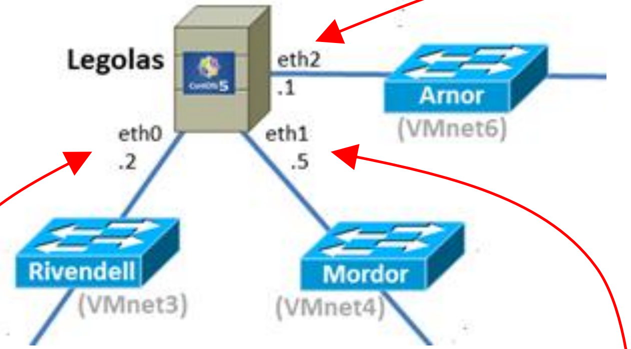
○ NAT: Used to share the host's IP address

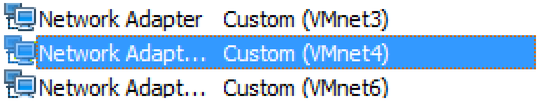
○ Host-only: A private network shared with the host

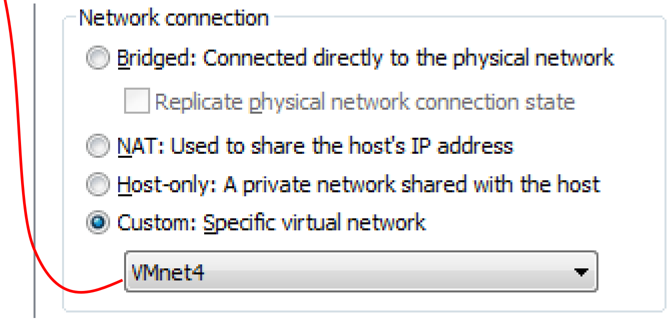
● Custom: Specific virtual network

VMnet6

On a CIS Lab workstation




Network Adapter Custom (VMnet3)
Network Adapt… Custom (VMnet4)
Network Adapt… Custom (VMnet6)


Network connection

○ Bridged: Connected directly to the physical network

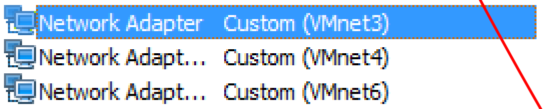
□ Replicate physical network connection state

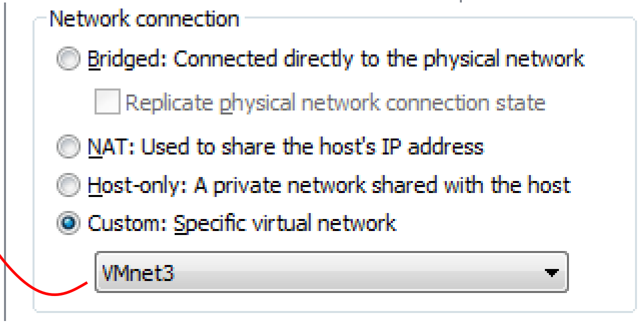
○ NAT: Used to share the host's IP address

○ Host-only: A private network shared with the host

● Custom: Specific virtual network

VMnet4


Network Adapter Custom (VMnet3)
Network Adapt… Custom (VMnet4)
Network Adapt… Custom (VMnet6)


Network connection

○ Bridged: Connected directly to the physical network

□ Replicate physical network connection state

○ NAT: Used to share the host's IP address

○ Host-only: A private network shared with the host

● Custom: Specific virtual network

VMnet3

Activity

Cabling Legolas for Lab 4

Live Demo

Telnet Server

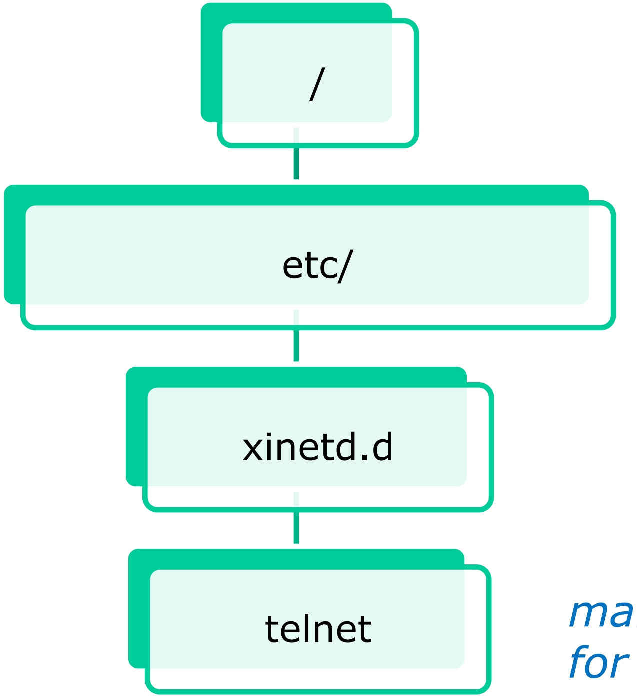
Installing and Configuring Telnet

Step 1 *Install software*

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

Installing and Configuring Telnet

Step 2 *Customize the configuration files*



main configuration file for telnet

Installing and Configuring Telnet

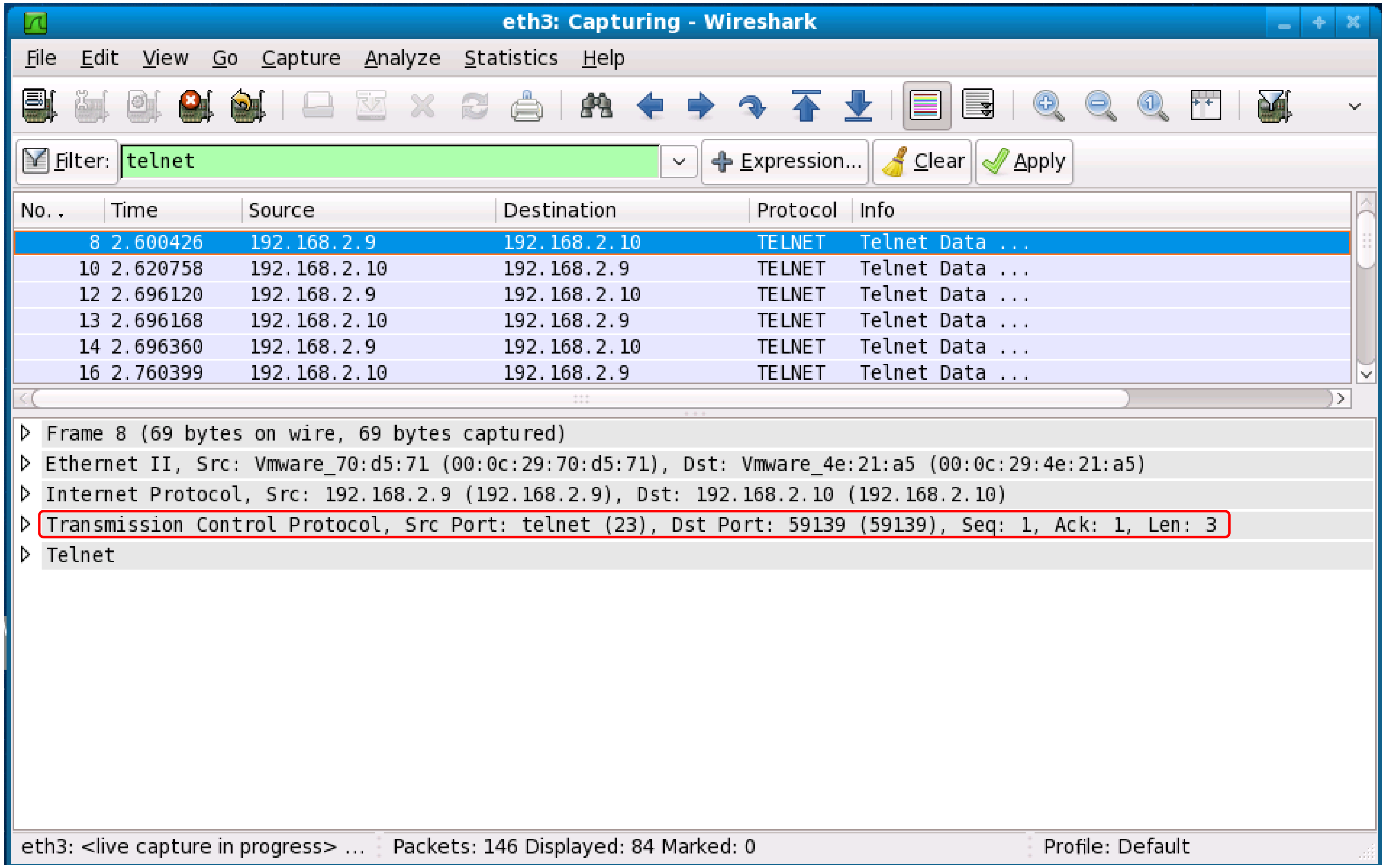
Step 2 *Customize the configuration file*

```
[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

Step 3 *Modify the firewall*

Firewall must be modified to accept new packets to TDP port 23



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

```
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

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

Installing and Configuring Telnet

Step 3 *Modify the firewall*

Show the firewall rules with line numbers

iptables -L --line-numbers

Insert rule to allow new incoming telnet connections

iptables -I INPUT 5 -p tcp -m state --state NEW -m tcp --dport 23 -j ACCEPT

Verify

```
[root@celebrian ~]# iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
2    ACCEPT     icmp --  anywhere             anywhere
3    ACCEPT     all  --  anywhere             anywhere
4    ACCEPT     udp  --  anywhere             anywhere            udp dpt:router
5    ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:telnet
6    ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
7    REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited

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

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

Installing and Configuring Telnet

Step 4 *Configure SELinux*

More later

Installing and Configuring Telnet

Step 5 *Start the service*

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

Step 6 *Start the service automatically during system startup*

```
[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
        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

