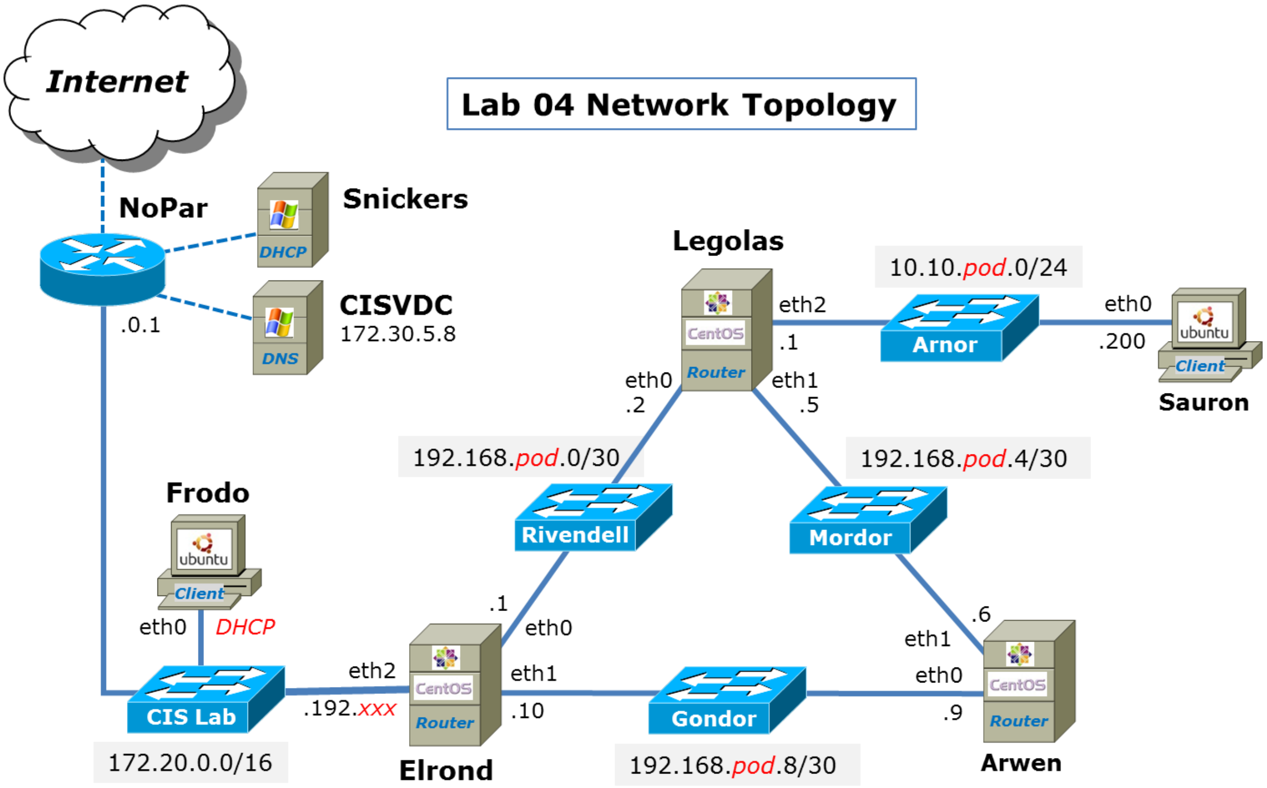
Cabrillo College

CIS 192 Linux Lab Exercise

Lab 4: Dynamic routing
Spring 2013

Lab 4: Dynamic routing
In this lab we will be using the Quagga package to implement dynamic routing across the three routers shown in the diagram below.



pod=your pod number, *xxx*=one of your assigned IP addresses

Supplies

- VLab pod
- 192 VMs shown above

Forum

Use the forum to ask questions, collaborate, post tips and any lessons learned when you have finished. Forum is at: http://oslab.cabrillo.edu/forum/

Background

Quagga is a GPL licensed routing software suite that implements OSPF, RIP and BGP routing protocols on Linux. See http://www.quagga.net/ for details.

Preparation

- □ Revert all VMs to the "Pristine" snapshot.
- □ On Opus, make a copy of the *lab04* report template file in */home/cis192/depot* in your home directory. Edit the header of this file with your own information and record all the information requested.
- □ Make a network map and crib sheet. The map should include networks and IP addresses on each interface. For the CIS Lab network, only use your assigned IP addresses in: http://simms-teach.com/docs/cis192/Pod-Assignments-192-sp13.pdf. The crib sheet content is up to you. Try and make it a useful reference document you can use for this lab and in the future.

