KTH/CSC

Static routing lab

Juniper version

Group Nr	
Name 1	
Name 2	
Name 3	
Name 4	
Date	
Grade	
Instructor's Signature	

Table of Contents

1 Goals

The goal with this lab is to introduce you to configuring a Juniper router and your host. You will also learn how to set up the router for static routing.

More specific goals are the following:

1. Log-in and start the Command Line Interface (CLI) on a Juniper router.
2. Understand the Juniper CLI.
3. Examining the state and the configurations of a router. Be able to save, load and rollback configurations.
4. Make static IP routing configurations.
5. Understand the limitation of static routing.

This lab instruction contains a number of questions as well as configuration milestones. You must complete the questions. At the configuration milestones, you must display a working network to an assistant.

The scenario of the routing lab is that you are managing an ISP network with a number of customers. The routers constitute your network, and the hosts are your customers. You will later create more virtual customer networks, to make this scenario more realistic.

Before you begin this lab, please consult document[1]. Ensure that you have the network map of Topology 1[2]. In this lab, you will not need the shared network in the middle of the topology map, that is, 192.168.15.0/24.

2 Preparation questions

If you have not done the preparation questions you will be asked to leave and complete the lab at another time.

Before you start the lab you study the following material:

- Read through the lab instructions.
- Use the *command reference*[1] – it contains most of the JunOS commands needed in this lab.
- The network map[2] contains the network topology for all groups. Use the one for your group.
- The lecture slides are useful as background information as is the relevant sections of the book.
- JunOS Cookbook from O´Reilly by Aviva Garrett. Available as on-line version is an excellent reference.

- The Juniper web-pages can be used as reference and use-cases. See for example: http://www.juniper.net/techpubs/software/junos/junos93/

These questions must be answered in writing before you start the lab.

1. What is the purpose of routing?

   _____

2. What is the difference between a learning bridge, a hub, and a router?

   _____

   _____

3. Explain the following terms: Route, Metric, Routing Table, Routing process, Packet and next hop.

   _____

   _____

   _____

   _____

4. Can NAT (Network address translation) be done without routing? Why/Why not?

   _____

5. Should ICMP, UDP and TCP traffic be treated the same way when routing? Why/Why not?

   _____

6. On the Linux hosts, how do you configure address 10.9.0.2/24 on interface eth0?

   _____

   _____

7. On the Linux hosts, how do you configure a route to subnet 192.168.0.0/16 to next-hop 10.10.0.1?

   _____

   __

   _____

   __

8. On the router, which CLI command shows the status of the hardware platform?

   _____

   __

   _____

   __

9. Which CLI modes are there? How do you change mode from one to the other?

   _____

   __

_____
__

10. In the CLI, after you have edited a configuration, which command do you use to make the changes take effect?

_____
__
_____
__

11. In the CLI, how do you *replace* the current configuration with a configuration stored on file?

_____
__
_____
__

12. In the CLI, how do you configure address 192.168.3.49/30 on interface fe-1/0/0?

_____
__
_____
__

13. Which command is used to show the whole routing table?

_____
_
_____
___

Milestone 1: Preparation.
Signature: _____

Figure 1: Mobile rack with four Juniper routers, one power panel and a hub.

3 The routers

The central equipment in the router lab are 20 Juniper J4300 software routers. The routers are accessed remotely and do not need to be accessed physically. They are essentially PC:s equipped with six Ethernet interface cards with limited forwarding capacity in terms of performance, but without limitations in terms of functionality. The operating system is *JunOS,* a variant of the UNIX BSD operating system. The routers can be configured via html, or

via XML. But in the labs, the Command Line Interface (CLI) will only be used to configure the routers.

The routers are mounted in five mobile racks as shown in Figure 1. The racks are numbered by letters – the one in the figure is rack A. At the top of the rack, there is a power panel to which each router is connected. Below the power are the four routers, one above the other. At the bottom of the rack, there is an Ethernet hub.

Each rack has four routers. The front of one router is shown magnified in Figure 2. The interfaces are named as *type-x/y/z*, where *type* is the network interface type (eg, fe for "Fast Ethernet"); *x* is the FPC (Flexible PIC Concentrator) the position of the card in the router chassis; *y* is the PIC (Physical Interface Card) slot number and *z* is the port number on the card. On the J4300, all PICs are number 0, since there is only one PIC per FPC. On more advanced routers, like the M- and T- series, a FPC typically hosts several PICs.