Installing and Configuring Telnet

Step 7 *Monitor and verify service is running*

```
[root@celebrian ~]# netstat -tlp
[root@celebrian ~]# netstat -tln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 127.0.0.1:2601              0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:2602              0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:22                  0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:25                0.0.0.0:*                   LISTEN
tcp        0      0 :::22                       :::*                        LISTEN
tcp        0      0 :::23                       :::*                        LISTEN
tcp        0      0 ::1:25                      :::*                        LISTEN
[root@celebrian ~]#
```

telnet daemons listens on TCP port 23

Installing and Configuring Telnet

Step 8 *Troubleshoot*

More later

Telnet

Step 9 *Monitor log files*

```
Nov 15 09:13:19 celebrian xinetd[6922]: failed to parse
192.168.2.* [file=/etc/xinetd.d/telnet] [line=10]
Nov 15 09:13:19 celebrian xinetd[6922]: xinetd Version 2.3.14
started with libwrap loadavg labeled-networking options
compiled in.
Nov 15 09:13:19 celebrian xinetd[6922]: Started working: 1
available service
Nov 15 12:29:49 celebrian xinetd[6922]: Exiting...
Nov 15 12:29:49 celebrian xinetd[6998]: failed to parse
192.168.2. [file=/etc/xinetd.d/telnet] [line=10]
Nov 15 12:29:49 celebrian xinetd[6998]: xinetd Version 2.3.14
started with libwrap loadavg labeled-networking options
compiled in.
Nov 15 12:29:49 celebrian xinetd[6998]: Started working: 1
available service
[root@celebrian ~]#
```

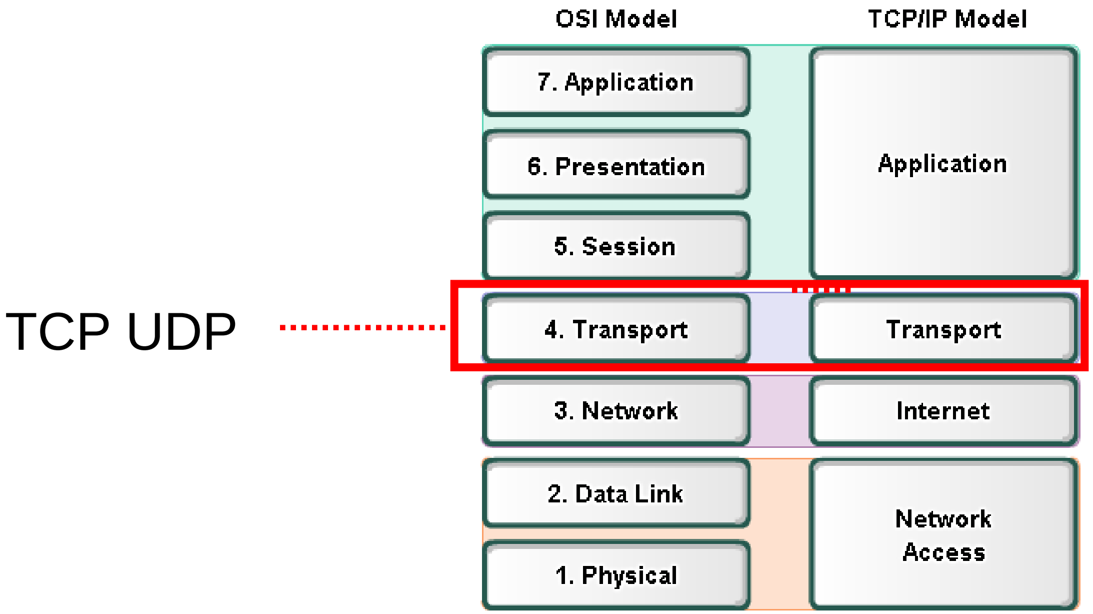
Quagga

Step 10 *Configure additional security*

More later

Transport Layer Overview

Transport Layer



- 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

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***

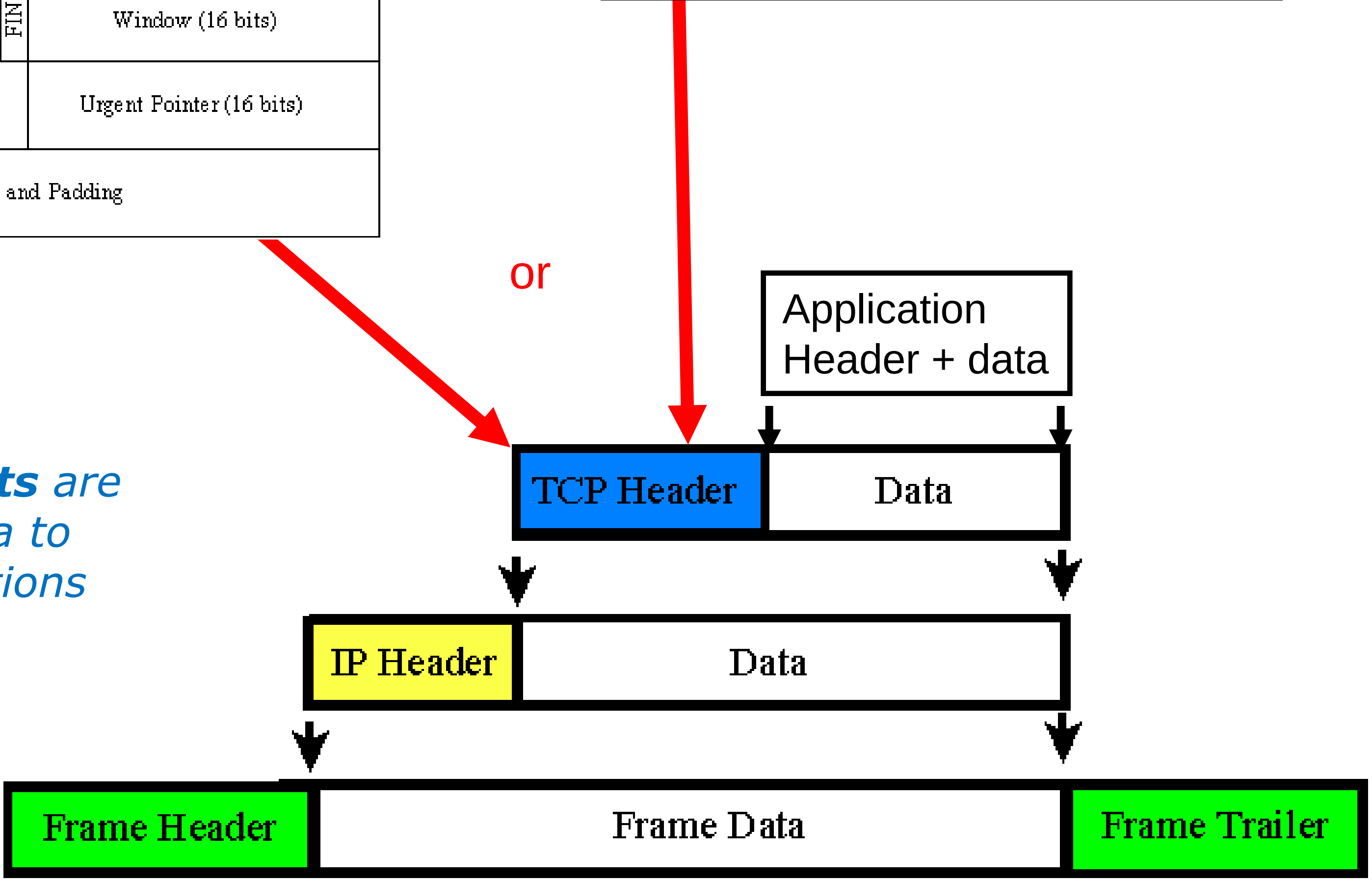
Transmission Control Protocol (TCP)
Connection-oriented
Statefull *(like new or established states in firewalls)*
Reliable The TCP packet is called a ***segment***

TCP Header

<table>
<tr><td colspan="8">Source Port (16 bits)</td><td>Destination Port (16 bits)</td></tr>
<tr><td colspan="9">Sequence Number (32 bits)</td></tr>
<tr><td colspan="9">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>Window (16 bits)</td></tr>
<tr><td colspan="8">Checksum (16 bits)</td><td>Urgent Pointer (16 bits)</td></tr>
<tr><td colspan="9">Options and Padding</td></tr>
</table>

UDP Header

Source Port (16 bits)	Destination Port (16 bits)
Length (16 bits)	Checksum (16 bits)
Data....	