Part I – Install dynamic routing software

On Elrond, Legolas and Arwen, install the telnet client and Quagga routing software as follows:

- □ Cable one of their interfaces to the CIS Lab network and use **`dhclient -v eth`***n* to get an IP address and Internet access.
- □ Install the software with: **`yum install telnet quagga`**
- □ Release the IP addresses with **`dhclient -v -r`** on each of the three VMs.

Part II – Cable and join systems to their networks

- □ Cable all VMs to the networks shown in the diagram above.

Part III – Configure permanent network settings

Use the diagram above to configure each host. Example configurations are shown in the Appendix.

- □ Configure permanent addresses on all interfaces.
- □ Add a default gateway to Elrond (towards the Internet).
- □ Add a default gateway to Sauron (towards the Internet).
- □ Add static routes on Frodo to reach each network on the diagram. To reduce the number of static routes, supernet the routes to Rivendell, Mordor and Gondor into a single route.
- □ Test what you have done so far using the ping command. You should be able to ping immediate neighbors. Because the routing tables are incomplete it is not yet possible to ping non-connected networks.

Part IV – Punch some holes in the firewalls

- □ For the Elrond, Legolas and Arwen routers, the firewalls need to be permanently modified to allow incoming RIP traffic (UDP port 520) and allow packets to be forwarded:

```
iptables -I INPUT 4 -p udp -m udp --dport 520 -j ACCEPT
iptables -D FORWARD 1
service iptables save
```

Part V – Enable packet forwarding

- □ Enable permanent packet forwarding on Elrond, Legolas and Arwen so they can function as routers.

Elrond, Legolas and Arwen

/etc/sysctl.conf

< snipped >

```
net.ipv4.ip_forward = 1
```

< snipped >

Then use:

```
service network restart
```

or

```
sysctl -p
```

Part VI – Dynamic routing table updates

Rather than use manually configured static routes, in this lab we will use dynamic routing. Quagga will be used to implement the RIPv2 protocol across Elrond, Legolas and Arwen.

- □ Edit the Quagga configuration files on each router:

Elrond

/etc/quagga/zebra.conf

```
hostname pxx-elrond
!
password quagga
enable password quagga
!
log file /var/log/quagga/zebra.log
```

/etc/quagga/ripd.conf

```
hostname pxx-elrond
log file /var/log/quagga/ripd.log
!
router rip
 network eth0
 network eth1
 redistribute connected
 default-information originate
!
line vty
  password quagga
!
```

Legolas

/etc/quagga/zebra.conf

```
hostname pxx-legolas
!
password quagga
enable password quagga
!
log file /var/log/quagga/zebra.log
```

/etc/quagga/ripd.conf

```
hostname pxx-legolas
log file /var/log/quagga/ripd.log
!
router rip
 network eth0
 network eth1
 redistribute connected
!
line vty
  password quagga
!
```

Arwen

/etc/quagga/zebra.conf

```
hostname pxx-arwen
!
password quagga
enable password quagga
!
log file /var/log/quagga/zebra.log
```

/etc/quagga/ripd.conf

```
hostname pxx-arwen
log file /var/log/quagga/ripd.log
!
router rip
 network eth0
 network eth1
 network eth2
 redistribute connected
!
line vty
  password quagga
!
```

□ Start the daemons:

```
service zebra start
service ripd start
```

- □ Configure the daemons to run at system startup:

```
chkconfig zebra on
chkconfig ripd on
```

- □ Check the routing tables and confirm each router has routes to each network as shown on the map.

- □ Test and make sure all the VMs can ping each other now.
  Note: On newer distributions you may need to disable return path filtering so Frodo or Sauron can ping all router interfaces. This filtering will drop the echo reply if the return path differs from the path taken by the incoming ping request. To disable return path filtering, do the following on Elrond, Legolas and Arwen:

```
echo 0 > /proc/sys/net/ipv4/conf/eth0/rp_filter
echo 0 > /proc/sys/net/ipv4/conf/eth1/rp_filter
```

  To disable it on startup, edit */etc/sysctl.conf*:

```
# Controls source route verification
net.ipv4.conf.default.rp_filter = 0
```

- □ Record the following information to your lab report:

  - □ **route -n** output for Frodo
  - □ **route -n** output for Sauron
  - □ **route -n** output for Elrond
  - □ **route -n** output for Legolas
  - □ **route -n** output for Arwen
  - □ Successful **ping -R 10.10.*pod*.200 -c2** output from Frodo to Sauron

To turn in

Check your work for completeness then submit as many times as you wish up until the due date deadline. Remember, **late work is not accepted**, so start early, plan ahead for things to go wrong and use the forum to ask questions.

