COE865 Advanced Computer Networks

Lab Manual

Professor: Dr. Muhammad Jaseemuddin
TA: Niusha Sabri Kadijani

Winter 2024

Dept. of Electrical, Computer and Biomedical Engineering

© Toronto Metropolitan University, Toronto, Canada

Lab Schedule

Labs	Title	Due Weeks	Marks
Lab 1	Design and Configuration of a single domain network	Week 4 (Jan 29)	5
Lab 2	Design and Evaluation of a RIP Network	Week 6 (Feb 12)	5
Lab 3	Design and Evaluation of an OSPF Network	Week 8 (Mar 4)	5
Lab 4	Design and Evaluation of a BGP network	Week 10 (Mar 18)	5

Pre-Lab

The objective of pre-lab is to get you familiarized with the lab hardware and software tools, and to review some Linux commands that are used in the labs.

1. Description of the hardware

The Computer Networks Lab consists of a set of PCs (which work as both routers and hosts), a switch, and Ethernet hubs. Each computer has three Ethernet connections (eth0, eth1, and eth2). The three numbers on your computer box correspond to the numbers on the patch panel located in the middle of the lab. You can find the number related to your Ethernet port, and connect it to the Ethernet port of another PC on the patch through a cross connect cable. The network you configure on your bench is isolated from the campus network. Find below the map of routers in the lab to the lab topology:

R1: Guelph, R2: Finch, R3: Whitby, R4: Malton

R5: Brampton, R6: Bloor, R7: Kipling, R8: Dixie

R9: Danforth, R10: Caledon, R11: Acton, R12: Keele

R13: Eglinton, R14: Clarkson, R15: TheEx, R16: Appleby

R17: Ajax, R18: York, R19: Oshawa, R20: Bronte

2. How to log in

Use your "ee" account to log in.
In this lab, we use a virtual system under Linux for network setup. On the *Application* menu, choose *ComLab*. This gives you two options, one of them is used to handle your virtual disk and folders. Choose '*Comlab Virtual Linux*' to start the Linux system.
Now you need to logon to the virtual system. Use the following information.
Username: *bob*
Password: *letbobin*
Here, you can open a new terminal and start typing the commands. If you needed to be 'root' in order to be able to change some network parameters, you can change the user to '*root*' anytime. Here is what you type in the command line:

```
$su
$password: letrootin
```

! Remember that every time you log out of the virtual system, all your files get deleted. To avoid losing your files, you need to save everything under your own *home directory* on the host machine. This is little tricky since the host machine and virtual machine need a way to communicate with each other. *Shared Folders* provides the capability to achieve this. To enable sharing, do the following on the running virtual machine:

1. From the *ComLab* menu navigate to: Virtual Machine > Virtual Machine Settings > Options > Shared Folders.

2. Check “Enabled” option.
3. “Add” a shared folder and choose/create it within the `Documents/coe865/` folder of the host’s file system by clicking “Browse”.
4. Fill in the name as requested by the Add dialog. Note: this name for virtual machine use only.
5. To access the folder from the virtual machine, navigate to path: `/mnt/hgfs/<name_of_shared_folder>/`
6. To access the folder on the host machine, navigate to path: `~/Documents/coe865/<sharedFolder>/`

Once you have access to the shared folder, you may use it to transfer files between the host and virtual machine systems. Multiple shared folders may also be added.

To transfer files on the fly, you may also drag-and-drop them from the host to the virtual machine and vice-versa. Another option is to use USB key drives for the transfers.

3. An overview of Linux commands

This section is to remind you of some Linux commands that are frequently used in this lab.

Command	Usage
Man	Help manual
Pwd	Print Working Directory
Ls	List information about file(s)
Cd	Change Directory
Rm	Remove files
Cp	Copy one or more files to another location
Mv	Process status
Clear	Clear terminal screen
Locate	Find files
Ps	Process status
Kill	Stop a process from running
Chmod	Change access permissions
Find	Search for files that meet a given criteria
Cat	Display the contents of a file
Su	Substitute user identity
Ethtool	Ethernet card settings
Vi	Text Editor
Rpm	Packet Manager, querying and verifying packages

Frequently used network commands:

Ifconfig	ifconfig <eth1> <10.1.1.2> netmask <255.255.255.0> up
Route	To show or to change IP routing table

For more information about these commands and/or to find the more complete set of commands see this links:

1. Internet Lab Manual: Chapter 1 (Sections 2 and 3), Chapter 2 (Sections 2 and 6) and Chapter 3 (Sections 4 and 5). This document is very useful to get familiar with Linux devices and network commands, and some network analyzing tools: http://www.tcpiplab.net/.
2. Linux Bash Commands: http://www.ss64.com/bash/index.html

Lab1: Design and Configuration of a Single Domain Network

Lab1 (a): Networking Basics

1. Objective

The objective of this lab is to get you to familiarize with the basic configurations of Linux PCs as routers and setting up network with them.

