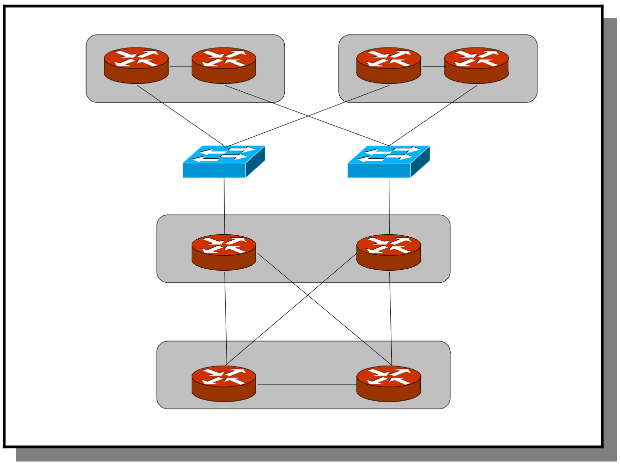
KTH EECS

Introduction to Routing



STUDENT:

Group#: ______________________

Name1: ______________________

Name2: ______________________

Table of contents

1 Goals, background, and overview

The goal of this lab is to give an introduction to dynamic routing and quagga routing software suite. After the lab, you should be able to configure the command-line interface of a quagga routing software to make a simple dynamic routing set-up. The network is very simple but provides an insight into several important routing issues.

Before you start with the lab, there are some preparations that you must complete, as follows:

- Complete the preparation questions quiz in Canvas;
- Find your assigned group number in Canvas. You will use it in your lab configuration;
- Read through the lab instruction and complete tasks as per instructions from the introduction parts of the lab in Section 2–4.

The “real” lab starts in Section 5, in which you will configure network interfaces of routers and Linux hosts. You will learn to configure OSPF on a pair of routers representing a minimal interior network in Section 6. Finally, you will learn to configure BGP towards an external network in Section 7. Moreover, you will answer questions in each section (all questions are in the routing lab questions quiz in Canvas) and submit the configuration files as a part of the routing lab activities.

2 Preparations

There are some preparations you must complete before you start the lab.

2.1 Routing lab group number

Each student is assigned a group number in *Canvas→ People → EP Lab 4 Group*. In this lab, you will configure routers/hosts based on the group number and submit the lab results individually.
IMPORTANT: Your assigned group number is ≥ 3.

2.2 Preparation questions

The preparation questions are available on Canvas in a form of a quiz. The quiz is mandatory and we recommend that you complete the quiz beforehand. You will pass the routing lab only if you pass both the quiz and the actual lab. The actual lab comprises two parts: the routing lab questions quiz and the routing lab submission.

You should study the following materials before doing the preparation questions:

- The lecture slides are useful as background information, and so are the relevant sections of the book.
- The homework assignment that covers routing.
- The lab instructions. You must read through the lab instruction to understand what this lab is all about.
- References for network and routing configurations:
  - Quagga Routing Suite Documentataion is the main reference for router configuration. You should browse through the following sections.

        - Basic commands
        - OSPFv2
        - BGP
    - For Linux network configuration command, see examples below:
        - how to use the ip command instead of ifconfig
        - ip command cheatsheet

2.3 Setup VirtualBox and routing_lab VM

You must install routing_lab VM on your laptop before starting the lab! The VM requires 1GB memory and 10GB disk space.

Our lab is based on a virtual machine on VirtualBox. You are expect to run this VirtualBox VM on your laptop and must set up your system accordingly before you start to the lab configuration.

VirtuaBox
You can download VirtualBox platform packages that suits your operating system from URL below:
https://www.virtualbox.org/wiki/Downloads

The installation should be intuitive. You can also find many guide online. For example, on wikihow.

You may also download and install the VirtualBox Extension Pack, which gives you support for USB devices, etc.

Routing_lab VM
To setup routing_lab VM, follow the instruction on Section 3.

3 Setup routing_lab VM

Before you proceed, make sure to have sufficient resources available on your laptop. The VM requires **1GB memory** and **10GB disk space**.

3.1 Install routing_lab VM

You can download routing_lab VM from https://home.ug.kth.se/ by open the URL in your browser. You will see a pop-up window asking for an authentication as shown in Figure 1 below.

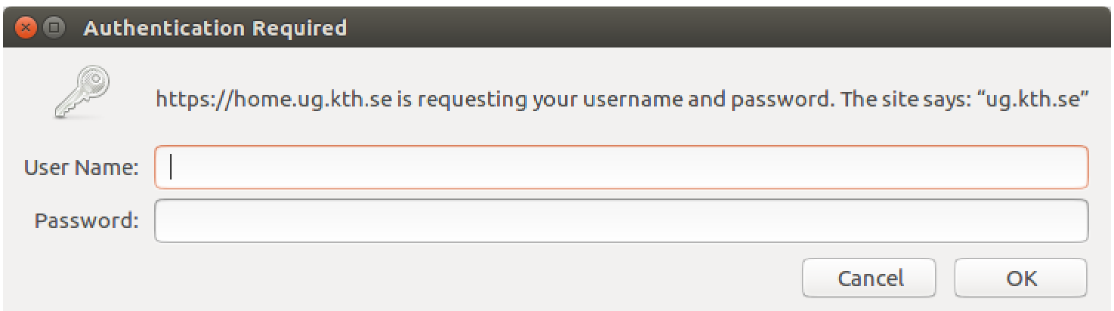


Figure 1: Authentication required for https://home.ug.kth.se

Use your KTH username and password to log in.