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

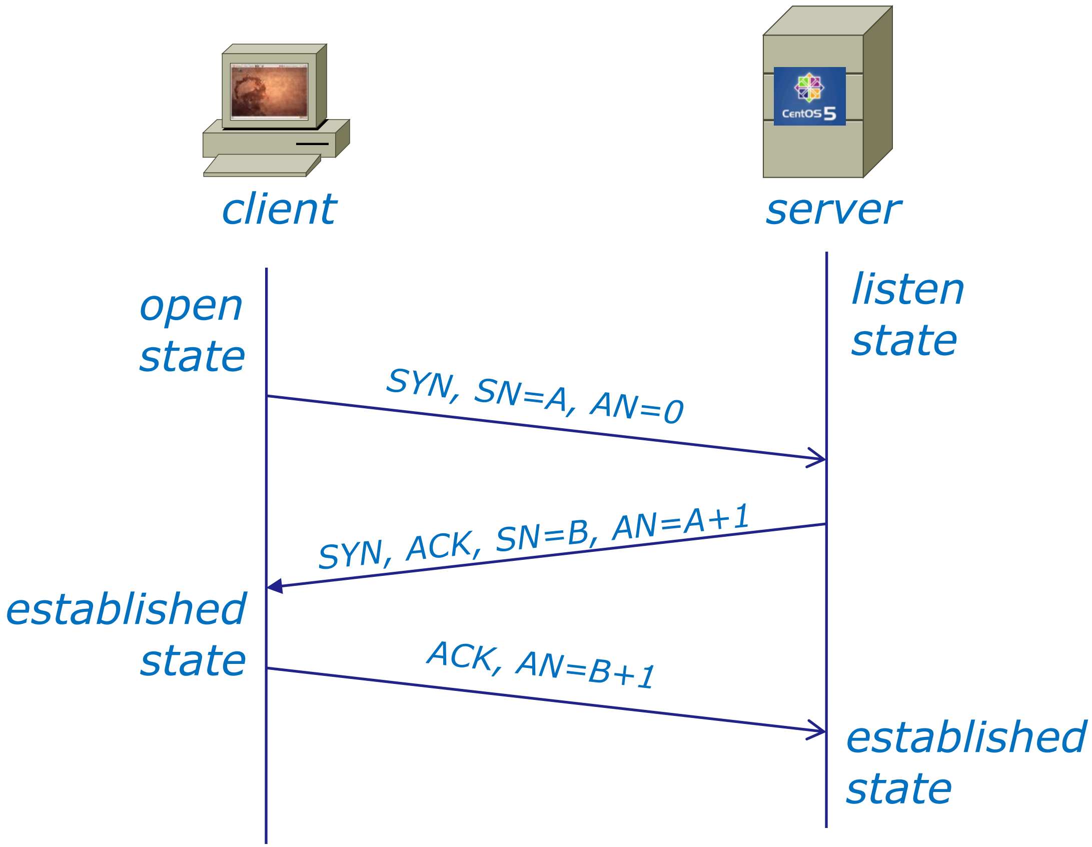
Transport Layer

The Transmission Control Protocol

Initial Connection

Three-Way Handshake

1. SYN
2. SYN-ACK
3. ACK



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

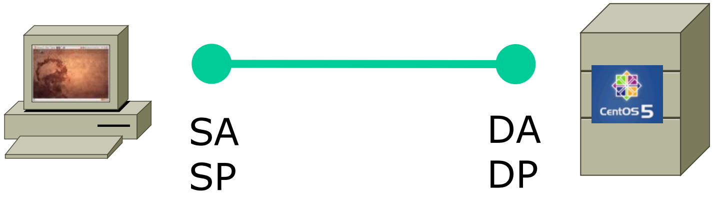
Transport Layer

Sockets

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

- Source IP address
- Source port number

- Destination IP address
- Destination port number



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

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

Transport Layer

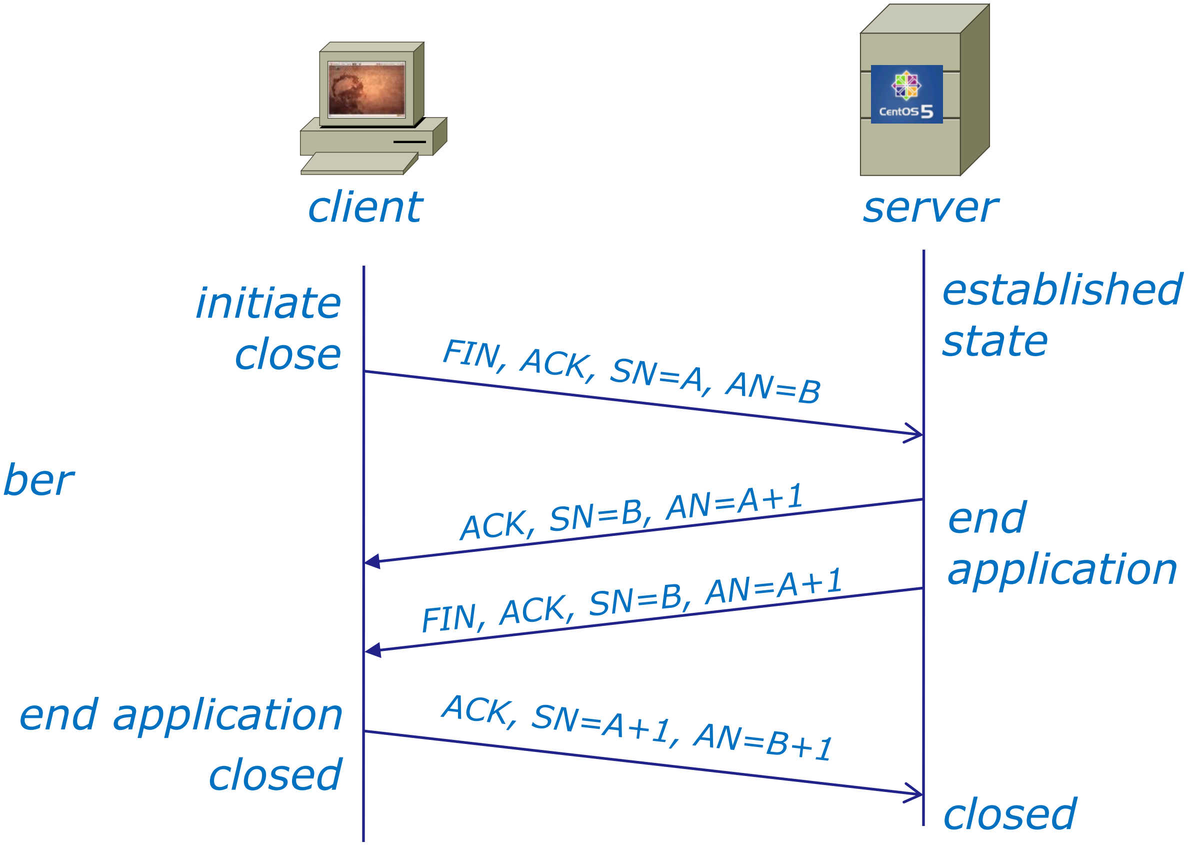
The Transmission Control Protocol

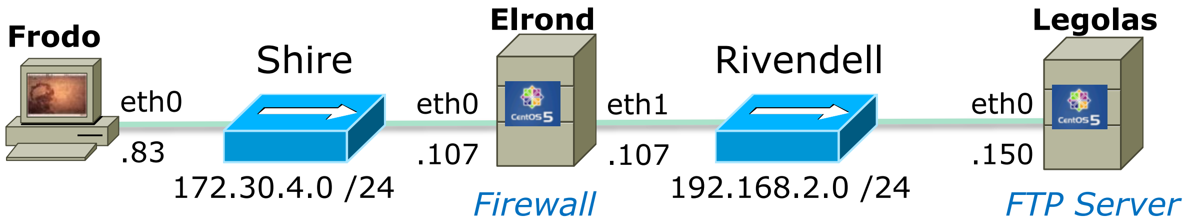
Closing a Connection

Four-Way Handshake

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

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



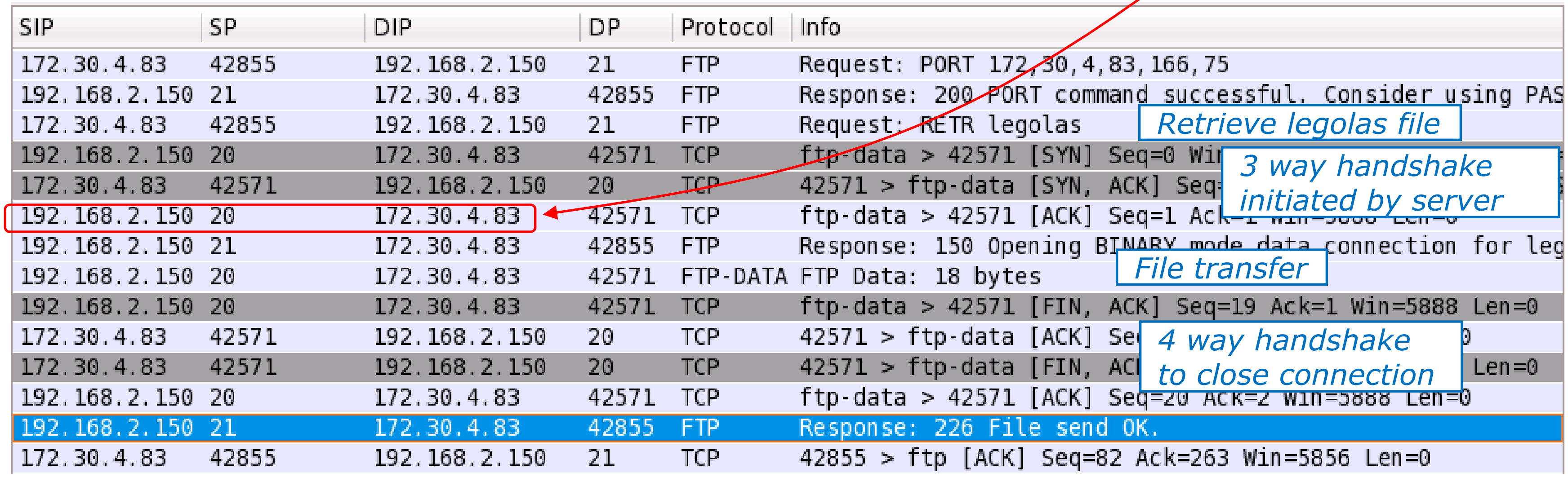


***Active Mode** is when server initiates new connection for data transfer*

Socket for data transfer

Client	Server
172.30.4.83	192.168.2.150
42571	20

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



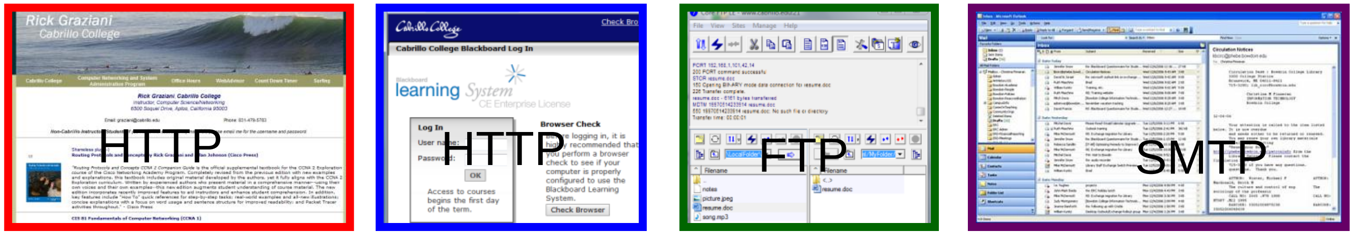
SIP	SP	DIP	DP	Protocol	Info
172.30.4.83	42855	192.168.2.150	21	FTP	Request: PORT 172,30,4,83,166,75
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 200 PORT command successful. Consider using PAS
172.30.4.83	42855	192.168.2.150	21	FTP	Request: RETR legolas
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [SYN] Seq=0 Wi
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [SYN, ACK] Seq
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [ACK] Seq=1 Ack=1 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
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [FIN, AC 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