2. Router configuration

1. Configure the interface address, subnet mask (24-bit) and broadcast address for each interface of your router using */sbin/ifconfig* command. The broadcast address should be set to the subnet broadcast address. For example, the broadcast address for the subnet 10.1.1.0 would be 10.1.1.255.
2. Check your configuration by running the commands */sbin/ifconfig* and *netstat –i*. "/sbin/ifconfig -a to show interface status"
3. Restart the virtual system. Are the interfaces set to the prior configuration values?
4. Create an executable file to save your interface address configuration in your own home directory. You can copy this file into virtual Linux environment every time you logon, and then run the file to setup the interface addresses. This will save in your time. Here is how to do it:

   Create a file and name it "IPaddr.exec":

   ```
   $gedit IPaddr.exec
   ```

   IPaddr.exec:

   ```
   /sbin/ifconfig eth1 10.1.1.20 netmask 255.255.255.0 up
   /sbin/ifconfig eth2 10.1.1.30 netmask 255.255.255.0 up
   ```

   Save this file and allow executable privileges:

   ```
   $chmod +x IPaddr.exec
   ```

   Run the file with the command:

   ```
   $./IPaddr.exec
   ```

   Now check the eth1 and eth2 configuration with *ifconfig* command.
5. Write '1' to the file `/proc/sys/net/ipv4/ip_forward` to turn on IP forwarding. You can modify the file as follows:

   ```
   $echo 1 > /proc/sys/net/ipv4/ip_forward
   ```
6. To bring those network interfaces that are not used down use the following command (e.g. to bring eth0 down):

   "$/sbin/ifconfig eth0 down"

   To bring interfaces up:

"$/sbin/ifconfig eth0 up"

7. For more information about commands use the "man" command (e.g. to know more about "route" command):

```
$man route
```

3. Routing table configuration using static routes

Note: The following steps use router R4 as an example. Refer to Figure 1 for the IP addresses that have to be used to configure your specific router.

1. Show the routing table using command */sbin/route –n.*
2. Remove default route (route with unspecified destination address, i.e. 0.0.0.0) using command *route delete default.*
8. Ping *Rx*, where *x* is the lowest router index in your group. Can you ping *Rx*?
9. Add host route for *Rx* to the routing table using command *route add -host*. For example, @R4: *route add -host* 192.168.98.10 *gw* 10.1.3.2 *dev* eth0. Can you ping *Rx*?
10. Set up routing table at all the routers such that the table contains routes to all the subnets. First check the routing table for missing subnets, and then add their routes using *route add -net*. For example, @R4: *route add -net* 10.1.1.0 *netmask* 255.255.255.0 *gw* 10.1.4.1 *dev* eth1.
11. Show the routing table using command *route -n*.
12. Ping any router in your network. Can you ping them now?

4. Usage of tools: sniffer (wireshark, tcpdump), ping and traceroute

1. Ping the loopback address, i.e. ping 127.0.0.1, and explain what the latency shows?
2. Run */usr/sbin/tcpdump* on the interface that is used to route the packet for *Rx*. For example, */usr/sbin/tcpdump –v –i eth1&*
3. Ping *Rx*. What are the values of maximum/average/minimum ping latency? What is the TTL value of ICMP ECHO REQUEST and ICMP ECHO REPLY?
4. Remove the entry corresponding to the next hop MAC address for the *Rx* route from the arp cache, using the */sbin/arp* command. For example, @R4: */sbin/arp –d* 10.1.3.2.
5. Ping again and explain the difference, if any, in the ping latency.
6. Run ethereal on the interface that is used to route packets for *Rx*.
7. Ping *Rx* with the option *ping -R* to record the route. Which interface address is recorded in the record route option by the router in the path?
8. Run traceroute to trace the route. What is the difference?
9. Ping the subnet directed broadcast address for the subnet 10.g.1.0, i.e. ping -b 10.g.1.255, where *g* is your group number. How many echo replies are received?