After logged in, from the directory list on the sidebar menu, choose:
KTH Studentresurser→kurser→ICT→ik2218
You should see a routing_lab.ova file as shown in Figure 2 below.

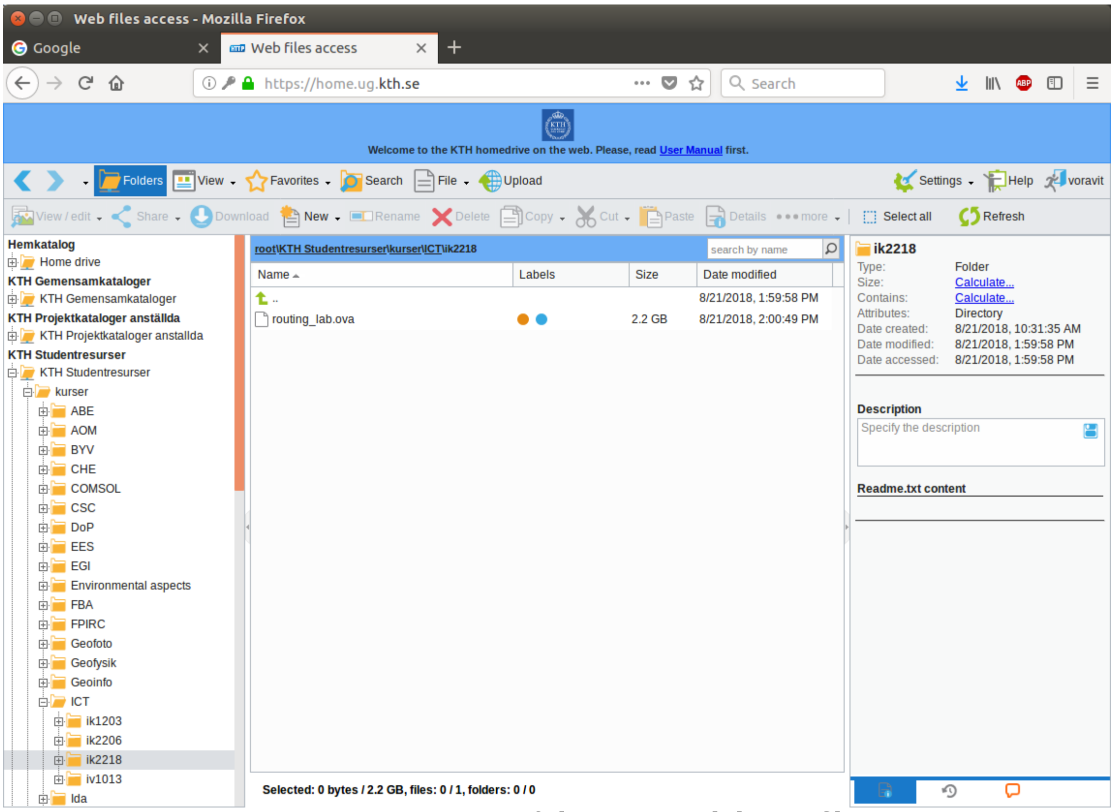


Figure 2: Location of the routing_lab.ova file

Download the file to your computer. Also check that the filesize is correct.

After download, just double-click on the routing_lab.ova file to install. The *Import Virtual Appliance* window will pop-up.

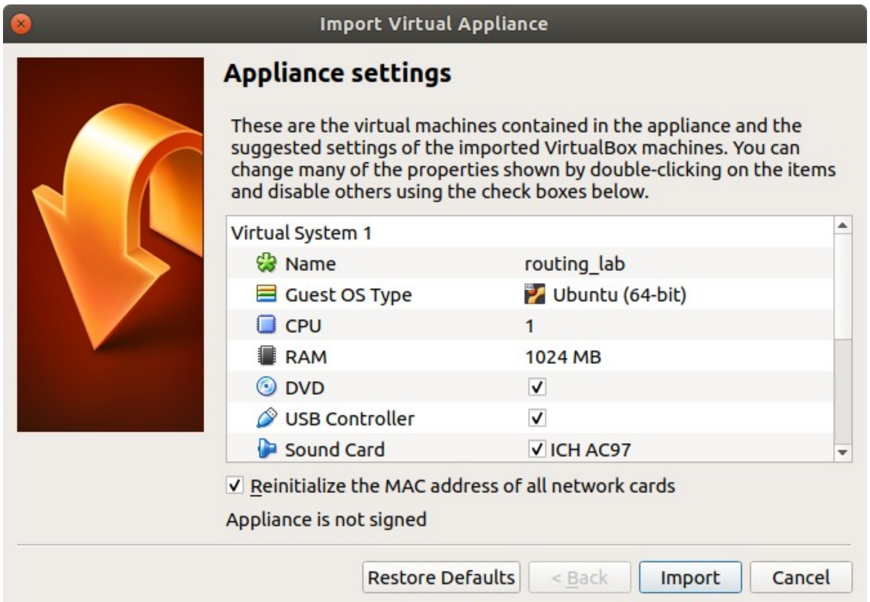


Figure 3: Import Virtual Application: the routing_lab VM

Mark the check box for *Reinitialize the MAC address of all network cards*
Then, click *import* as shown in Figure 3.

3.2 Setting up host-only Adapter

After the VM is imported, you will now set up host-only adapter.

From the menu of the Oracle VM VirtualBox Manager windows,
Select File→Tools→Network Manager
(or simply click the *Tools's List* icon and select *Network)*
Add a new network by click on the add icon as shown in Figure 4.