Retrieve legolas file

3 way handshake initiated by server

File transfer

4 way handshake to close connection

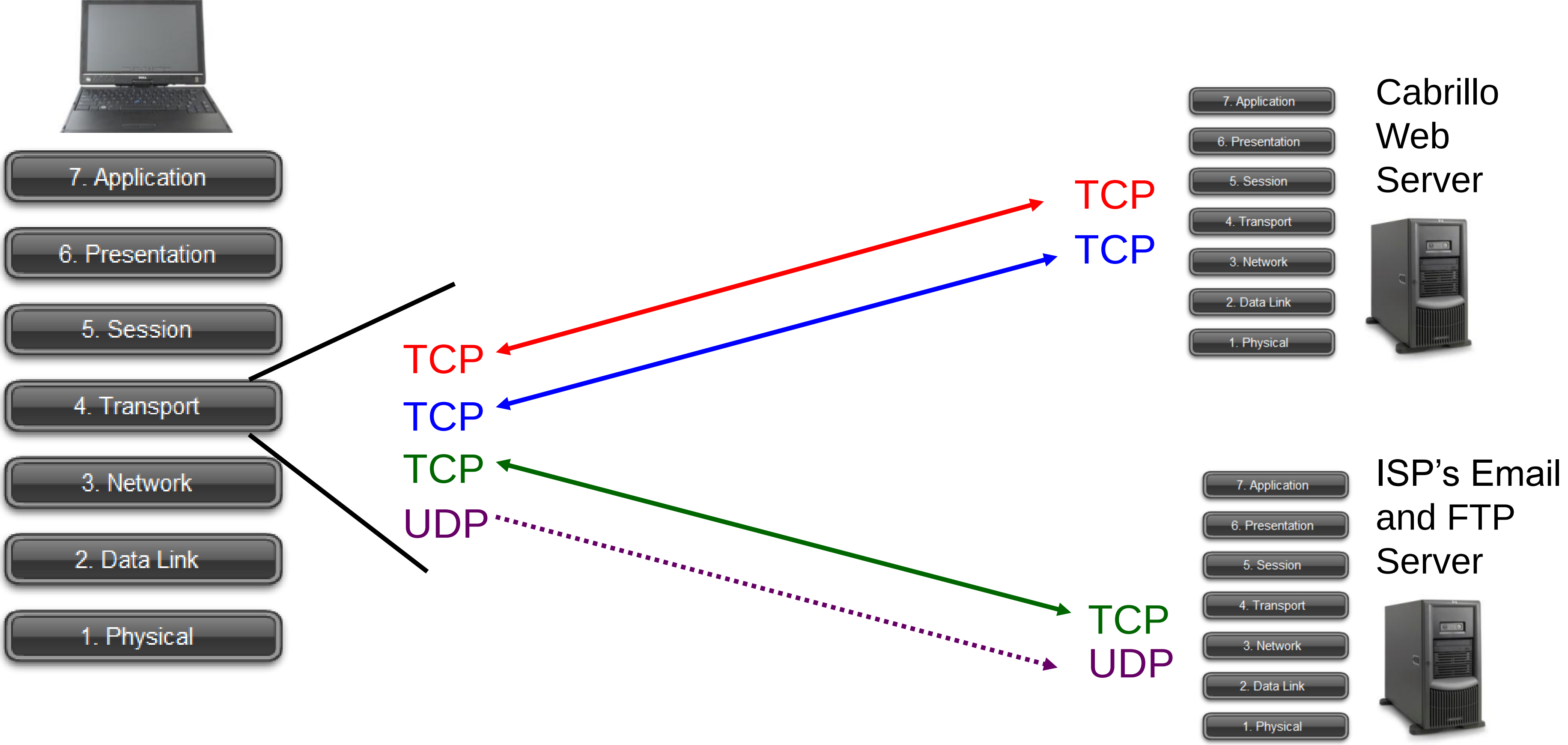


HTTP

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.

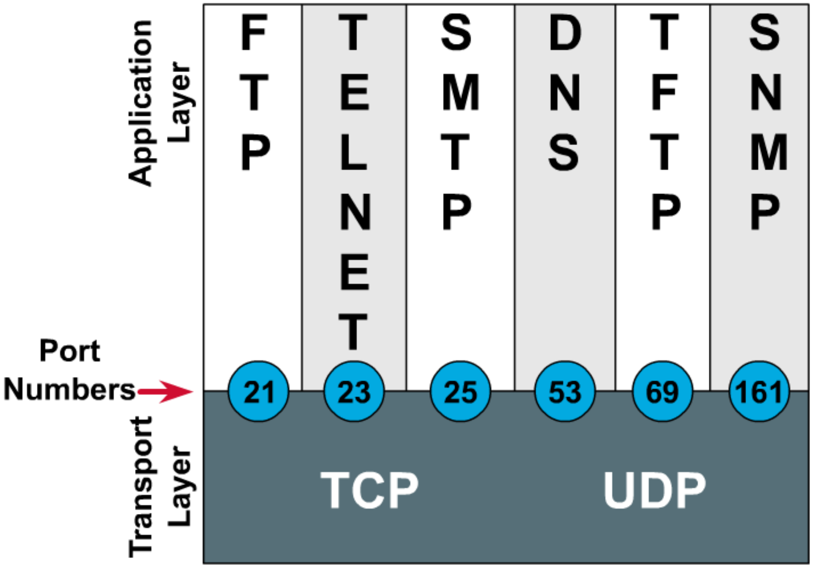
UDP Header

Source Port (16 bits)	Destination Port (16 bits)
Length (16 bits)	Checksum (16 bits)
Data....	

TCP Header

<table>
<tr><td colspan="8">0   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>U
R
G</td><td>A
C
K</td><td>P
S
H</td><td>R
S
T</td><td>S
Y
N</td><td>F
I
N</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>

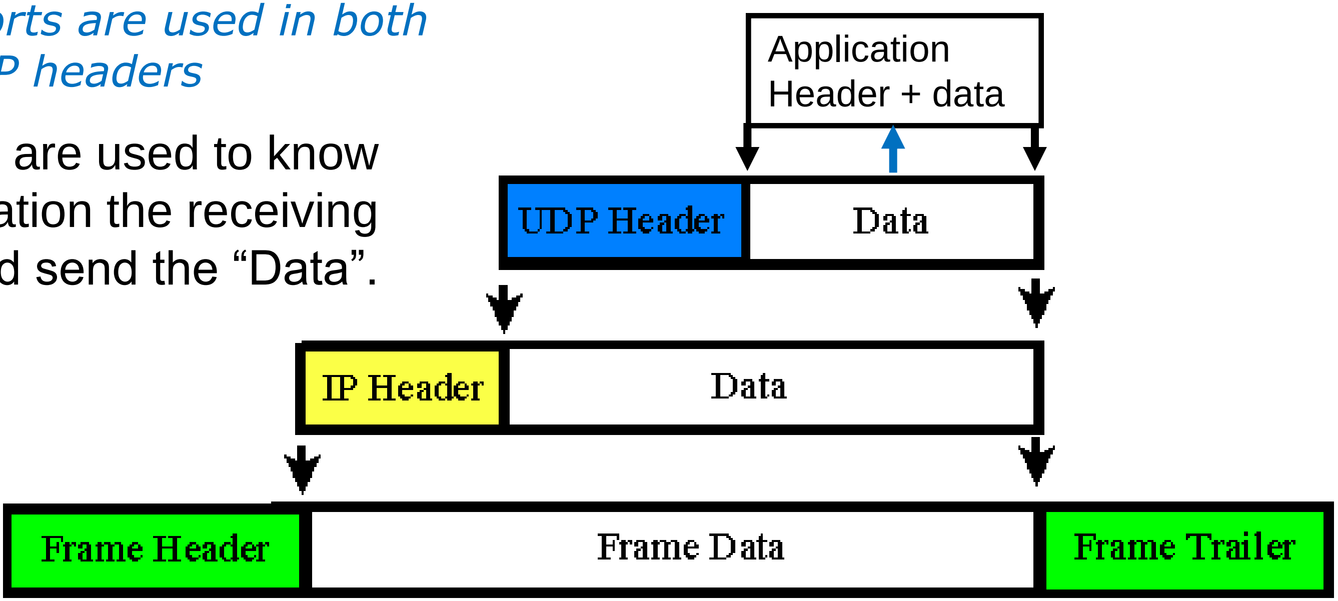
Port Numbers



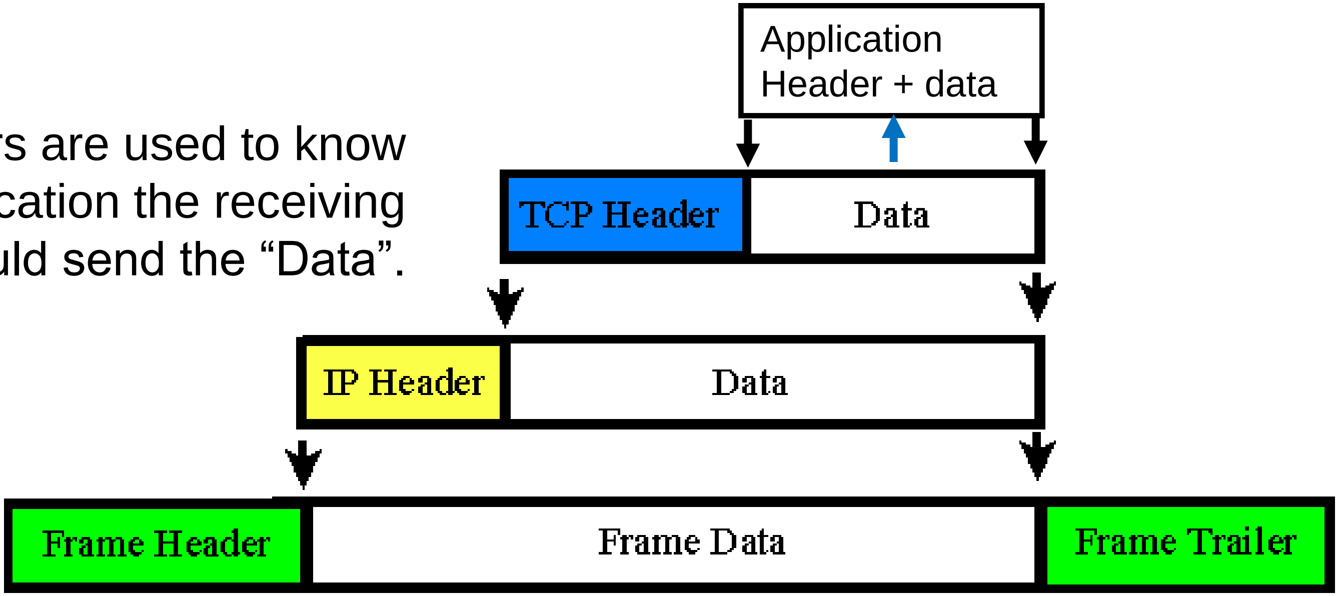
E.g. HTTP is Port 80

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”.