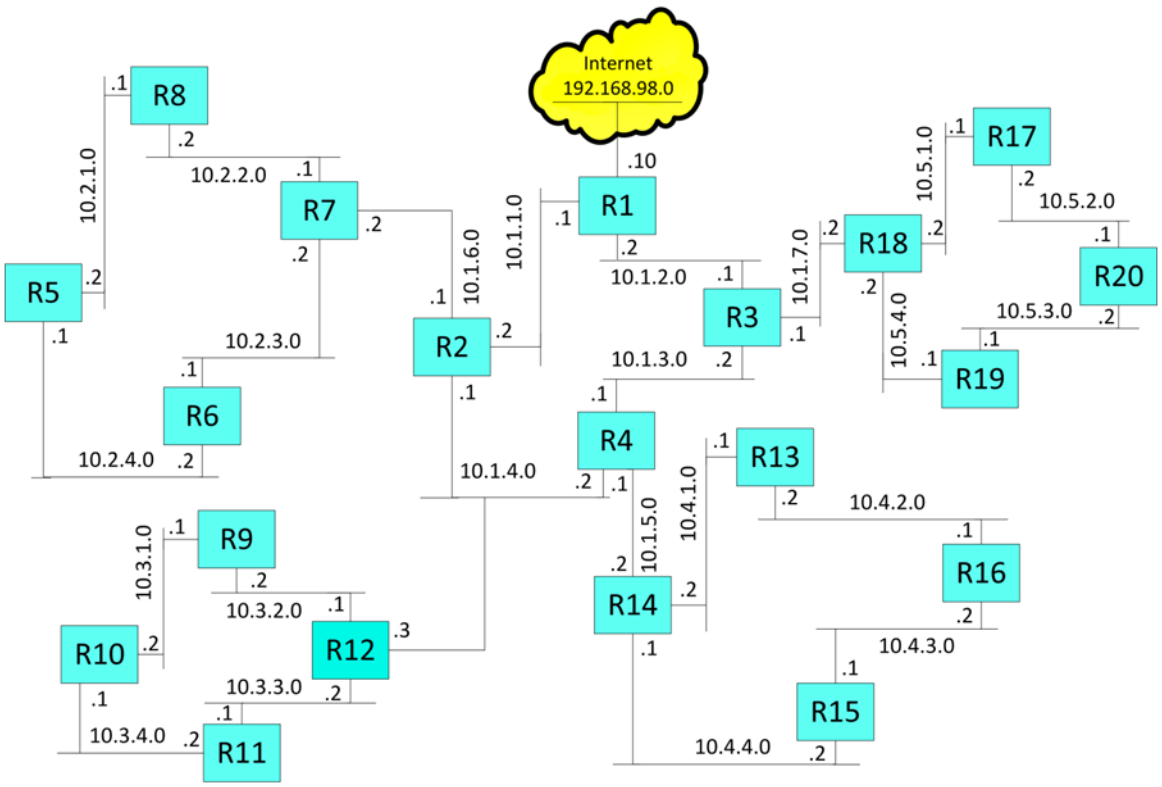


Figure 1: Physical Topology for Lab 1

Readings:

a. http://www.tldp.org/HOWTO/Networking-Overview-HOWTO.html

b. http://www.tldp.org/HOWTO/Net-HOWTO/

c. man pages of *ping*, *traceroute*, *netstat*, *ifconfig*, and *route*.

Lab1 (b): DNS

1. Objective

In this lab you will learn how to configure a DNS server and use *dig* (name resolver) to query the server.

2. DNS Configuration

1. Configure your workstation (e.g. R4) as the DNS server in the network shown in Figure 2. The address of the DNS server is the IP address of the workstation's interface, e.g. 10.1.1.4/24.
2. Define your own domain name (e.g. galaxy.net) and assign each terminal a name (e.g. mars.galaxy.net). Name the DNS server in your domain (e.g. dns.galaxy.net)
3. Create your configuration file that should specify the following records:
   SOA. (Start Of Authority)
   NS – set DNS server address (e.g. 10.1.1.4).
   A – IP address corresponding to each machine.
   MX – set mail server address to be the same as the DNS server address (e.g. 10.1.1.4).
   CNAME – set aliases to some of terminals (addresses).
4. You can write your configuration on the DNS server in the file `/var/named/galaxy.net.zone` (i.e. your zone file).
   Sample zone file: `/var/named/localhost.zone`
5. Modify the file `/etc/named.conf` to contain pointer to your zone file.
6. Modify the file `/etc/resolv.conf` at both the server side and client side to introduce the server.
   Read more about it at:
   http://www.linuxhomenetworking.com/linux-hn/dns-static.htm
7. Run dns server with the command */usr/sbin/named –u named*
   service named start "stop, restart, status"
8. Run wireshark on both the server and the client station.
9. Use *dig* (or *nslookup*) to query the server, e.g. **dig** *mars.galaxy.net*.
10. For more information about DNS read:
    a. http://tldp.org/HOWTO/DNS-HOWTO.html .
    b. http://www.linuxhomenetworking.com/linux-hn/dns-static.htm (Configuring A Regular Nameserver)
    About Hub and Switch:
    c. http://www.practicallynetworked.com/networking/howhub.htm

3. Sample DNS Configuration File

For detailed explanation, please see the "Sample Forward Zone File" at:
http://www.linuxhomenetworking.com/linux-hn/dns-static.htm#_Toc92808689

```
$TTL 86400
$ORIGIN galaxy.net.
@ 1D IN SOA dns.galaxy.net. hostmaster.galaxy.net. (
                            1 ; serial
                            3H ; refresh
                            15M ; retry
                            1W ; expiry
                            1D ) ; minimum
;
;
@  1D IN NS dns.galaxy.net           ; inet address of the name server
1D IN MX 10 mail.galaxy.net          ; mail server
1D IN MX 20 mail2.galaxy.net        ; backup mail server
;
; dns and mail server addresses
;
dns IN A 10.1.1.7
mail IN A 10.1.1.7
mail2 IN A 10.1.1.8
;
; address
;
jupiter IN A 10.1.1.21
saturn IN A 10.1.1.22
mars IN A 10.1.1.23
pluto IN A 10.1.1.24
venus IN A 10.1.1.25
mercury IN A 10.1.1.26
earth IN A 10.1.1.27
;
; aliases
;
planet IN A earth
```