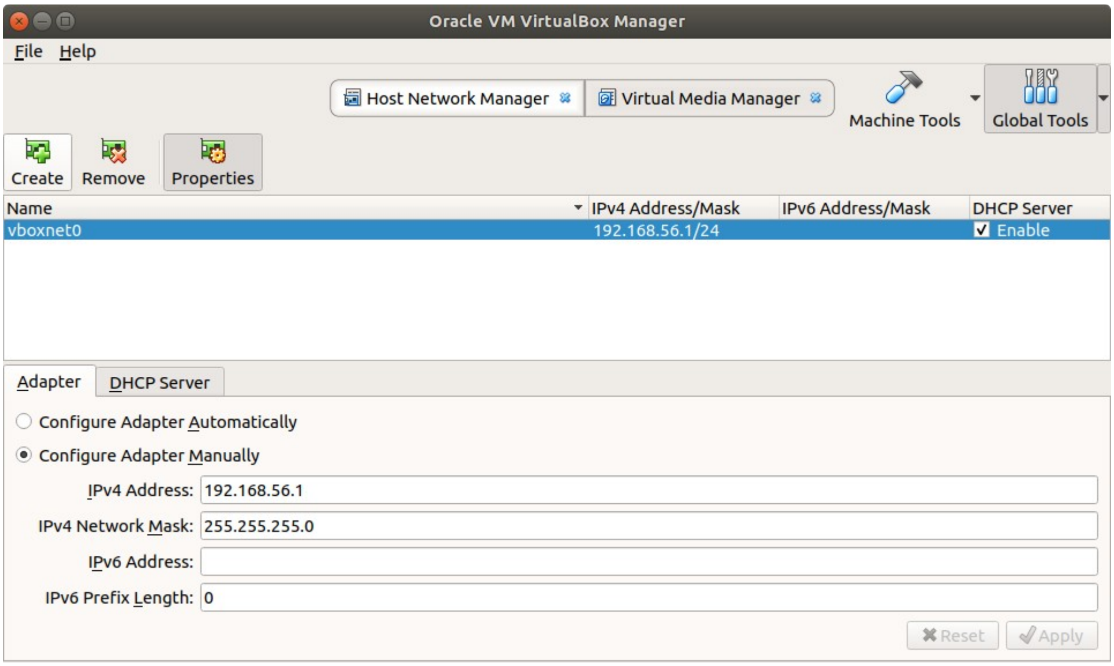


Figure 4: Add a new Host-only Network Adapter

Use the default Host-only Networks name: *vboxnet0*
Similarly, you should use the default IPv4 address (192.168.56.1) and Mask (255.255.255.0) as shown in Figure 4.

Now go back to the main VirtualBox Manager, select routing_lab VM.
Click on *Settings* icon at the top of the windows.
Select *Network* on side-bar list and click on *Adapter 2* tab
Mark the check box for *Enable Network Adapter*
Make sure that it is configured as follows.

Attached to:	*Host-only Adapter*
Name:	*vboxnet0*

Then, click *OK* as shown in Figure 5.
At this point, your routing_lab VM is set up and ready for the lab.

If your computer has more RAM than 8 GB, you can allocate more RAM to the VM to make it run faster by modifying the VM settings:
From the *Settings* windows, Select *System*. Then, on the *Motherboard* tab, you can increase the base memory to 2048 MB.

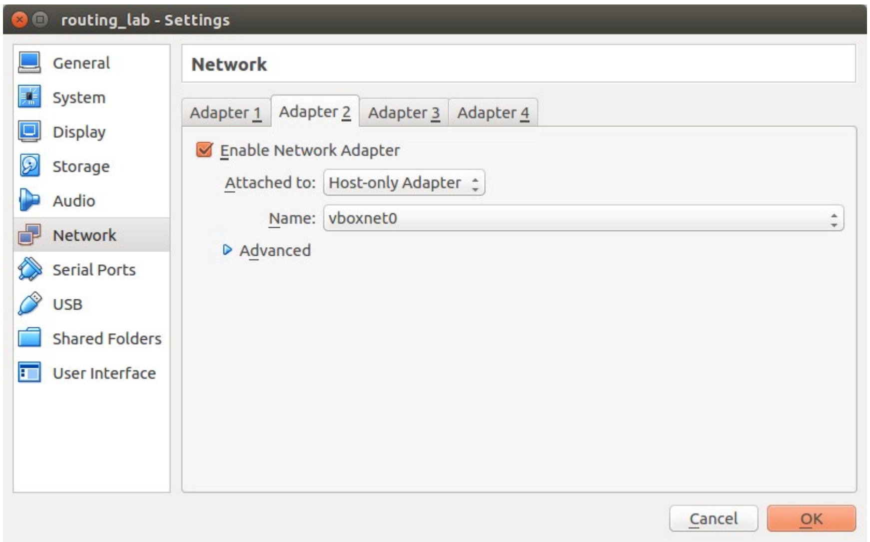


Figure 5: Setting Adapter 2 of routing_lab VM

4 Routing Lab Overview

This section gives you an overview of the network topology used in this lab and how you can access them. You must read through this section and understand what you are expected to do before you proceed to the next section. **The actual configuration starts in Section 5!**

4.1 Network Topology

The routing lab VM runs multiple containers to form a topology as shown in Figure 6. You will be configured Linux hosts and routers in the top part of the topology, i.e., rta, rtb, hosta, and hostb.