1) Submit your lab report on Opus using:

```
cp  lab04  /home/rsimms/turnin/cis192/lab04.$LOGNAME
```

2) Email your map and crib sheet to **risimms@cabrillo.edu**

Grading rubric (30 points)

4 points for emailing map & crib sheet
2 points for 100% complete header in lab report
4 points for correct route -n output on Frodo
4 points for correct route -n output on Sauron
4 points for correct route -n output on Elrond

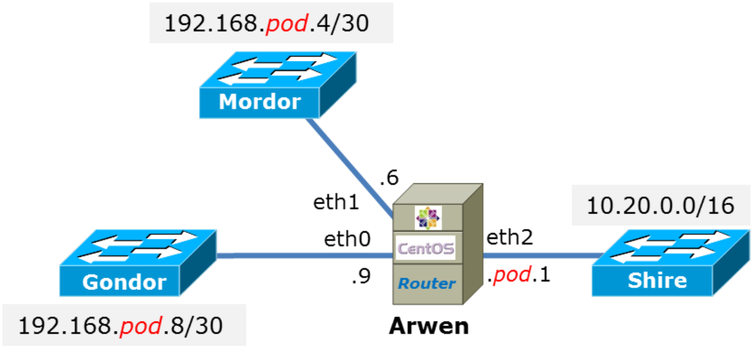
4 points for correct route -n output on Legolas
4 points for correct route -n output on Arwen
4 points for successful end-to-end ping

Extra Credit (1 point each)

1. Use the Quagga to view the routing table. On Arwen, use **telnet localhost 2601** (or **vtysh**) and **enable** to start. Record output from **show ip route** in your lab report.

2. Use Quagga to debug rip. On Arwen, use **telnet localhost 2602** (or **vtysh**) and **enable** to start. Use **debug rip packet** to start debugging and then record the exchange of one or more route packets from the log file into your lab report.

3. Connect Arwen's eth2 interface to the shire-27 network and see if you can ping Sauron-27 at 10.10.27.200 from your pod's Sauron.



Record the **route -n** output from Arwen and **ping -R 10.10.27.200 -c2** from Sauron in your lab report.

Appendix

Permanent network settings for each system:

Frodo

/etc/network/interfaces

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

up route add -net 192.168.pod.0/28 gw 172.20.192.xxx
up route add -net 10.10.pod.0/24 gw 172.20.192.xxx
```

Then use:

```
/etc/init.d/networking restart
```

Sauron

/etc/network/interfaces

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 10.10.pod.200
netmask 255.255.255.0

gateway 10.10.pod.1
```

Then use:

```
/etc/init.d/networking restart
```

Elrond

/etc/sysconfig/network

```
NETWORKING=yes
HOSTNAME=pxx-elrond.cislab.net
GATEWAY=172.20.0.1
```

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

```
DEVICE="eth0"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=192.168.pod.1
NETMASK=255.255.255.252
```

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

```
DEVICE="eth1"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=192.168.pod.10
NETMASK=255.255.255.252
```

/etc/sysconfig/network-scripts/ifcfg-eth2

```
DEVICE="eth2"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=172.20.192.xxx
NETMASK=255.255.0.0
```

Then use:

```
service network restart
```

Legolas

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

```
DEVICE="eth0"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=192.168.pod.2
NETMASK=255.255.255.252
```

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

```
DEVICE="eth1"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=192.168.pod.5
NETMASK=255.255.255.252
```

/etc/sysconfig/network-scripts/ifcfg-eth2

```
DEVICE="eth2"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=10.10.pod.1
NETMASK=255.255.255.0
```

Then use:

```
service network restart
```

Arwen

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

```
DEVICE="eth0"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=192.168.pod.9
NETMASK=255.255.255.252
```

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

```
DEVICE="eth1"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=192.168.pod.6
NETMASK=255.255.255.252
```

Then use:

```
service network restart
```