** Tip: Make sure that the SOA and the email address of the person in charge are on the same line.

Resource Records

db.mylab.com

```
$TTL 86400
mylab.com. IN SOA PC4.mylab.com.
hostmaster@mylab.com. (
                        1 ; serial
                        28800 ; refresh
                        7200 ; retry
                        604800 ; expire
                        86400 ; ttl
                        )
;
mylab.com.        IN       NS       PC4.mylab.com.
;
localhost                  A        127.0.0.1
PC4.mylab.com.    A        10.0.1.41
PC3.mylab.com.    A        10.0.1.31
PC2.mylab.com.    A        10.0.1.21
PC1.mylab.com.    A        10.0.1.11
```

Max. age of cached data in seconds

•Start of authority (SOA) record. Means: "This name server is authoritative for the zone Mylab.com"
•PC4.mylab.com is the name server
•hostmaster@mylab.com is the email address of the person in charge

Name server (NS) record. One entry for each authoritative name server

Address (A) records. One entry for each hostaddress

(Source: http://www.cs.virginia.edu/~itlab/book/slides/index.html) not valid link

4. Network Topology for DNS Lab

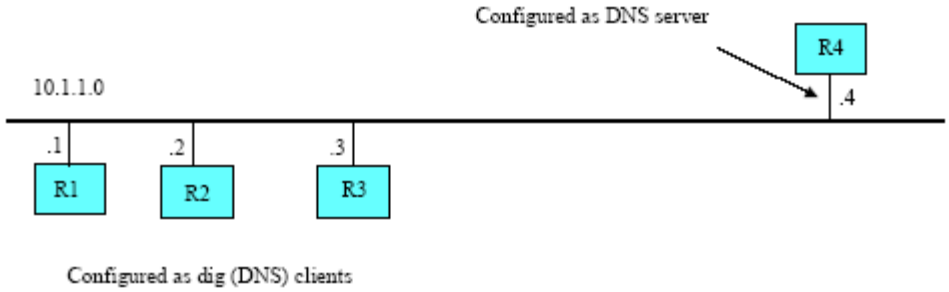


Figure 3: Topology for Lab 3

Lab1 (c): DHCP

1. Objective: In this lab you will learn how to configure a DHCP server and DHCP clients (hosts).

2. DHCP Configuration

1. Configure one interface of your station (e.g. R1:eth1) on the subnet as shown in the Figure below.
2. In this you will configure R4 as DHCP server and use R1 to R3 as DHCP clients.
3. Create your configuration file that should include following items:
   – Set default lease time and maximum lease time.
   – Set DNS server address to be the address of your station on the given subnet.
   – Set address range to be 10.1.1.10 10.1.1.20.
4. Set DHCPDARGS parameter in file /etc/sysconfig/dhcpd to "eth0".
   DHCPDARGS="eth0".
   And then restart the service using etc/init.d/dhcpd restart.
5. You can write your configuration in the file */etc/dhcpd.conf*.
   Sample dhcpd.conf: /usr/share/doc/dhcp*/dhcp.conf.sample
6. Run dhcp server with the command */usr/sbin/dhcpd* eth1.
7. Run wireshark on both the server and your client station.
8. Set your client station interface address using *ifconfig* to .56, e.g. 10.1.1.56.
9. Run dhcp client on the host using the command */sbin/dhclient* eth1.
10. Modify the *dhcpd.conf* to include manual address assignment for the host. Redo steps 5-9 to do manual allocation.
11. For more information read:
    - http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch08_:_Configuring_the_DHCP_Server
    - http://www.tldp.org/HOWTO/Net-HOWTO/
    - Man pages : *dhcpd, dhcpd.conf, dhclient*

Tip: After you finished your tests, make sure to stop the DHCP by killing the dhcpd process, or using the command: *service dhcpd stop*

3. Sample DHCP Configuration

```
ddns-update-style none;
default-lease-time 1200;
max-lease-time 9200;
option domain-name-servers 10.1.1.10;
option domain-name "galaxy.net";
subnet 10.1.1.0 netmask 255.255.255.0 {
  range 10.1.1.21 10.1.1.30;
  range 10.1.1.201 10.1.1.250;
  option routers 10.1.1.9;

  host jupiter {
```

```
    hardware ethernet 08:00:2b:4c:59:23;
    fixed-address 10.1.1.100;
  }
}
```

4. Network Topology for DHCP Lab

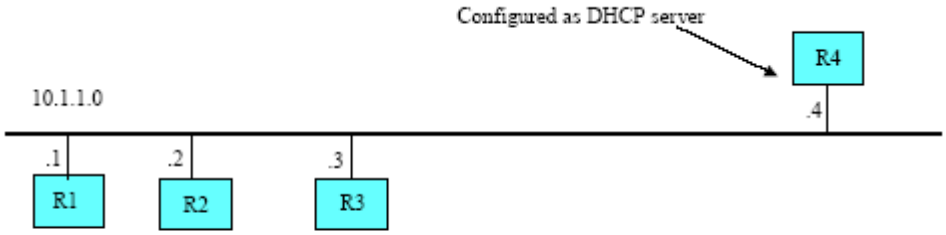


Figure 3: DHCP Configuration

Lab 2: Design and Evaluation of a RIP Network

1. Objective

The objective of this lab is to learn the network design and configuration running RIP.