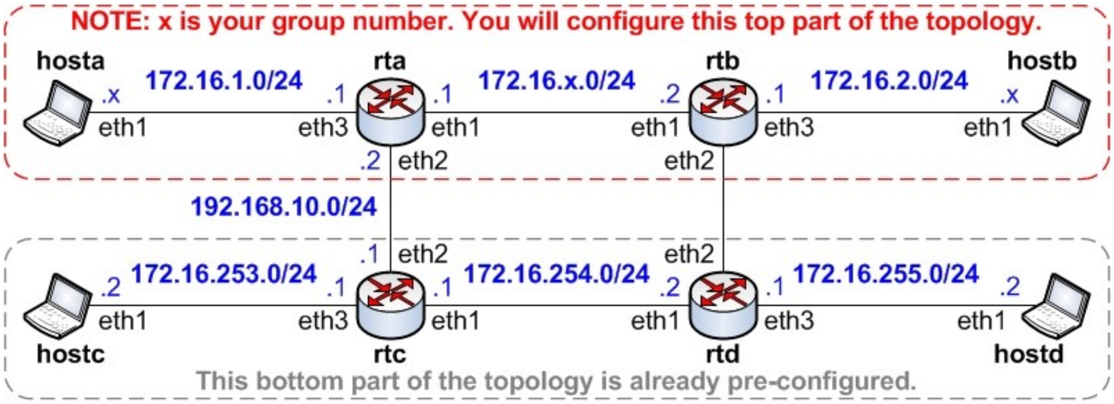


Figure 6: Topology of how containers inside the routing_lab VM are connected

You will configure different interfaces of the Linux hosts and routers using different subnets and IP addresses as shown in Figure 6. The physical interfaces are labeled in black and have convention interface names (eth1, eth2, and eth3). The IP addresses are labeled in blue and consists of two parts: the subnet and the host address. The network subnets are written in bold while the host address in normal text.

This means that to configure IP addresses of hosta and rta, you will use a subnet **172.16.1.0/24** and the host address for hosta and rta are **.x** and **.1** respectively. You will replace **.x** with your group number. For example, if you have selected **group 5** in Canvas, you will configure hosta IP address on **eth1** to **172.16.1.5/24** and you will configure rta IP address on **eth3** to **172.16.1.1/24.**

4.2 Accessing the routing_lab VM and containers

Our lab setup utlizes a nested virtualized system. The main virtual machine is the routing_lab VM running on VirtualBox. It runs multple isolated Linux systems using Linux containers (LXC). We may refer to the routing_lab VM as *the host VM* interchangeably.

NOTE: After you start the routing_lab VM, it may take some time before the routing_lab VM completes the booting process. In the background, the containers will also be launched and these contianers will take even longer time to complete the booting process. Thus, you may need to wait some minutes before the containers are ready. **Be patient!**

4.2.1 Accessing the routing_lab VM

For the ease of access, we recommend you to use SSH to access the routing_lab VM. This is possible once you set up the host-only adapter as described in Section 3.2.

You will need to start the routing_lab VM from the VirtualBox Manager. After the routing_lab VM finished the booting process, you can SSH to it directly or via a terminal application, such as PuTTY and PowerShell from Linux, MAC OS, and Windows OS.

To SSH to the routing_lab VM, use the following credential.

USERNAME:	`student`
PASSWORD:	`time2work`
IP:	`192.168.56.2`

From your computer’s terminal, you can SSH with the command below.

```
ssh student@192.168.56.2
```

4.2.2 Accessing the containers

To access a container, you will need to SSH to the routing_lab VM first. Then, from the routing_lab VM, you can connect to the console of a container using the LXC command line tool.

From the rouing_lab VM, use the following command to access rta.

```
lxc console rta
```

You will also need to log in to the container, use the following credential.

USERNAME:	`ubuntu`
PASSWORD:	`ubuntu`

We strongly recommend that you open one terminal per container and connect to all of them simultaneously, i.e., you should have 4 terminals to connect to rta, rtb, hosta, and hostb. On each terminal, you will first SSH to the routing_lab VM and then connect to the console of a container with the `lxc console <CONTAINER_NAME>` command.

NOTE: You can connect to only one console per container at a time!

5 Lab start: Interface configuration

In this section, you will configure the network interfaces of Linux hosts and routers in the top part of the topology in Figure 6, i.e. hosta, hostb, rta, and rtb. But, first, you need to start the routing_lab VM as follows.

Launch VirtualBox and start the routing_lab VM on your laptop.
Wait for the VM to boot up, you should see `ubuntu login:` on the screen.
Now, you can connect to each container as described in Section 4.2.

TIPS: It is helpful to write your group number next to the **.x** in Figure 6.

5.1 Configure interface of Linux hosts

For Linux hosts (hosta and hostb), you can configure them using standard Linux command for network interface configuration (either `ifconfig` or `ip` command). For each host, you need to do the following.

- Configure eth1 interface IP address of the host.
- Set a default route to the IP address of the router on the same link

For example, if you are group 252, you will configure hosta as follows.

Add IP address on eth1 interface of hosta
Option1: `sudo ifconfig eth1 172.16.1.252 netmask 255.255.255.0`
Option2: `sudo ip addr add 172.16.1.252/24 dev eth1`

Set a default route of hosta to rta
Option1: `sudo route add default gw 172.16.1.1`
Option2: `sudo ip route add default via 172.16.1.1`

TIP: You can see the current interface configuration with either `ifconfig` or `ip addr` command. And you can see the IP routing table with with either `route -n` or `ip route` command.

Now, configure hosta and hostb according to your group number!

5.2 Configure interface of routers (via Quagga)

We use Quagga routing software suite to emulate routers in our topology. It supports many unicast routing protocols such as RIP, OSPF, BGP, and IS-IS. It also provides an integrated shell with configuration commands called VTY shell (vtysh). It is intuitive and similar to Cisco IOS command line interface (CLI). In this lab, you will use the vtysh to configure interfaces and routing protocols on all routers, i.e., rta and rtb.
For each router, you need to do the following.