The router also has a terminal port for serial access, and a power button.

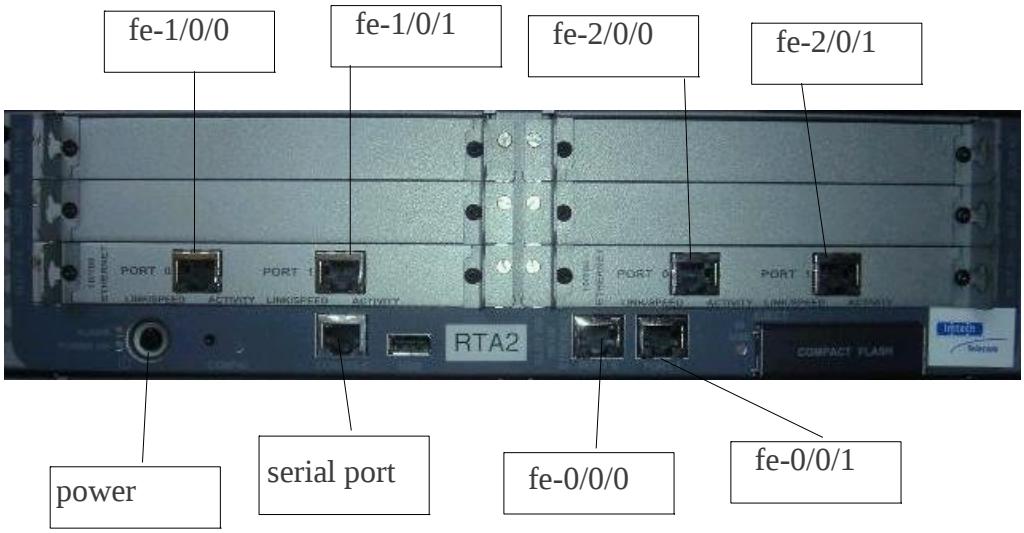

Figure 2: The front panel of one Juniper J4300 router

To connect to the routers, you use telnet to a console server. In this way, you access the routers serial port. The following table shows which address and telnet port to use to connect:

Routers	Terminal server	telnet port	user
RTA1-RTA4	terminal1.netlab.csc.kth.se	2001-2004	netusr
RTB1-RTB4	terminal1.netlab.csc.kth.se	2005-2008	netusr
RTC1-RTC4	terminal2.netlab.csc.kth.se	2001-2004	netusr
RTD1-RTD4	terminal2.netlab.csc.kth.se	2005-2008	netusr
RTE1-RTE4	terminal2.netlab.csc.kth.se	2009-2012	netusr

4 CLI tutorial

This section contains a CLI introduction in tutorial form. Go through the tutorial so that you can configure the routers properly when the “real” lab starts.

4.1 Login

In this lab, all configuration and management of the router is done with the CLI. The routers may be configured in other ways, including http and XML.

When you login to a router via the serial port, the CLI starts automatically.

By default, use the `netusr` user. The password to netusr should be given to you. Netusr is a “super-user” which means that there are no limitations on your capabilities when you configure the router.

Password to connect to your router?

4.2 User interface

Start by getting acquainted with the CLI user interface. Try the characters '?' and <TAB> and <space> to understand how help and command-completion works.

The other control-editing commands are similar to the ones found in, for example, emacs: <ctrl-b>, <ctrl-f>, <ctrl-a>, <ctrl-e>, <ctrl-p>, <ctrl-n>. Ensure that you can master these control sequences.

4.3 On-line documentation

There is on-line help on the routers. Apart from the '?' command, that gives a summary information of each command, the `help topic` and `help reference` commands gives more extensive help.

4.4 Operations mode commands

The first mode of the CLI is operations mode. In operations mode, you can examine the state of the router or perform commands, including reboot, ping or traceroute.
The following list includes a selection of commands that you should try (try the '?' to find out all sub-commands):

`show chassis`
: Show information about the router chassis: hardware, alarms, etc.