2. Router configuration

1. Configure the interfaces of your router with the address, subnet mask (24-bit) and broadcast address as shown in the figure. The configuration should be persistent, so that you can restart the router without losing the configuration.
2. Check your configuration by running the commands */sbin/ifconfig*.
3. For this lab we will use Quagga routing package. Quagga uses Zebra IP routing manager. It provides kernel routing, interface lookups, and redistribution of routes between different routing protocols.
   To learn about Zebra go to:
   http://www.nongnu.org/quagga/docs/quagga.pdf (Chapter 4)

4. Keep all configuration files in your directory.
5. Prepare a configuration file for *zebra* as *zebra.conf*, which must contain the following configuration commands:

   zebra.conf:

   ```
   hostname R1
   password zebra
   enable password zebra
   log file coe865xy/zebra.log
   ```

   ** To learn about zebra config command, go to:
   http://www.nongnu.org/quagga/docs/quagga.pdf (Chapter 3)

6. Prepare a configuration file for *ripd* as *ripd.conf*, which must contain the following commands:

   ripd.conf:

   ```
   router rip
   network 10.0.0.0/8
   redistribute kernel
   passive-interface eth0
   ```

   ** To learn about rip configuration, go to:
   http://www.nongnu.org/quagga/docs/quagga.pdf (Chapter 5)

7. Start zebra daemon as *zebra –df coe865xy/zebra.conf* from the home directory. If Zebra cannot be run, use the following command to add Zebra to your working path:
   `$export PATH=/usr/sbin:$PATH`

8. Start ripd as *ripd –df coe865xy/ripd.conf* from the home directory.

** Tip: Running zebra is mandatory to run ripd. Thus, keep the above sequence in running the software.

9. Run *route -n* on all of the routers in your subnet and observe the routing table.
10. Run wireshark to monitor the RIP traffic. Observe the RIP interval.
11. Ping a router in your network using a route advertised by your neighbor router.
12. Ping the farthest router in your network.
13. Disconnect the router's interface that is used in routing packets to the farthest router. Run wireshark on that interface to collect the trace for convergence. Then, run ping to the farthest router for more than rip flush time. Discuss your findings with the TA. Identify RIP phases in the trace.
14. Reconnect the interface (which was disconnected in step 13), run wireshark and collect the trace for reconvergence. Identify RIP convergence phases in the trace.
15. For more information on zebra read man pages on *zebra* and *ripd*, and info page on zebra using command info zebra.
16. You can get information about zebra from http://www.nongnu.org/quagga/index.html

3. Zebra Configuration

1. Telnet zebra to run CLI using the following command *telnet localhost* 2601.
2. Show IP routing tables for rip and kernel routing.

** Tip: Zebra Terminal Mode Commands at: http://www.nongnu.org/quagga/docs/quagga.pdf (Chapter 4.5), section "zebra Terminal Mode Commands".

3. Add few static routers into the routing table using route command using the linux shell interface.
4. Redistribute the static routes by modifying the *ripd.conf* file.

** Tip: go to http://www.nongnu.org/quagga/docs/quagga.pdf (Chapter 5.4), section "How to announce RIP route".

** Tip: Try various commands for zebra and RIP. Information is available here: http://www.nongnu.org/quagga/docs/quagga.pdf (Chapter 3, 4 and 5)

4. Troubleshooting

1. If zebra.log shows following:

```
2013/02/14 12:16:07 ZEBRA: Zebra 0.98.5 starting: vty@2601
2013/02/14 12:30:47 ZEBRA: Could not lock pid_file
/var/run/quagga/zebra.pid, exiting
```

2. Following steps should be followed to fix the problem:

```
$rm –rf /var/run/quagga/zebra.pid
$pkill zebra
$pkill ripd

Then, start zebra and ripd daemons.
```