- Configure IP address of the router’s interfaces (eth1 and eth3)

- Enable IP forwarding (to let the emulated router forward packets)

Similar to when you configure the hosts, you will need to SSH to the routing_lab VM and then access to the console of the router container.

Once you are on the router contianer, start vtysh with a command below:

```
sudo vtysh
```

You should see welcome text and a prompt with the container name follow by a # sign as shown below:

```
Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

rta#
```

Quagga commands are similar to Cisco IOS commands, which are structured in a hierarchical format with different command modes. In order to configure the router, you will need to enter configuration mode by issuing the following command:

```
rta# configure terminal
```

You will see the prompt changes indicating that you are in config mode:

```
rta(config)#
```

From this point, you can enter different mode of configuration.
To configure the interface, you need to entery interface configuration mode. For example, to configure eth1 interface of rta with IP address 172.16.252.1/24, you will need to do the following:

```
rta(config)# interface eth1
rta(config-if)# ip address 172.16.252.1/24
```

In Quagga, if you make a mistake in your configuration, you may remove the wrong configuration by running the same wrong command with the keyword **no** in front. For example, if you want to configure interface IP address 172.16.252.1/24, but you accidentally type 172.16.252.10/24, then you can correct it by doing the following:

```
rta(config-if)# ip address 172.16.252.10/24    (Add wrong IP)
rta(config-if)# no ip address 172.16.252.10/24   (Delete wrong
ip)
rta(config-if)# ip address 172.16.252.1/24     (Add correct
IP)
```

IMPORTANT: You should always delete any wrong configuration that you make. Otherwise, the router may operate incorrectly.

You can use the `exit` command to move up one step in the command hierarchy or use the `end` command to move to the top level:

```
rta(config-if)# exit          (move 1 step up)
rta(config)#
```

```
rta(config-if)# end           (move to the top level)
rta#
```

To enable IP forwarding, use the command below from the configuration mode:

```
rta(config)# ip forwarding
```

TIP: You can see the current operating configuration with a command `show running-config`.

Now, configure eth1 and eth3 interface as well as enable IP forwarding on both rta and rtb according to your group number!

5.3 Verify the interface configuration

In this section, you will verify that the interfaces are configured correctly. You will test that the interface are reachable using `ping` command.

By this point, if you have configured interfaces of hosts and routers correctly, the directly connected interfaces should be reachable. You can verify this by sending an ICMP message to the directly connected interfaces with a `ping` command, which you should get a response. Otherwise, it is likely that you make some mistake in your configuration. In this case, you will need to check and correct it accordingly.

To send a ping command you can simply use the following command:

```
ubuntu@hosta:~$ ping 172.16.1.1     (ping from host)
rta# ping 172.16.1.1                (ping from router)
```

Now, you will test interface reachability with ping command according to the instruction in the table below. If ping works, mark the **OK** box. Otherwise, you need to troubleshoot your configuration!

PING TEST	**OK**	**NOT OK**
From hosta, ping 172.16.1.1		
From hosta, ping 172.16.x.1		
From rta, ping 172.16.1.x		
From rta, ping 172.16.x.2		
From rtb, ping 172.16.x.1		
From rtb, ping 172.16.2.x		
From hostb, ping 172.16.2.1		
From hostb, ping 172.16.x.2		

At this point, you should have verified that the ping tests above are all ok. Now, answer the following questions.

Question1: From rta, ping 172.16.2.1. Does the ping work?

Question2: If ping did not work, what is the main reason?
HINT: On rta, run `show ip route` to see its routing table.

6 Dynamic routing with OSPF

In this section, you will configure OSPF as the main dynamic routing protocol. OSPF is a link state routing protocol. It uses link-state information to make routing decisions based on the shortest-path-first (SPF) algorithm (also known as Dijkstra algorithm). Each router running OSPF floods link-state advertisements (LSA) containing information about that router's attached interfaces and routing metrics. Each router collect LSAs to build a link-state database (LSDB), which it uses to calculate the shortest path to any other routers in the network.

To configure OSPF, you will need to do the following.

- Enable OSPF routing by entering OSPF router configuration mode
- Set router-id to the eth1 IP address of your router
- Enable OSPF on eth1 and eth3 interfaces on the backbone area (area 0)

Here are example configurations on rta for 1) enable OSPF routing, 2) Set router-id to 172.16.252.1, and 3) enable OSPF on eth3 interface on backbone area:

```
rta(config)# router ospf
rta(config-router)# router-id 172.16.252.1
rta(config-router)# network 172.16.252.0/24 area 0
```

IMPORTANT: You must set the router-id first before you enable OSPF interface. Otherwise, the router may advertise its link-state with wrong information!

TIP: You can save your router configuration with the commands below:

Option1: `rta# write memory`
Option2: `rta# copy running-config startup-config`

Now, configure OSPF on rta and rtb according to your group!
Set the router-id first, then configure OSPF interface for both eth1 and eth3.

6.1 Verifying OSPF configuration

After you finish configuring rta and rtb, you will test OSPF routing reachability with ping command according to the instruction in the table below. You should **wait for one minute** to let the network converge before performing the ping test. If ping works, mark the **OK** box. Otherwise, you need to troubleshoot your OSPF configuration!

PING TEST	OK	NOT OK
From rta, ping 172.16.2.1		
From rtb, ping 172.16.1.1		

From hosta, ping 172.16.x.2		
From hosta, ping 172.16.2.1		
From hosta, ping 172.16.2.x		
From hostb, ping 172.16.x.1		
From hostb, ping 172.16.1.1		
From hostb, ping 172.16.1.x		

At this point, you should have verified that the ping tests above are all ok.