`show interfaces`
: Show information about the interfaces on the router.

`show route`
: Show IP route information. Usually extended with the command `protocol` followed by protocol name.

`show system`
: Underlying operating system status.

`monitor`
: Start a monitoring of the system for debugging purposes.

`show log`
: Display log files

`set cli`
: Set CLI properties

`restart`
: Restart a process/ software module

`request system reboot`
: Reboots the system

`ping`
: Ping another host

`traceroute`
: Traceroute to another host

Try also the 'brief', 'detail', and 'extensive' keywords after the commands.

Example:

```
netusr> show chassis hardware detail
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                               JN003057AB        J4300
Midplane         REV 05   710-010001   AD05070170
System IO        REV 07   710-010003   AE05070607        System IO
board
```

```
Routing Engine   REV 08   750-010005   BTRD45100011      RE-J.2
  ad0     244 MB  Hitachi XX.V.3.4.0.0 X0503 2004120222  Compact
Flash
FPC 0                                                    FPC
  PIC 0                                                  2x FE
FPC 1            REV 04   750-010353   AF05320272        FPC
  PIC 0                                                  2x FE
FPC 2            REV 04   750-010353   AF04452056        FPC
  PIC 0                                                  2x FE
```

What is the name of the vendor of the compact flash on your router?

_____

___

4.5 Configure mode

The second mode is called *configure* mode (or edit mode). You enter this mode by the `configure` command, and leave it by typing `exit`. In edit mode you can change the state of the router. While in configure mode, you can run operations command by prefixing the command with `run`.

The configurations in a Juniper router are central to the operation of the router. By editing a configuration, you manipulate the state of the router and program its behaviour.

You change a configuration by using the CLI configure mode. Note that the state of the router does not change until you issue the `commit` command. In other words, you can safely edit your configuration without worrying about its effects until you commit it. You can view your current (non-committed) changes with the `show` command. You can also see the difference between your configuration and the committed with the `show compare` command. Note that one of the most common problem for Juniper beginners is to forget the commit command!

Example:

```
student@RTA3> configure
Entering configuration mode
[edit]
student@RTA3# set system host-name London
student@RTA3# show | compare
[edit system]
-  host-name RTA3;
+  host-name London;
student@RTA3# commit
student@London#
```

Extra options to commit are `commit confirmed`, and `commit check`. How do these commands work?

The configuration is tree-structured. When you change a configuration, you modify the nodes of the tree. There are several ways to do this. You can make all the changes from the top-level of the configure mode using the set command. You can also use the edit command to place yourself at a specific node in the tree.

The example below shows how to set the login class using both alternatives:

```
student@RTA3# set system login user student class super-user
[edit]
student@RTA3# edit system login user student
[edit system login user student]
student@RTA3# set class super-user
[edit system login user student]
student@RTA3# show
class super-user;
[edit system login user student]
student@RTA3# top
[edit]
student@RTA3#
```

You can also use the `delete`, `add`, `rename`, and `insert` commands to modify the tree.

Try to make some configuration changes and commit them. Make some new changes and commit again. When does the routers state change?

_____

___

_____

___

4.6 Rollbacks

It is also possible to make *rollbacks* of a configuration. A rollback deletes the current configuration and installs the previous version. You can give an integer argument to the command indicating how many commits you wish to rollback.

Rollback 0, for example, purges your current (non-committed) edits, rollback 1 restores the configuration before your previous commit.

Try to rollback a configuration previously committed. What happens?

_____

___

4.7 Saving and loading

You can also save your configurations to file, and load them from file. This is an operation that is completely independent to committing configurations.

When you save a configuration, place yourself at the top and type `save`. Load a configuration by using the `load override` command. You can also merge configurations, or save/load sub-sections of your configurations.

Example:

```
student@London# top save londonconf
Wrote 67 lines of configuration to 'londonconf'
student@London# load override labconf
load complete
student@London# commit
student@RTA3#
```

For the purposes of these labs, there should be a configuration called

```
~root/labconf
```

which contains a basic configuration without any interface or protocol configurations. You can always revert to this configuration if you have made mistakes.