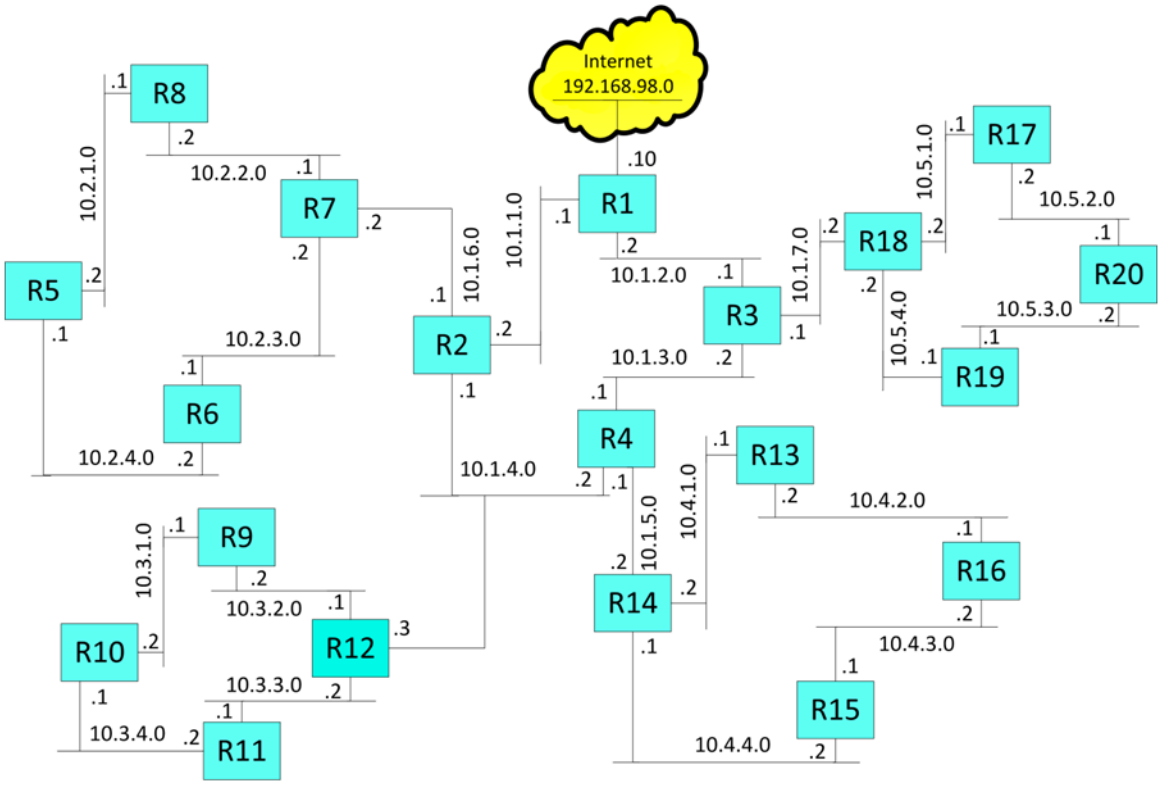


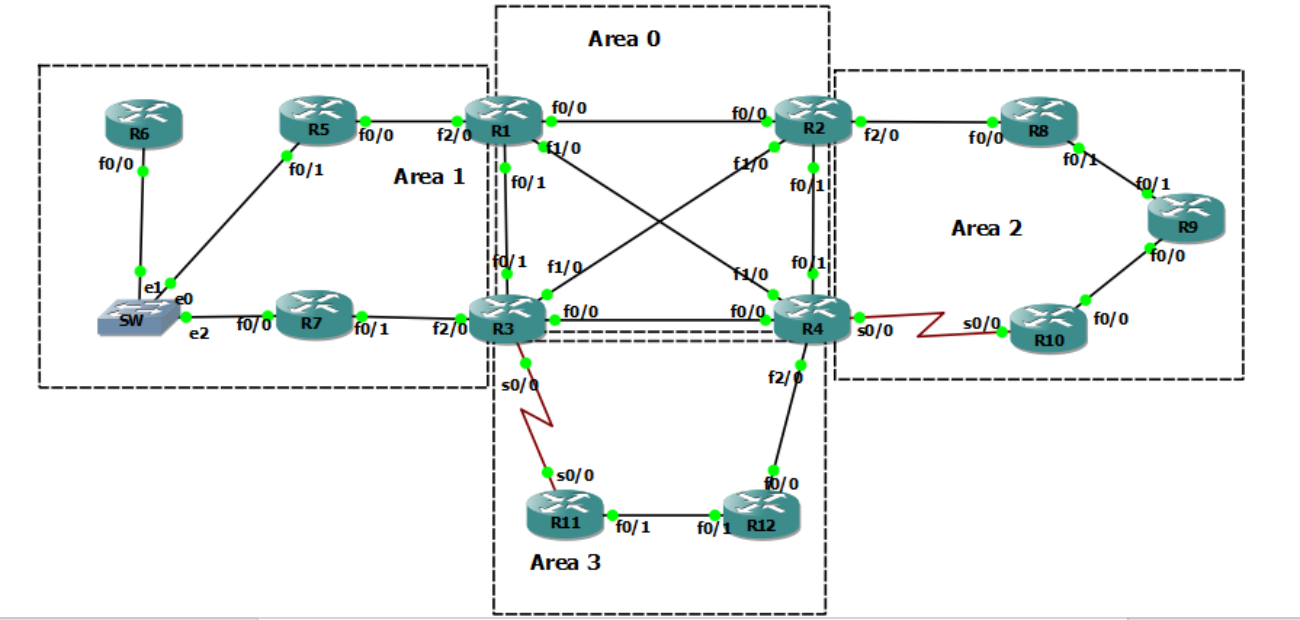
Figure 4: Physical Topology for Lab 2

Readings:

1. For Zebra and RIP, refer to: http://www.nongnu.org/quagga/docs/quagga.pdf (Chapter 4 and 5 respectively)
2. http://tldp.org/HOWTO/Adv-Routing-HOWTO/index.html
3. man pages of *zebra* and *ripd*. Also, see *info zebra*

Lab 3: OSPF

I) Network Topology:



II) IP addresses table:

Table 1: IP addresses.