Now, write the command below on both terminals of rta and rtb:

```
clear ip ospf interface
```

After you write the command, press enter to execute the command on both terminals. You need to do this in succession as quickly as you can to ensure that you get the correct behavior.

Wait one minute for the network to converge. Then, answer the following questions.

Question 3: Once the network is converged, you should see identical routes in the routing tables of rta and rtb. Now, check the OSPF route on both rta and rtb with a command `show ip ospf route`.How many OSPF routes do rta and rtb learn via OSPF?

Question 4: On rta and rtb, check the IP routing table with a command `show ip route`.Do the routers select all OSPF routes as FIB route?

Question 5: If not all OSPF routes as FIB route, explain why were not all OSPF routes selected?
HINT: The symbol > indicates that the route is selected and the symbol * indicates the FIB route. A FIB route is what a router uses for the actual packet forwarding.

Question 6: Check the OSPF neighbor with a command `show ip ospf neighbor`.Can you identify which router is DR and which is backup?

Question 7: Choose statements that correctly describes the DR.

Question 8: Check the OSPF database with a command `show ip ospf database router`. You should see that the command output of both rta and rtb contains the same information. You can also see which links connected to stub networks and which links connected to transit networks. Can you identify how many links being advertised by each router?

7 BGP peering

In the previous section, you have configured OSPF to provide dynamic routing within your group. In BGP context, this type of internal routing that is intented to be used within an Autonomous System (AS) is referred to as an interior gateway protocol (IGP). In this section, you will use BGP to connect to external networks residing in other ASes.

Your task is to announce your internal networks to an external BGP peer so that your networks are reachable from other ASes. You will set up your BGP using AS number (ASN) 65000. The other ASes (65001 and 65002) are already pre-configured for you on rtc and rtd respectively. The overview of the network topology is shown in Figure 7.

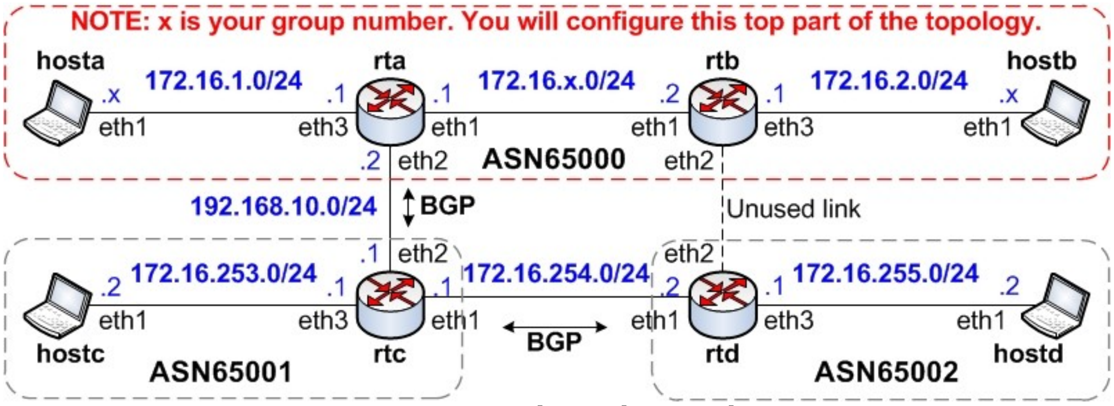


Figure 7: Network topology with BGP

Before you start with BGP configuration, you must configure rta's eth2 interface with IP address 192.168.10.2/24.

To configure BGP, you will need to do the following.

- Enable BGP on rta with ASN 65000
- Set BGP router-id to the eth2 IP address of your router
- Set up BGP peering with rtc IP 192.168.10.1 on ASN 65001
- Advertise three networks below:
    - 172.16.1.0/24
    - 172.16.2.0/24
    - 172.16.x.0/24 (where x is your group number)

Here are example configurations on rtc for 1) enable BGP with ASN 65001, 2) set router-id to 192.168.10.1, 3) set up BGP peering with rta IP address 192.168.10.2 and ASN 65000, and 4) advertise a network 172.16.253.0/24:

```
rtc(config)# router bgp 65001
rtc(config-router)# bgp router-id 192.168.10.1
rtc(config-router)# neighbor 192.168.10.2 remote-as 65000
rtc(config-router)# network 172.16.253.0/24
```

IMPORTANT: You must set the router-id first before you enable BGP interface. Otherwise, the router may advertise wrong information!

At this point, your BGP peering should be up and running. To verify this, you can run `show ip bgp summary` command, you should see a neighbor 192.168.10.1. Look at the last field (`State/PfxRcd`), if you see `Idle` or `Active`, it means that your BGP peering is still down. In this case, you will need to check your configuration and correct it accordingly. You will also try to check the routing reachability with ping command according to the instruction in the table below. If ping works, mark the **OK** box. Otherwise, you need to troubleshoot your configuration!

PING TEST	**OK**	**NOT OK**
From rta, ping 192.168.10.1		
From rta, ping 172.16.253.1		
From rta, ping 172.16.253.2		
From rta, ping 172.16.254.1		
From rta, ping 172.16.254.2		
From rta, ping 172.16.255.1		
From rta, ping 172.16.255.2		

After you have verified that BGP peering is up and you can ping all IP addresses in the table above, answer questions below.

Question 9: Run a command `show ip bgp summary`. Look at the last field (`State/PfxRcd`), you should see a number. What do you think this number means?