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

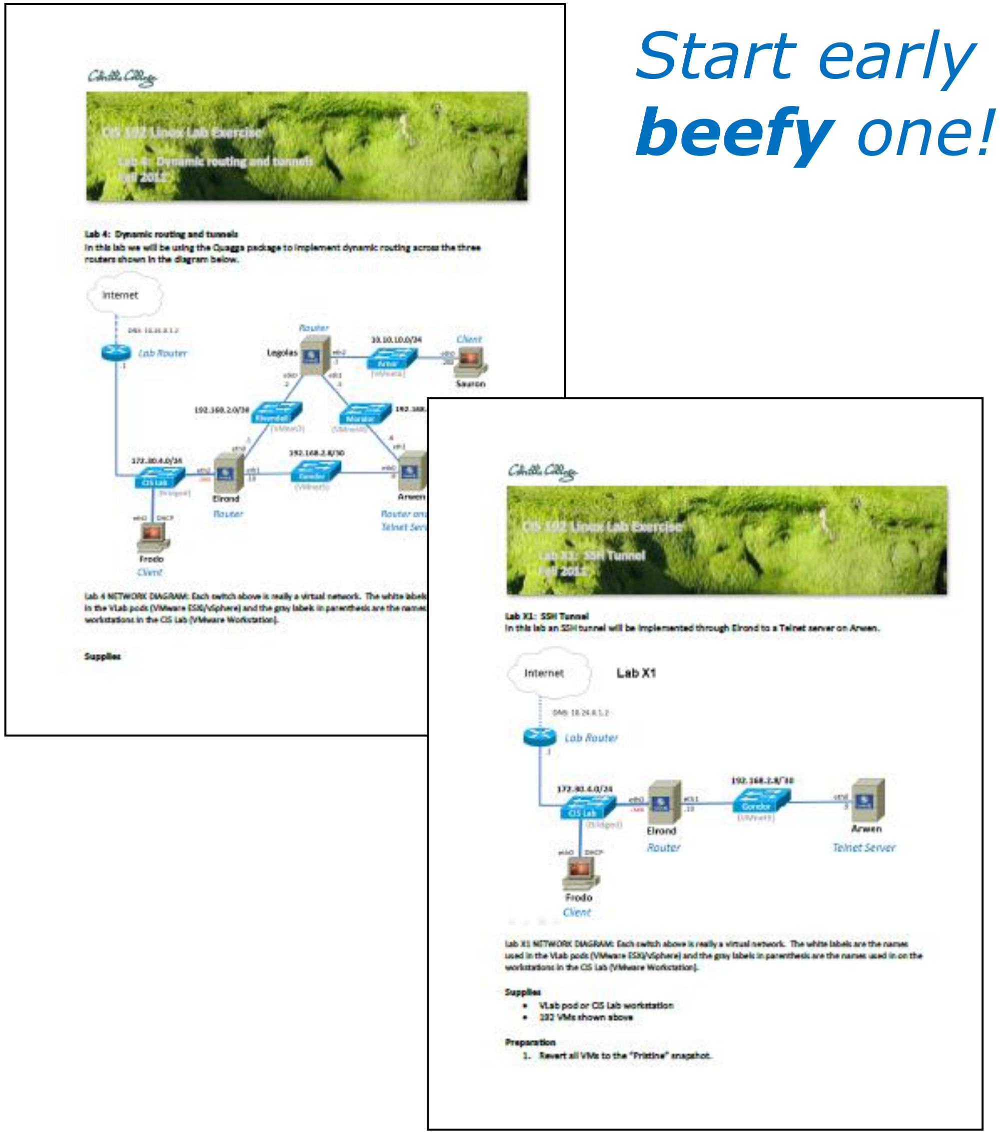
```
< snipped >
# 21 is registered to ftp, but also used by fsp
ftp             21/tcp
ftp             21/udp          fsp fspd
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp                          # SSH Remote Login Protocol
telnet          23/tcp
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 >
```

Not a Wrap Yet

Test Coming

Start early on Lab 4 … it’s a ***beefy*** *one!*

FYI, note extra credit Lab X1 has also been posted



Cabrillo College

CIS 192 Linux Lab Exercise

Lab 4: Dynamic routing and tunnels
Fall 2012

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.

Lab 4 NETWORK DIAGRAM: Each switch above is really a virtual network. The white labels in the VLab pods (VMware ESXi/vSphere) and the gray labels in parenthesis are the names workstations in the CIS Lab (VMware Workstation).

Supplies

Cabrillo College

CIS 192 Linux Lab Exercise

Lab X1: SSH Tunnel
Fall 2012

Lab X1: SSH Tunnel
In this lab an SSH tunnel will be implemented through Elrond to a Telnet server on Arwen.

Lab X1 NETWORK DIAGRAM: Each switch above is really a virtual network. The white labels are the names used in the VLab pods (VMware ESXi/vSphere) and the gray labels in parenthesis are the names used in on the workstations in the CIS Lab (VMware Workstation).

Supplies

- VLab pod or CIS Lab workstation
- 192 VMs shown above

Preparation

1. Revert all VMs to the "Pristine" snapshot.

New commands, tools and services:

```
iptables –L -–line-numbers
service ripd restart
service xinetd restart
service zebra restart
service ripd restart
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
```

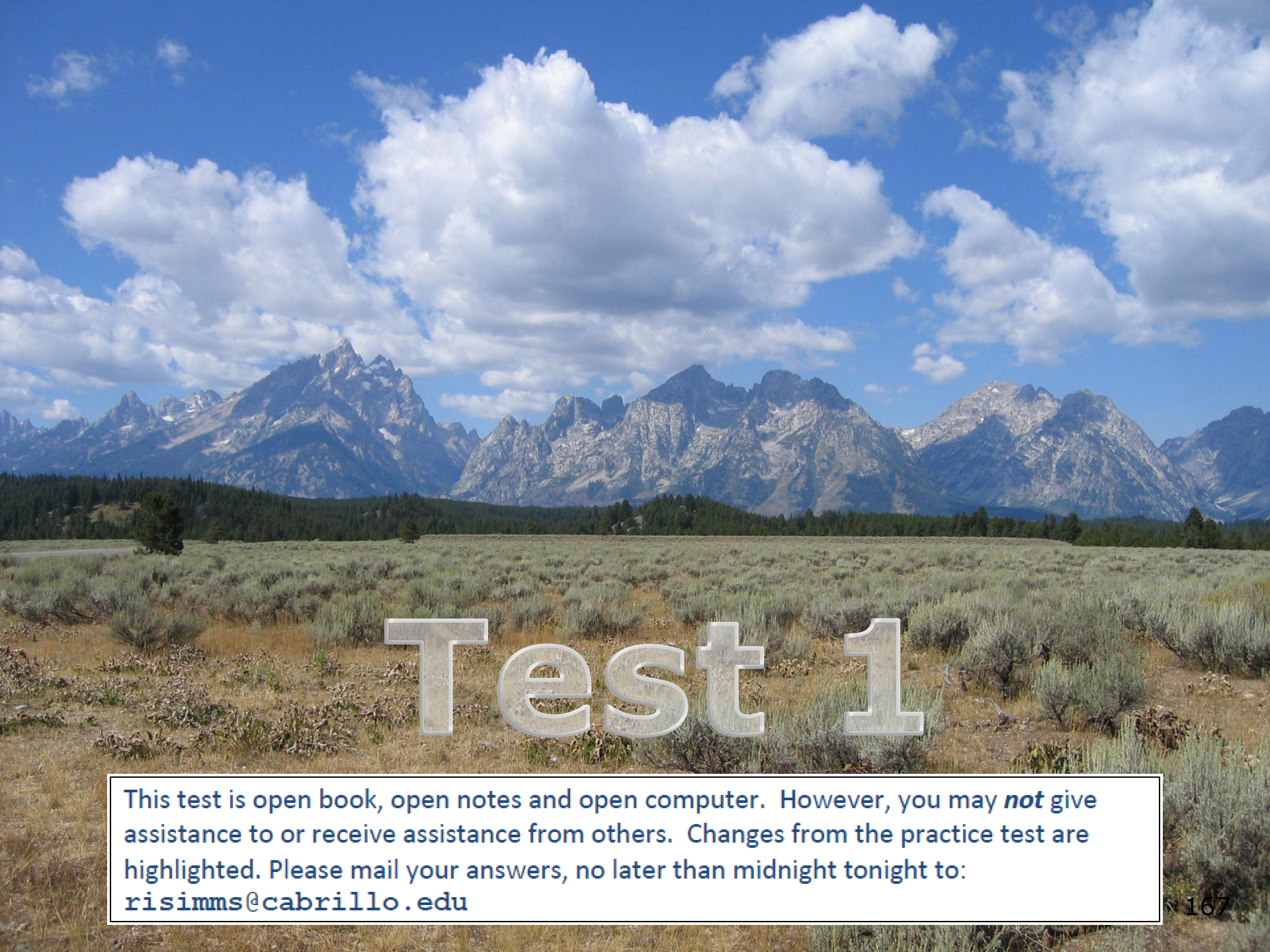
Next Class

Lab 4 due

Assignment: Check Calendar Page
http://simms-teach.com/cis192Acalendar.php

Quiz questions for next class:

- What command will flush the routing table cache?
- What Quagga daemons must be started to enable dynamic routing with RIPv2?
- What port is used for RIP advertisements?

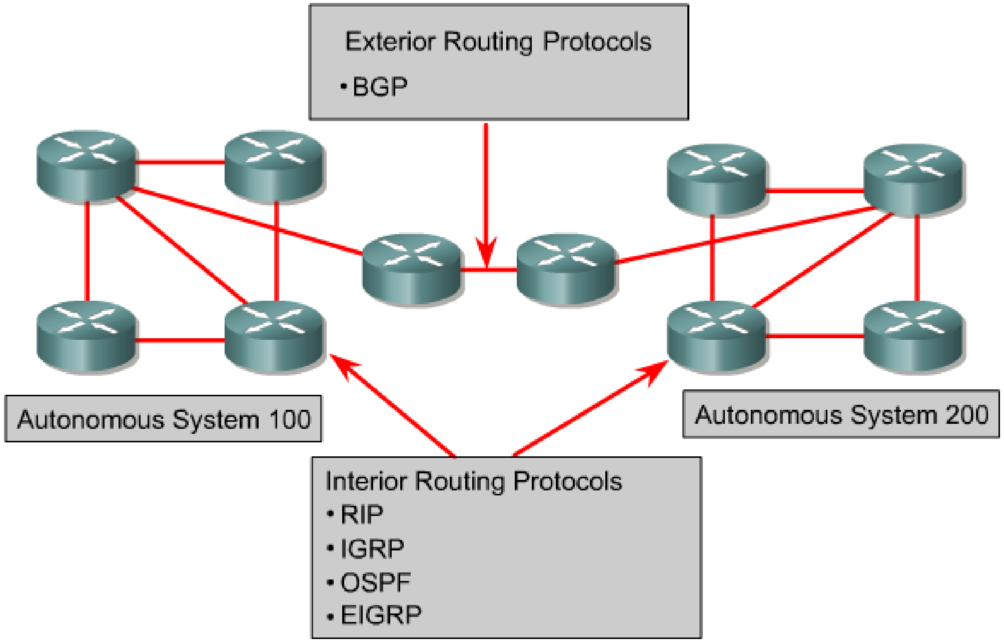


Test 1

This test is open book, open notes and open computer. However, you may ***not*** give assistance to or receive assistance from others. Changes from the practice test are highlighted. Please mail your answers, no later than midnight tonight to: `risimms@cabrillo.edu`