Router	IP of Fa0/0	IP of Fa0/1	IP of Fa1/0	IP of Fa2/0	IP of Se0/0
R1	10.0.12.1/24	10.0.13.1/24	10.0.14.1/24	10.1.15.1/24	-
R2	10.0.12.2/24	10.0.24.2/24	10.0.23.2/24	10.2.28.2/24	-
R3	10.0.34.3/24	10.0.13.3/24	10.0.23.3/24	10.1.37.3/24	10.3.113.3/24
R4	10.0.34.4/24	10.0.24.4/24	10.0.14.4/24	10.3.124.4/24	10.2.104.4/24
R5	10.1.15.5/24	10.1.123.5/24	-	-	-
R6	10.1.123.6/24	-	-	-	-
R7	10.1.123.7/24	10.1.37.7/24	-	-	-
R8	10.2.28.8/24	10.2.89.8/24	-	-	-
R9	10.2.109.9/24	10.2.89.9/24	-	-	-
R10	10.2.109.9/24	10.2.104.10/24	-	-	-
R11	-	10.3.112.11/24	-	-	10.3.113.11/24
R12	10.3.124.12/24	10.3.112.12/24	-	-	-

III) Configurations Part I: OSPF with single area

1. Configure the above topology and assign the IP addresses as shown in Table1.

2. Configure loopback interface for each router Rx where IP address is x.x.x.x/24.
   For example: IP address for loopback interface on R4 is 4.4.4.4/24.

3. Run OSPF protocol on all routers R1-R12 and assign them in a single area.
   - Use ***router ospf xx*** where ***xx*** is the process ID and can be any number between 1-65535
   - Define networks under ospf process using the following command:

   (config-router)# ***network*** A.B.C.D {network ID} A.B.C.D {wild card} ***area 0***
   For example: ***network 10.1.0.0 0.0.255.255 area 0***

4. On R1 perform the following commands:
   a) ***show ip ospf neighbor***
      - how many neighbors R1 does have?
      - Observe the state of R1 with its neighbor.
   b) Configure ***ip ospf network point-to-point*** under interface fa0/0 on R1 and R2.
      - Perform ***show ip ospf neighbor*** command on R1 and check the state of neighbor 2.2.2.2

- Discuss the difference in a & b.

c) ***show ip ospf database***

- How many LSA types you could find.

5. Examine the DR, BDR state on the broadcast domain of R5, R6 and R7, discuss the result. Then, shutdown int Fa0/0 on R7 and observe the changes. Capture the OSPF messages on R6 interface using Wireshark.

6. Redistribute the loopback interfaces using the following command:
   (config-router)# ***redistribute connected subnets***
   a) Repeat 4-c and compare the difference in the LSA types in each case.
   b) Perform ***show ip route*** command and observe how many external (E2 type) routes are there.

7. Make sure that all interfaces are reachable by using ***ping*** command.

IV) Configuration Part II: OSPF with multiple areas.

1. Configure OSPF areas as shown in the topology, where:
   - R5, R6, and R7 in area 1 and R1 and R3 are ABRs for this area.
   - R8, R9, and R10 in area 2 and R2 and R4 are ABRs for this area.
   - R11 and R12 in area 3 and R3 and R4 are ABRs of this area.

2. Study OSPF database of R1 for example and compare the LSA types to a single OSPF domain.

3. In area 1, record the size of the routing table in an interior router.

4. Configure area 1 as stub area, to do that, make sure all interior routers are not ASBR, remove the redistribution command and add loopback network to the OSPF, then configure the area as a stub.

   (config-router)#***area 1 stub***

5. Repeat step 3 and compare the routing table with the previous case.

6. Configure area 1 as totally stub area.

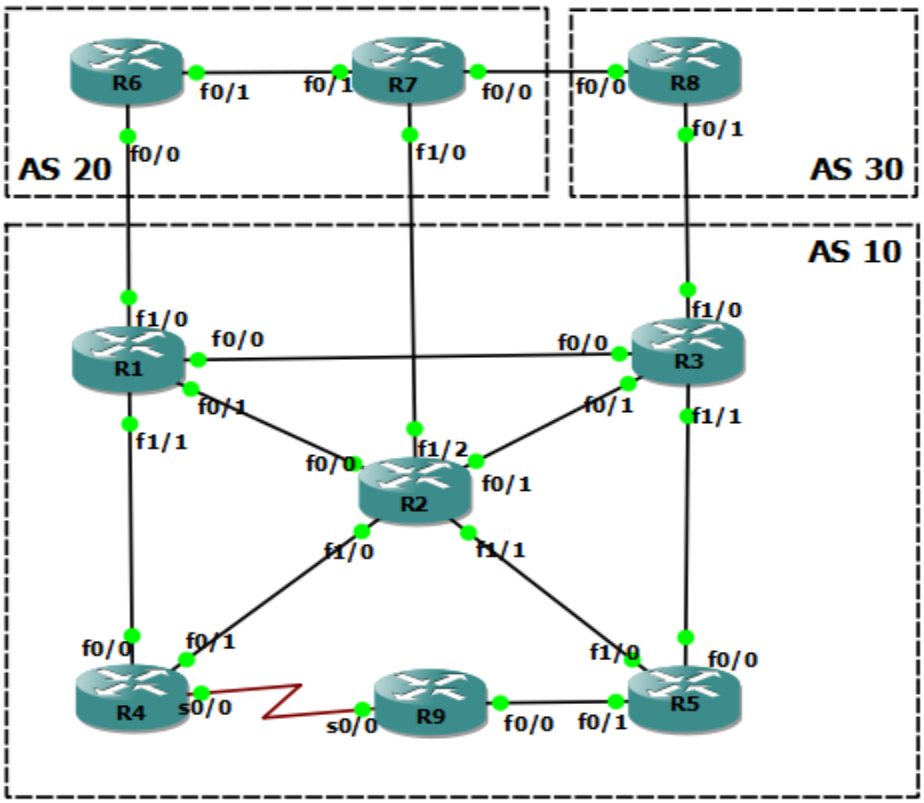
   (config-router)#***area 1 stub no-summary***