Question 10: Run a command `show ip bgp` to show BGP routing entry. You should see three routes learned from BGP peers listed below. Identify which ASN these routes are originated from?

Question 11: Run a command `show ip bgp` to show BGP routing entry. At the end of BGP routes learned from the BGP peer, you see some ASNs and a letter `i`, can you explain what do they mean?

Question 12: From hosta, can it ping 1) 172.16.255.2 and 2) 192.168.10.1?

Question 13: From rtb, can it ping 1) 172.16.255.2 and 2) 192.168.10.1?

Question 14: From the question above, if rtb cannot ping, can you explain why it didn’t work?
HINT: check the IP routing table on the routers

8 Routing between BGP and IGP

Typically, routers inside an AS may not run BGP and have no routing information to reach external networks outside your AS. In order to make these non-BGP routers communicate with the outside world, you can use one of the two methods: 1) inject BGP routes into IGP, and 2) following default routes inside an AS. In this lab, you use the second method.

It is impractical to inject all BGP routes into IGP since there may be hundred-thousand active routes in BGP. Thus, it is more common for the non-BGP routers inside an AS to follow the default routes within the AS to the closest BGP border router that can get your traffic to the external networks.

Now, you will generate a default route in OSPF from your border router (rta). This can be done by issue the command below from the OSPF configuration mode:

```
rta(config-router)# default-information originate always
```

When you are done, check that you can ping all IP address as listed in the table below. If ping works, mark the **OK** box. Otherwise, you need to troubleshoot your configuration!

PING TEST	**OK**	**NOT OK**
From rtb, ping 192.168.10.1		
From rtb, ping 172.16.253.1		
From rtb, ping 172.16.253.2		
From rtb, ping 172.16.254.1		
From rtb, ping 172.16.254.2		
From rtb, ping 172.16.255.1		
From rtb, ping 172.16.255.2		
From hostb, ping 192.168.10.1		
From hostb, ping 172.16.253.1		
From hostb, ping 172.16.253.2		
From hostb, ping 172.16.254.1		
From hostb, ping 172.16.254.2		
From hostb, ping 172.16.255.1		
From hostb, ping 172.16.255.2		

After you have verified that you can ping all IP addresses above, answer questions below.

Question 15: Run a command `show ip route` to show IP routing table on both rta and rtb. Compare and identify which routes are different. What is the IP address prefix of the new route that rtb learned that rta does not have in its routing table?

Question 16: On rtb, run a command `show ip ospf route` to show OSPF routing table. Observe the entry for the default route, what do you think E2 mean?

After you answer all questions, save the configuration on both rta and rtb. You should see the configuration is saved into three files as shown below:

```
rta# write memory
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
Configuration saved to /etc/quagga/ospfd.conf
```

```
Configuration saved to /etc/quagga/bgpd.conf
```

IMPORTANT: Before you save the configurations, make sure to check with `show running-config` that the configurations are correct. All mistakes must be fixed before you save the configurations.

9 Lab submission

For lab assessment, you will submit all configurations of hosts and routers. Therefore, you need to save your configurations and create a submission file that must be a compressed tarball containing eight files below:

- rta configurations (3 files below)
  - zebra.conf.rta
  - ospfd.conf.rta
  - bgpd.conf.rta
- rtb configurations (2 files below)
  - zebra.conf.rtb
  - ospfd.conf.rtb
- hosta.conf
- hostb.conf
- group.txt

9.1 Creating a submission file

Open a new terminal and SSH to the routing_lab VM. You should be at the home directory of the user `student`.

Create a new folder called `submission` and move into it as shown below:

```
student@ubuntu:~$ mkdir submission
student@ubuntu:~$ cd submission
```

9.1.1 Prepare router configuration files

You will fetch router configurations from rta and rtb containers and save them to new filenames that are appended with the router name as shown below:

Copy rta configuration:

```
sudo lxc file pull rta/etc/quagga/zebra.conf zebra.conf.rta
sudo lxc file pull rta/etc/quagga/ospfd.conf ospfd.conf.rta
sudo lxc file pull rta/etc/quagga/bgpd.conf bgpd.conf.rta
```

Copy rtb configuration:

```
sudo lxc file pull rtb/etc/quagga/zebra.conf zebra.conf.rtb
sudo lxc file pull rtb/etc/quagga/ospfd.conf ospfd.conf.rtb
```

By default, the files are owned by the user `root`. You need to change the ownership of the files to user `student` with the command:

```
sudo chown student:student *
```

Check that the files are readable and double-check that there is no mistake in your configuration files.

9.1.2 Prepare host configuration files

You will write down the Linux commands you use to configure the hosts to files, one file per host. `hosta.conf` for hosta configuration and `hostb.conf` for hostb configuration. You can use either `vi` or `nano`. For example, you can create the file for hosta configuration with either `vi hosta.conf` or `nano hosta.conf`.

Each file should contain two lines: the first line is the Linux commands you use to configure the IP address of the host and the second line is the Linux commands you use to the default route for the host.

9.1.3 Prepare group.txt file

You will create `group.txt` file that has one line containing your group number. For example, if you are group 252, the file should have one line containing a number 252, as shown in an output below:

```
student@ubuntu:~/submission$ cat group.txt
252
```

9.1.4 Create a compress tarball

At this point, you should have 8 files (zebra.conf.rta, ospfd.conf.rta, bgpd.conf.rta, zebra.conf.rtb, ospfd.conf.rtb, hosta.conf, hostb.conf, and group.txt) in the submission folder. You will create a compressed file of the submission folder for submission by doing the following:

On the SSH terminal, you should still be at the submission folder. Check that you have 8 files in the folder with the command:
`ls`

Verify that all files are readable. If you have any permission issue, you can try to change the ownership of the file with the command:
`sudo chown student:student *`

Change directory to the home directory with the command:
`cd`

Create a compressed file of the submission folder using the command:
`tar zcvf submission.tgz submission`

9.1.5 Check the compress tarball

You can verify that the compressed tarball is properly created and the configurations are correct by running the command below from the student home directory on the SSH terminal, where `<NUM>` is your assigned number:
`./check_routing_lab.sh <NUM>`
The script expects that the `submission.tgz` file is in the home directory.