Backup

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)

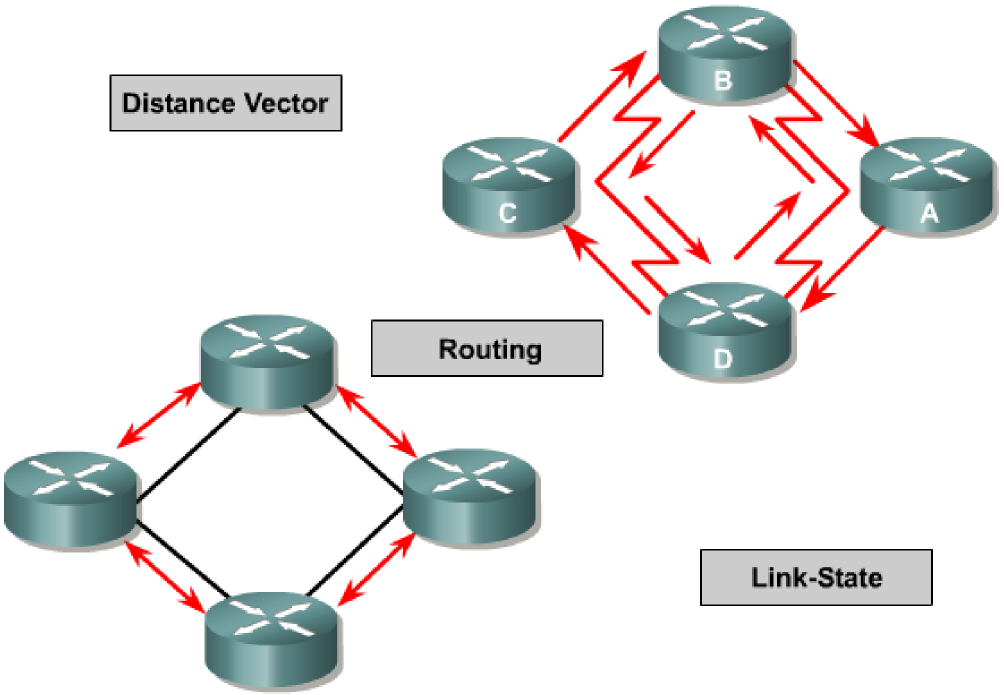
- IS-IS is an Open System Interconnection (OSI) routing protocol originally specified by International Organization for Standardization (ISO) 10589.
- It is a **link-state** routing protocol.

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

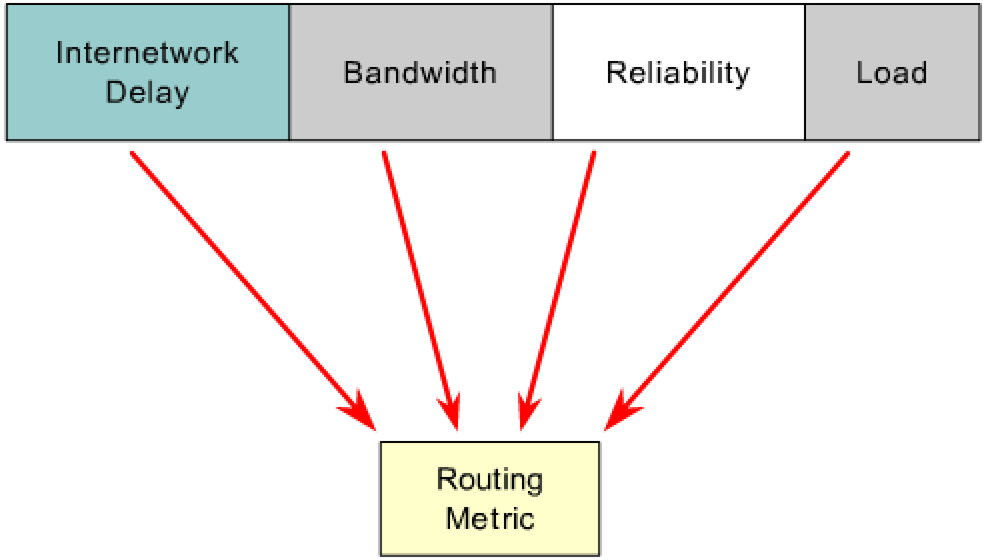
Routing

Link-State

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

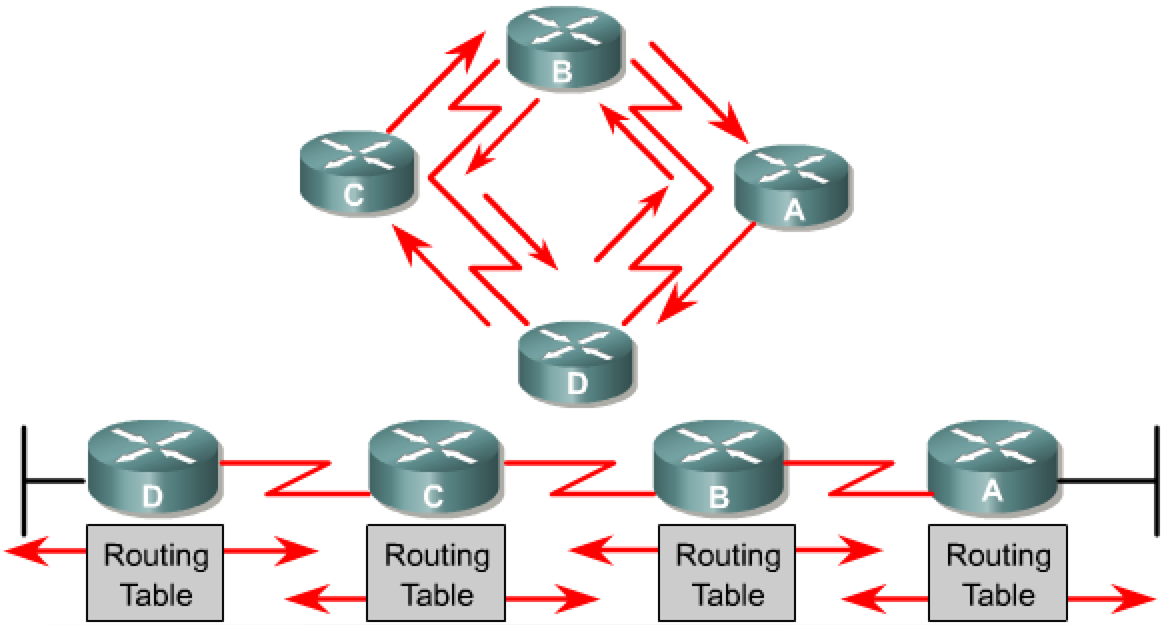
Path vector protocols (like BGP) are a class of distance vector protocols and not a link-state protocol

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.

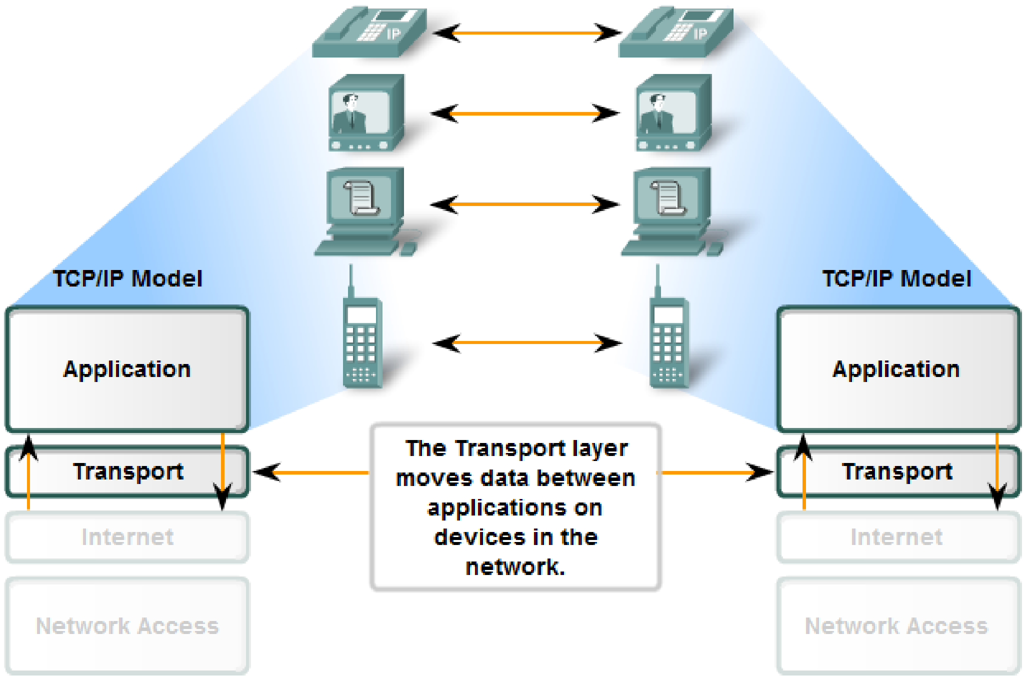
Router B adds a distance vector number (such as a number of hops), which increases the distance vector.

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.

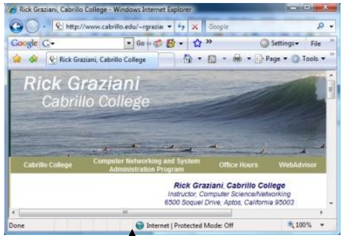
- “Routing by rumor”
- Each router receives a routing table from its directly connected neighbor routers.

Transport Layer

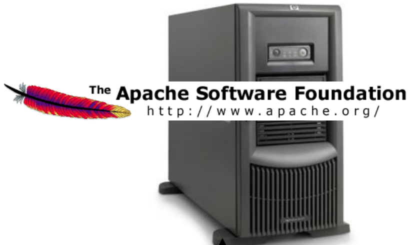


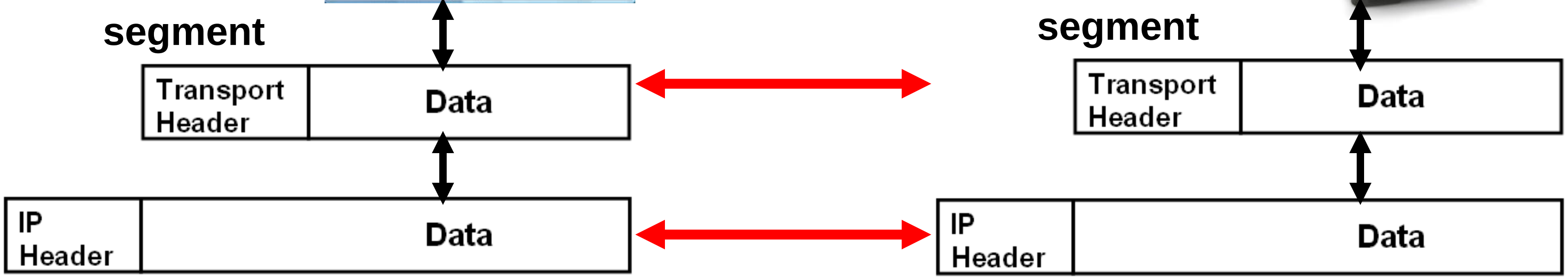
The Transport layer moves data between applications on devices in the network.