7. Repeat step 3 and compare the routing table with the previous cases.

8. Configure Area 2 as NSSA and compare the between areas 1, 2, and 3 in term of IP routing and OSPF database LSA types.

Lab 4: BGP

1. Network Topology:



2. IP addresses Table:

Router	IP of Fa0/0	IP of Fa0/1	IP of Fa1/0	IP of Fa1/1	IP of Fa1/2	IP of Se0/0
R1	10.10.13.1/24	10.10.12.1/24	12.12.16.1/24	10.10.14.1/24	-	-
R2	10.10.12.2/24	10.10.23.2/24	10.10.24.2/24	10.10.25.2/24	12.12.27.2/24	-
R3	10.10.13.3/24	10.10.23.3/24	13.13.38.3/24	10.10.35.3/24	-	-
R4	10.10.14.4/24	10.10.24.4/24	-	-	-	10.10.49.4/24
R5	10.10.35.5/24	10.10.59.5/24	10.10.25.5/24	-	-	-
R6	12.12.16.6/24	20.20.67.6/24	-	-	-	-
R7	23.23.78.7/24	20.20.67.7/24	12.12.27.7/24	-	-	-
R8	23.23.78.8/24	13.13.38.8/24	-	-	-	-
R9	10.10.59.9/24	-	-	-	-	10.10.49.9/24

Note:

- IP address of the Loopback interface for Rx is x.x.x.x/24
- Configure another loopback on R8 with 30.30.8.8/24 address

3. Configuration Part I: Network setup

1- Run OSPF in AS10, make sure OSPF carry all 10.10.0.0/16 routes.
2- Configure BGP connection peers between:
   - R1-R2, R1-R3, R1-R6
   - R2-R3, R6-R7, R7-R8
   - R3-R8

Use ***neighbor x.x.x.x remote-as y*** command under BGP configuration mode. Where x.x.x.x is the IP address of the remote peer interface and y is the Autonomous System AS.

3- Verify the BGP connections using the following command:
   Show ip bgp summary

4- Advertise the following networks:
   10.10.12.0/24, 10.10.13.0/24, 10.10.23.0/24, 10.10.14.0/24, 10.10.24.0/24, 10.10.25.0/24, 10.10.35.0/24, 10.10.49.0/24, 10.10.59.0/24, 20.20.67.0/24, 30.30.8.0/24
   Use network command under BGP, ***network x.x.x.x mask y.y.y.y***

5- Check and record the routing table of R1, R3, R6 and discuss the bgp routes, what is missing?

6- Configure next-hop-self on all iBGP neighbor peers, repeat step 5 and discuss the difference.

7- Instead of advertising all 10.10.x.0/24 networks, you may advertise one network 10.10.0.0/16. And for that, two steps needed:
   - Configure network 10.10.0.0 mask 255.255.0.0 under BGP 10.
   - Make sure that 10.10.0.0/16 in routing table and this can be done by configure static route to null 0
     Ip route 10.10.0.0 255.255.0.0 null 0
   - Check routing table of R6 or R8 for verification.

8- Examine the connectivity of the network by pinging network 10.10.59.0/24, 10.10.49.0/24 from AS20 and AS30.
   - Default route for non BGP router in AS10 is needed.
     Use: ***default-information originate*** command under OSPF.

- Make sure that the source address of the ping command is one of advertised network.

4. Configuration Part II: Traffic control

1- Record the routing table of R6 and R7.
   - Use ***show ip bgp*** to understand how BGP protocol select the preferable route.
   - Use Local Preference attribute to configure (AS 20) routers (R6, R7) in such that the traffic passing through AS20 to AS10 is divided as follow: traffic to 10.10.12.0/24, 10.10.13.0/24, 10.10.14.0/24, 10.10.23.0/24 selects R6-R1 path and the reset is going through R2-R7 path.

2- Use Weight attribute on R7 to make traffic to 10.10.12.0/24 net passing through R7 instead of R6.

3- Using As-Path attribute, configure AS20 to make sure that traffic to 20.20.67.0/24 network passing through AS10.