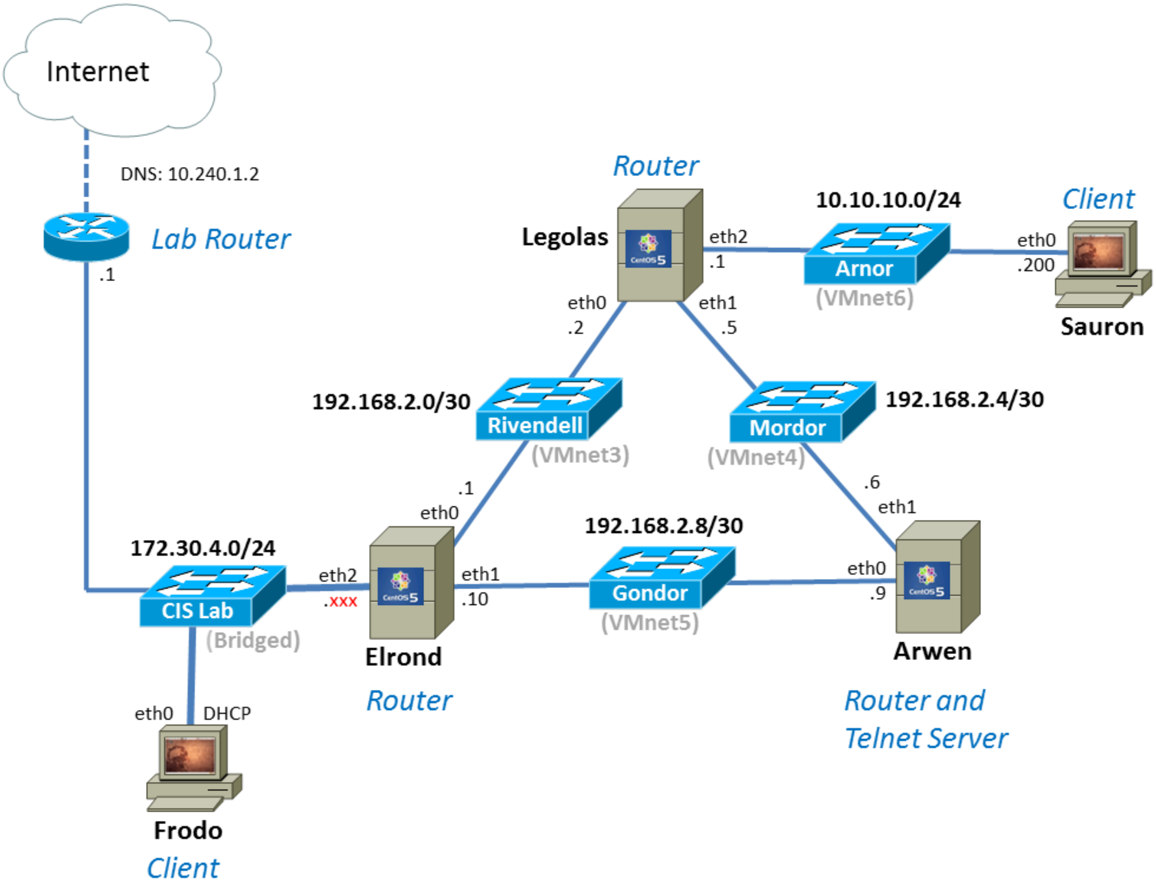
Cabrillo College

# CIS 192 Linux Lab Exercise

## Lab 4: Dynamic routing and tunnels
## Fall 2011

### 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 are the names used in the VLab pods (VMware ESXi/vSphere) and the gray labels in parenthesis are the names used 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.
2. 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.

**Forum**

Use the forum to ask questions, collaborate, post tips and any lessons learned when you have finished. Forum is at: http://opus.cabrillo.edu/forum/viewforum.php?f=39

**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.

**Part I – Go get and install some new hardware and software**

1. Revert all VMs used in this lab back to their snapshots.
2. Add a third NIC to **Elrond** and **Legolas**. These systems will be directly attached to three networks so three interfaces are required.
3. On the CentOS VMs, **Elrond, Legolas and Arwen,** install the telnet client and Quagga routing software. Do the following steps:
   a. Cable their eth0 Ethernet interface to the 172.30.4.0/24 CIS Lab network.
   b. Power on the three VMs.
   c. Use **`dhclient eth0`** on each to temporarily join the CIS Lab network for Internet access.
   d. Install the new software with:
      **`yum install telnet quagga`**
4. Release the IP address with **`dhclient -r eth0`** on each of the three VMs.