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


Rick Graziani, Cabrillo College - Windows Internet Explorer
http://www.cabrillo.edu/~rgrazia
Google
Go
Settings
File
Rick Graziani, Cabrillo College
Page
Tools
Rick Graziani
Cabrillo College
Cabrillo College
Computer Networking and System Administration Program
Office Hours
WebAdvisor
Done
Internet | Protected Mode: Off
100%

Rick Graziani, Cabrillo College
Instructor, Computer Science/Networking
6500 Soquel Drive, Aptos, California 95003



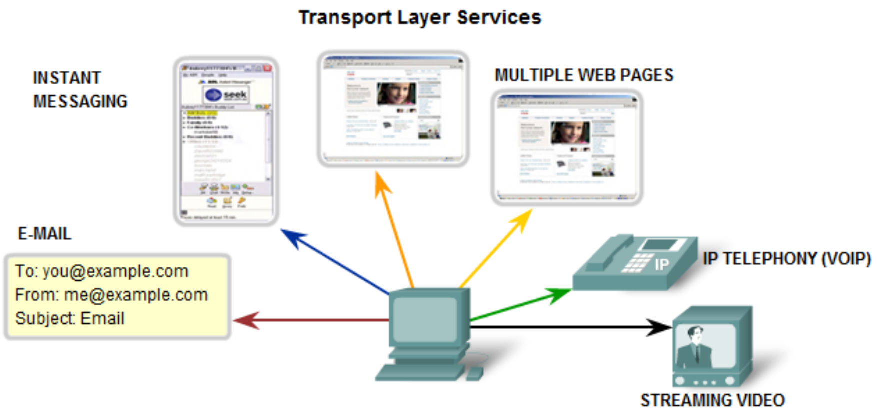


Transport Layer

- 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



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

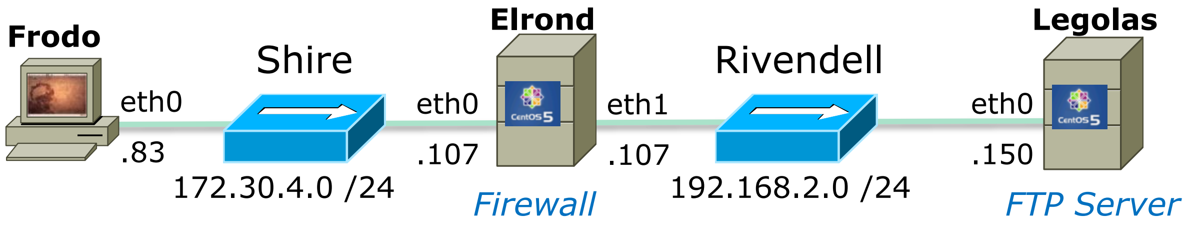
Continuing Communications

- 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

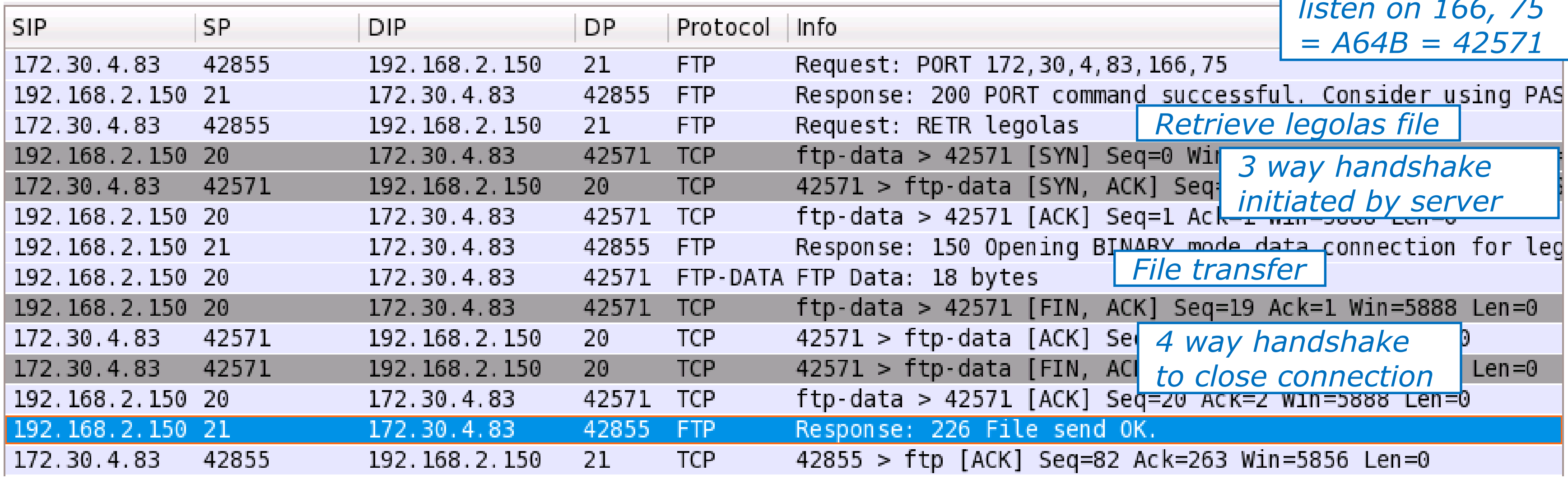
Client	Server
172.30.4.83	192.168.2.150
42855	21

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
172.30.4.83    42571  192.168.2.150  20     TCP      42571 > ftp-data [FIN, AC                   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

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
!
```

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

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

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 and the Firewall

Step 3 *Modify the firewall*

Firewall ports used for implementing RIPv2 with Quagga

UDP 520	*RIP advertisements*

Quagga and the Firewall

For Lab 4:

- The routers need UDP port 520 open to allow incoming RIP packets
- The routers need to allow packets to pass through (via packet forwarding)

For Lab X1:

- For the Telnet Server, TCP port 23 needs to be open for incoming Telnet connections

Quagga and the Firewall

Default firewall (in memory)

```
[root@celebrian ~]# iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
2    ACCEPT     icmp --  anywhere             anywhere
3    ACCEPT     all  --  anywhere             anywhere
4    ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
5    REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
[root@celebrian ~]#
```

- *There is no rule on the INPUT chain to accept incoming RIP packets (UDP port 520) so they will be rejected.*
- *All packets going through the FORWARD chain get rejected.*

Quagga and the Firewall

Default firewall (in configuration file)

```
[root@celebrian ~]# cat /etc/sysconfig/iptables
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
[root@celebrian ~]#
```

- *There is no rule on the INPUT chain to accept incoming RIP packets (UDP port 520) so they will be rejected.*
- *All packets going through the FORWARD chain get rejected.*

Quagga and the Firewall

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

Delete the first rule on the FORWARD chain

```
[root@celebrian ~]# iptables -I INPUT 4 -p udp -m udp --dport 520 -j ACCEPT
```

Insert a rule above rule 4 on the INPUT chain to accept incoming packets to UDP port 520

```
[root@celebrian ~]# iptables -I INPUT n -p tcp -m state --state NEW -m tcp --
dport 23 -j ACCEPT
```

(all on one line)

Insert a rule above rule 5 on the INPUT chain to accept incoming packets to TCP port 23

```
[root@celebrian ~]# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[  OK  ]
```

Save the rules in memory to the configuration file

Modifying the Firewall
(Centos)

Modified firewall (in memory)

RIP and Telnet ports open

```
[root@celebrian ~]# iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
2    ACCEPT     udp  --  anywhere             anywhere            udp dpt:router
3    ACCEPT     icmp --  anywhere             anywhere
4    ACCEPT     all  --  anywhere             anywhere
5    ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:telnet
6    ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
7    REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited

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

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
[root@celebrian ~]#
```

No filtering now on forwarded packets

Quagga and the Firewall

Modified firewall (in configuration file)