It is also recommended that you save configurations regularly. In particular, you should make a configuration containing the configurations for each topology you encounter in the labs. For example, after configuring topology1, you should save a configuration with that name, so that you can easily revert to that configuration when you start a new lab.

As an aside, configurations are regular files that are saved as files in the underlying operating system. It is possible to view them and edit them by starting a shell, or to transfer them via ssh to a remote host, for example.

Try saving a configuration (or part of a configuration) and load it using combinations of the different load commands: override, relative, merge, terminal.
How do these commands work?
Override:_____
relative:

_____
merge:

_____
terminal:

_____

5 Host configuration

The virtual hosts run Ubuntu Linux and are accessed using SSH. The user name on the host is student. The name of the hosts are

x1.xen.netlab.csc.kth.se, x2.xen.netlab.csc.kth.se, etc. where x is group name. The password will be provided to you at the time of the lab. To perform commands with super-user privileges, issue the sudo bash command.

The hosts have two Ethernet interfaces: eth0 and eth1. One interface (eth1) has an allocated IP address via DHCP which enables Internet access. Do not change the configuration of this interface. The other interface (eth0) is used to connect to a router. You will need to set its IP number and its netmask manually, but its physical connections are already made. Furthermore, at least one route must be added in the hosts routing table so it knows through which interface to send the IP packets to the router network.

Set up the hosts to connect both to the Internet and the private lab-net. One of your Ethernet interfaces (e.g. `eth1`) should already have a public IP address. Do not modify this interface.
Configure eth0 with IP-address and netmask.
Set the address to the IP number specified in the network topology handout. Do not forget the netmask!
You will need at least two static routes on your host so that packets destined to the 192.168.0.0/16 and 10.0.0.0/8 network will be sent to the lab net (and not to the Internet). Use the route add command.

Set up the interface fe-0/0/0 on the router.

Verify your configuration by using ping from the host to the router and vice-versa.

Milestone 2: The CLI.
Signature: _____

6 Real lab start: Static routing in pairs

First start working in pairs, RTX1 with RTX4 and RTX2 with RTX3.

Configure the interfaces between the two routers only. Do not configure any more interfaces on the router than necessary.

Try to ping your neighbour's host from your router, then from your host. What happens and why?

Now add a static route on your router to your neighbour's host network. You should now be able to `traceroute` from your host all the way to your neighbour's host.

Investigate the state of your configured network interfaces. Which mac-address do they have? What is the media status?

Which routes are there? Write down and explain every route that you see using the `show route` command. Ensure that you understand everything in the output, including type of route, age, preference, activity and next-hop.

Try to disable the interface (between the routers) and see what happens to the routing table.

Milestone 3: Show a working pair configuration.
Signature: _____

7 Static routing in quad

Now connect all four routers in the group by configuring the interface to your other neighbour. Make the connection as a “square”: do not use the shared 192.168.15.0/24 network.

Configure static routing so that all routers and all hosts can reach each other. But create only one route to your diagonal customer network and make the routing symmetric. This step may take some time to complete. Remember to cooperate with your neighbours in a “friendly” way.

List the extra routes that have been added in the quad set-up

_____

_____

_____

_____

Verify with ping, traceroute and/or ping record-route. Both from hosts and routers to all other hosts and routers.

8 Fault tolerance

You should now try to build a fault tolerant network using static routing. Add a second static route with a higher metric to the diagonal customer network. This is called a “floating” static route.

Disable one of the interfaces in order to make the secondary route take over. Can you still reach your diagonal neighbour? Explain your result.

_____

_____

_____

Milestone 4: Floating static route.
Signature: _____

9 Asymmetric routing

Create asymmetrical static routing to your diagonal customers. Verify with traceroute or record-route ping.

10 Load balancing

Make the cost equal between the two links to the diagonal customer network. Traceroute (or ping record-route) to the opposite host. Make the router load balance between the two links.

Does load balancing work? Why/why not?

_____

_____

What is required do to make the pings load balance?

_____

_____

Milestone 5: Asymmetric routing and load balancing
Signature: _____

11 Clean-up

Reset the configuration by loading (using override) the /root/labconf configuration.
Also do not forget to log out from your hosts.

12 References

[1] KTH/CSC Router lab Reference - Juniper version
[2] KTH/CSC Router lab Netmap - Topology 1