If there is an error, you must troubleshoot and fix it accordingly. Then, you will need to repeat the process to create the compress file again.

You can upload the `submission.tgz` file to Canvas once your compressed file pass the check script.

From your host computer, you can use the scp command below to copy the submission.tgz file to the current location on your terminal:

```
scp -p student@192.168.56.2:submission.tgz .
```

10 Last step

You have completed the lab! Now, you can exit from the containers and shutdown the routing_lab VM.

Exit from the Linux host containers: (hosta and hostb)

- Type `exit` to log out.
- Press: `<ctrl>+a q` to detach from the console
- You will fall back to the SSH session to the routing_lab VM
  Type `exit` to terminate the SSH session from the routing_lab VM
- You will fall back to the terminal window on your own laptop
  Type `exit` to close the terminal.

Exit from the router containers: (rta and rtb)

- Type `exit` to exit from vtysh back to the main console
- Then, follow the steps for exit from the Linux host containers

Shutdown the routing_lab VM:
If you have a log in session to the routing_lab VM, you may use a command `sudo shutdown -h now` to shutdown the VM.

Alternatively, you can simply close the main routing_lab VM window to shutdown the VM. In this case, a window may pop-up asking you to choose some alternatives. You can just press Enter or click OK to take the default option to power off the VM.

NOTE: If you choose to save the machine state instead of power off, you may have a problem with the console access next time you run the routing_lab VM. In this case, you may have to run a command `sudo killall lxc` from the routing_lab VM shell to terminal active console processes before you can access the console of the container again.

11 Linux/Quagga Commands

11.1 Introduction

This document lists a number of specific routing protocol commands that can be useful in the routing lab. However, it is not at all a complete reference guide. For this, please consult Quagga documentation, in which you can find all commands you want (and some more).

There is one section per topic which should list most of the commands that are required to complete the corresponding lab.

11.2 Linux hosts command references

The Linux hosts are Ubuntu containers running inside the routing_lab VM. Although they are not routers, the reference commands related to Linux network configuration are included here for completeness.

`ifconfig <interface> [up|down]`

> Activates/deactivates an interface, for example:
> `ifconfig eth0 up`

`ifconfig <interface> <address> netmask <netmask>`

> Assigns an IP number to an interface, e.g.
> `ifconfig eth1 192.168.1.2 netmask 255.255.255.0`

`route add default gw <gateway ip address>`

> Assigns a default gateway to a specific address, e.g.
> `route add default gw 192.168.1`

`route del default gw <gateway ip address>`

> Remove a default gateway to a specific address, e.g.
> `route del default gw 192.168.1`

`route -n`

> Shows the routing tables in the hosts.

`sysctl -a`

> Show all system configuration variable settings.

`sysctl <variable>`

> Show a specific system configuration variable.

`sysctl net.ipv4.ip_foward`

> Show Linux IPv4 forwarding flag.

11.3 Quagga basic commands

This section describes some general Quagga commands.

`configure terminal`

> Enter configure mode.

Config command: `interface <IFNAME>`

Enter interface configuration of a specified interface

Interface command: `ip address A.B.C.D/M`

Set an IP address of the interface to A.B.C.D/M

`hostname <NAME>`

Set hostname of the router.

`ping <DESTINATION>`

Ping another host

`write memory`
`copy running-config startup-config`

Save current configuration to default configuration files.

`show interface`

Show interface status and configuration

`show ip route`

Show IP routing table

`show running-config`

Show operating configuration

`terminal monitor`

Enable logging output to vty terminals.
However, this command does not work with the current vtysh. You need to access the routing daemon via telnet and issue the command from there to get the loggin output.

11.4 Quagga OSPF Commands

This section describes some OSPF-related commands.

11.4.1 Operation commands

`show ip ospf database`

Show the link-state database summary.

`show ip ospf interface`

Show OSPF interface information.

`show ip ospf neighbor`

Show which routes OSPF advertises to its neighbours.

`show ip ospf route`

Show OSPF routing table.

`clear ip ospf interface <IFNAME>`

Reset OSPF connection on a specified interface

11.4.2 Configure commands

Config command: `router ospf`

Enable OSPF process (enter OSPF routing configuration mode).

OSPF command: `router-id <IP>`
Set OSPF router-id.

OSPF command: `network A.B.C.D/M area <N>`
Enable OSPF routing on an IP network on the OSPF area `<N>`.

11.5 Quagga BGP Commands

11.5.1 Operation commands

`show ip bgp summary`
Show a summary of BGP neighbor status.
`show ip bgp`
Show BGP routes.
`clear ip bgp *`
Reset all BGP peering sessions

11.5.2 Configure commands

Config command: `router bgp <ASN>`
Enable a BGP protocol process with the specified `<ASN>`

BGP command: `bgp router-id <IP>`
Set BGP router-id.

BGP command: `neighbor <IP> remote-as <ASN>`
Set a BGP peering session with a specified `<IP>` and `<ASN>`

BGP command: `network A.B.C.D/M`
Advertise a network to all neighbors