```
[root@celebrian ~]# cat /etc/sysconfig/iptables
# Generated by iptables-save v1.4.7 on Tue Nov 15 00:41:40 2011
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [11:1740]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p udp -m udp --dport 520 -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 23 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Tue Nov 15 00:41:40 2011
[root@celebrian ~]#
```

No filtering now on forwarded packets

RIP and Telnet ports open

Quagga and the Firewall

Modified firewall

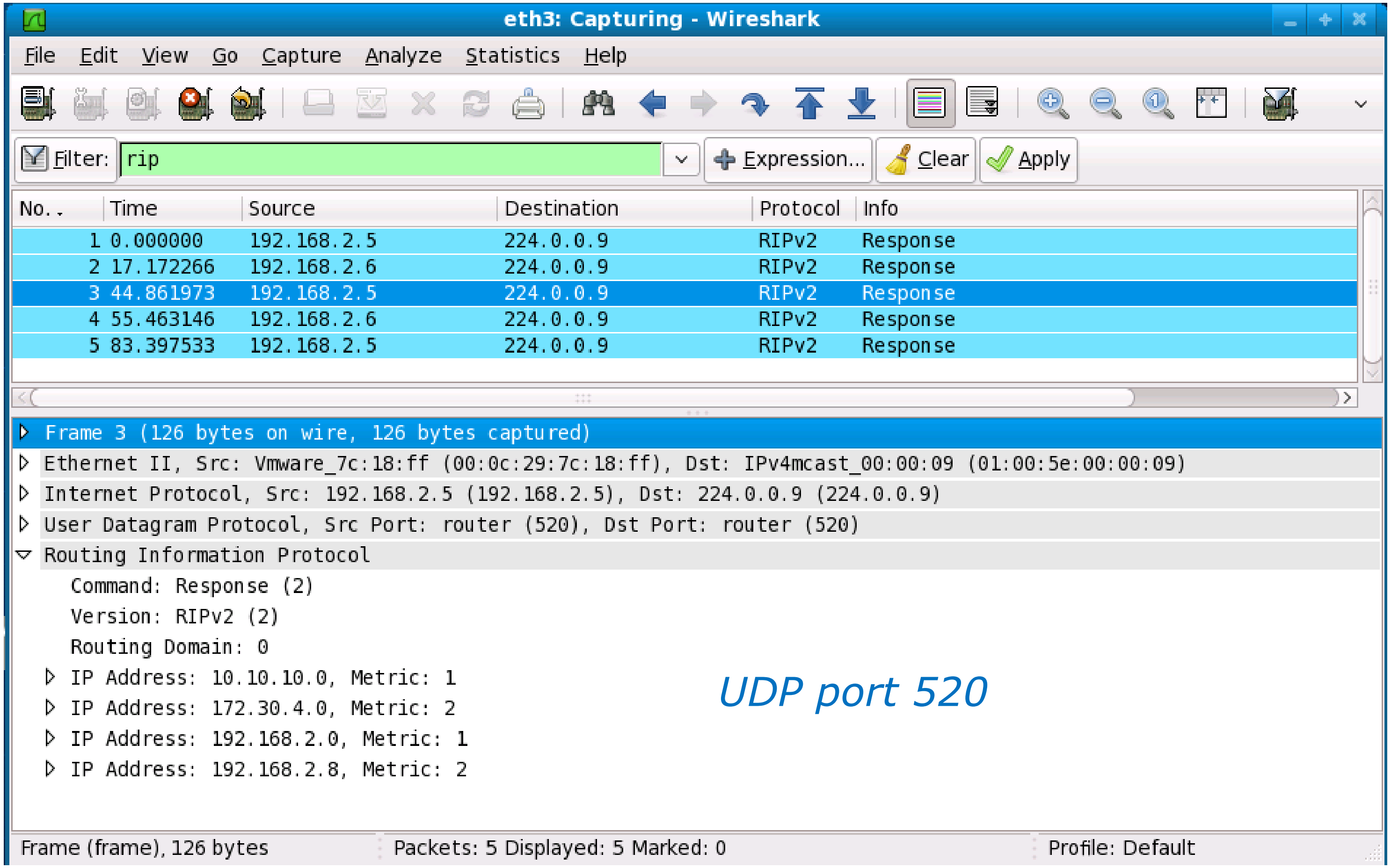
```
[root@arwen ~]# cat /etc/sysconfig/iptables
# Generated by iptables-save v1.3.5 on Thu Feb 26 08:22:29 2009
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [946:71747]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -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

Quagga and the Firewall

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
```

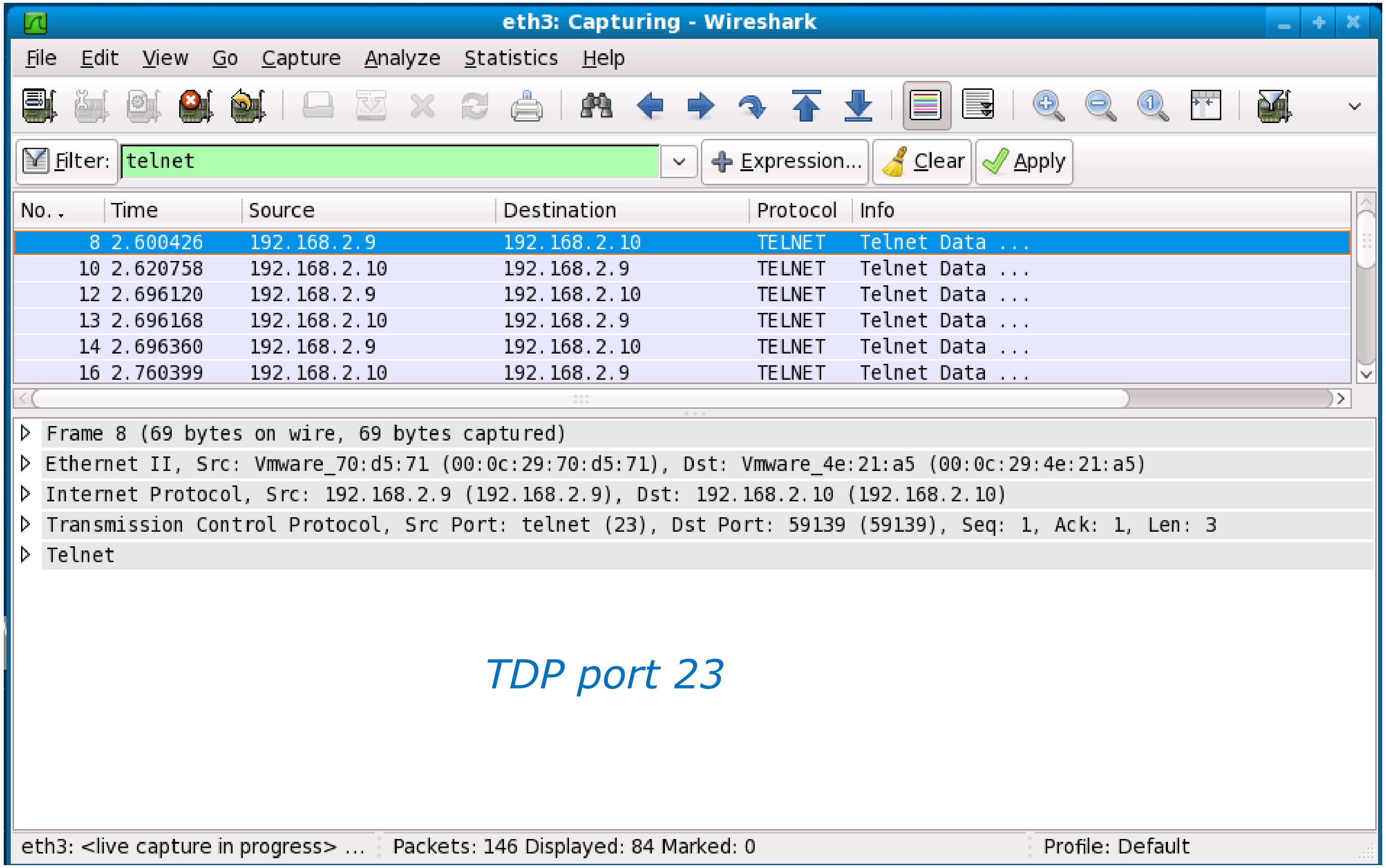
```
▷ Frame 3 (126 bytes on wire, 126 bytes captured)
▷ Ethernet II, Src: Vmware_7c:18:ff (00:0c:29:7c:18:ff), Dst: IPv4mcast_00:00:09 (01:00:5e:00:00:09)
▷ Internet Protocol, Src: 192.168.2.5 (192.168.2.5), Dst: 224.0.0.9 (224.0.0.9)
▷ User Datagram Protocol, Src Port: router (520), Dst Port: router (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


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

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
```

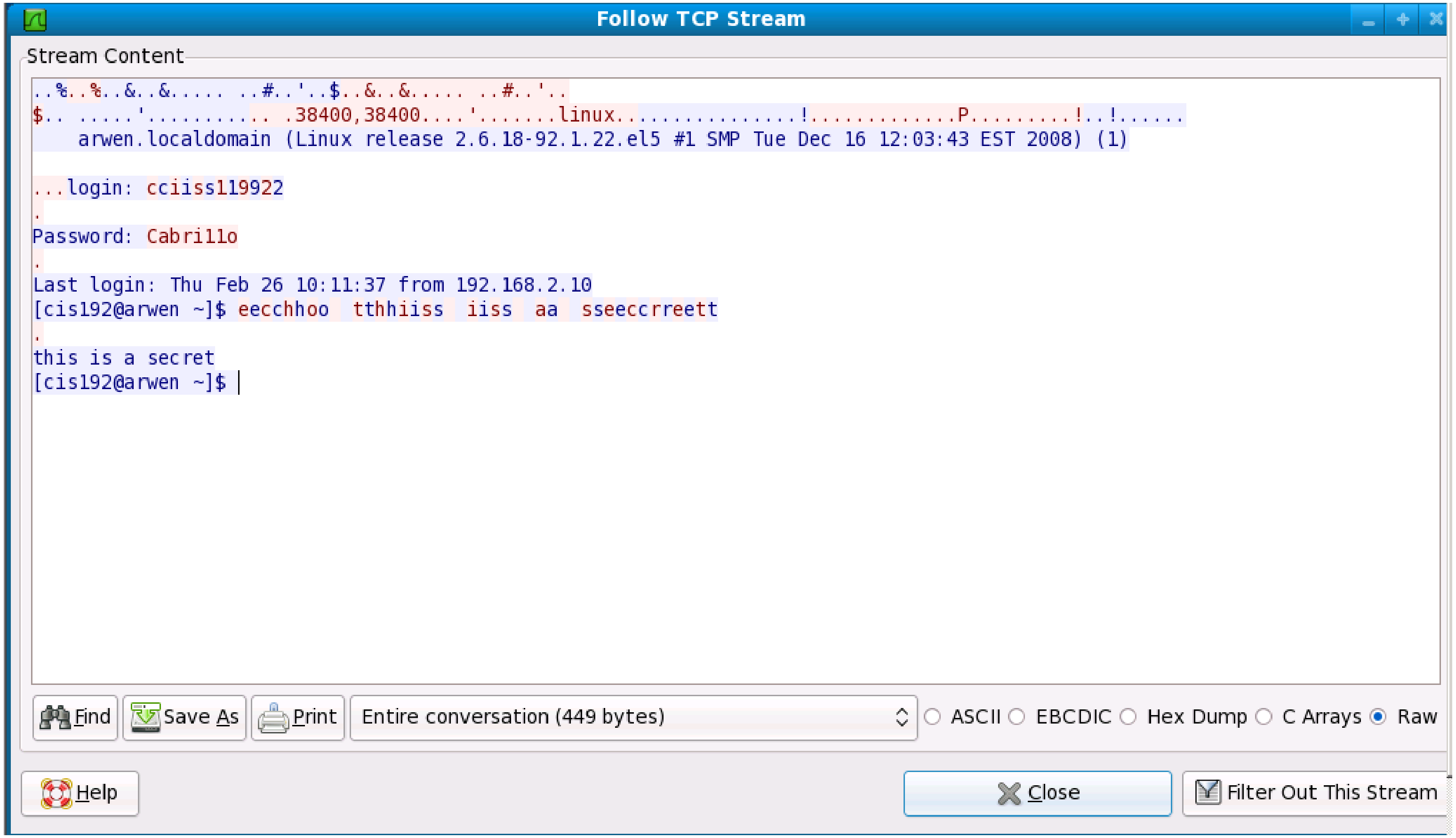
TDP port 23

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

/etc/quagga/zebra.conf and /etc/quagga/ripd.conf

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
```

/etc/quagga/zebra.conf and /etc/quagga/ripd.conf

```
[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