**Part II – Cable equipment**

1. Cable all VMs to the networks shown in the diagram above. This includes re-cabling the eth0 interfaces on Elrond, Legolas and Arwen. **Watch out: on some Elrond and Legolas VMs, the new third NIC may not be eth2!** Compare the MAC addresses in the **VM settings** with the MAC addresses on interfaces eth0, eth1 and eth2 to verify how interfaces were assigned to NICs. Use **ifconfig** command or view the **/etc/udev/rules.d/70-persistent-net.rules** file.
2. Power on the remaining VMs in the diagram. If you run out of memory on VMware Workstation, modify the Host Preferences (Memory Tab) to *Allow most virtual machine memory to be swapped.*

**Part III – Configure permanent network settings**

Use the diagram above to configure each host.

1. Configure Frodo as follows. Note, Frodo needs some static routes to get to the Mordor, Rivendell, Arnor and Gondor networks. The static routes needed for the three 192.168.2.x networks can be supernetted into a single route.

**Frodo**

/etc/network/interfaces

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

up route add -net 192.168.2.0/24 gw 172.30.4.xxx
up route add -net 10.10.10.0/24 gw 172.30.4.xxx
```

(Use Elrond’s eth2 IP address for .xxx)

Then use:

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

2. Configure Sauron as follows. Note, Sauron needs a default route to send packets towards the other networks.

**Sauron**

/etc/network/interfaces

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 10.10.10.200
netmask 255.255.255.0

gateway 10.10.10.1
```

Then use:

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

3. Configure Elrond as follows.

**Elrond**

/etc/sysconfig/network

```
NETWORKING=yes
HOSTNAME=elrond.localdomain
GATEWAY=172.30.4.1
```

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

```
DEVICE="eth0"
ONBOOT="yes"
BOOTPROTO="static"
```

```
IPADDR=192.168.2.1
NETMASK=255.255.255.252
```

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

```
DEVICE="eth1"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=192.168.2.10
NETMASK=255.255.255.252
```

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

```
DEVICE="eth2"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=172.30.4.xxx
NETMASK=255.255.255.0
```

(Use http://simms-teach.com/docs/static-ip-addrs.pdf for .xxx)

Then use:

```
service network restart
```

4. Configure Legolas as follows.

**Legolas**

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

```
DEVICE="eth0"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=192.168.2.2
NETMASK=255.255.255.252
```

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

```
DEVICE="eth1"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=192.168.2.5
NETMASK=255.255.255.252
```

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

```
DEVICE="eth2"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=10.10.10.1
NETMASK=255.255.255.0
```

Then use:

```
service network restart
```

5. Configure Arwen as follows.

**Arwen**

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

```
DEVICE="eth0"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=192.168.2.9
NETMASK=255.255.255.252
```

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

```
DEVICE="eth1"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR=192.168.2.6
NETMASK=255.255.255.252
```

Then use:

```
service network restart
```

6. 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

1. For **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.

1. Make this change on each router:

**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.

1. Edit the two Quagga configuration files.

**Elrond**

/etc/quagga/zebra.conf

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

/etc/quagga/ripd.conf

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

**Legolas**

/etc/quagga/zebra.conf

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

/etc/quagga/ripd.conf

```
hostname legolas
log file /var/log/quagga/ripd.log
!
router rip
```

```
 network eth0
 network eth1
 redistribute connected
!
line vty
!
```

**Arwen**

/etc/quagga/zebra.conf

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

/etc/quagga/ripd.conf

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

2. Set ownership of the configuration files with:

```
cd /etc/quagga/
chown quagga:quagga ripd.conf zebra.conf
```

3. Modify SELinux context to allow writes to configuration files:

```
chcon -v --type=zebra_conf_t ripd.conf zebra.conf
```

4. Start the daemons:

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

5. Configure the daemons to run at system startup:

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

6. After all three routers have Quagga installed, configured and running, check the routing tables. On Legolas you should see:

```
[root@legolas ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
Iface
```

```
192.168.2.8     192.168.2.1     255.255.255.252 UG    2      0         0 eth0
192.168.2.0     0.0.0.0         255.255.255.252 U     0      0         0 eth0
192.168.2.4     0.0.0.0         255.255.255.252 U     0      0         0 eth1
172.30.4.0      192.168.2.1     255.255.255.0   UG    2      0         0 eth0
10.10.10.0      0.0.0.0         255.255.255.0   U     0      0         0 eth2
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0         0 eth2
0.0.0.0         192.168.2.1     0.0.0.0         UG    0      0         0 eth0
[root@legolas ~]#
```

7. Use Zebra to view the routing table. On Legolas you should see:

```
[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> enable
legolas# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB
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, eth2
R>* 172.30.4.0/24 [120/2] via 192.168.2.1, eth0, 09:37:17
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, 09:37:17
```

8. 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**

9. Copy 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** 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.

Submit your lab report on Opus using:

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

**Grading rubric (30 points)**

3 points for correct submittal of lab report into turnin directory
3 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