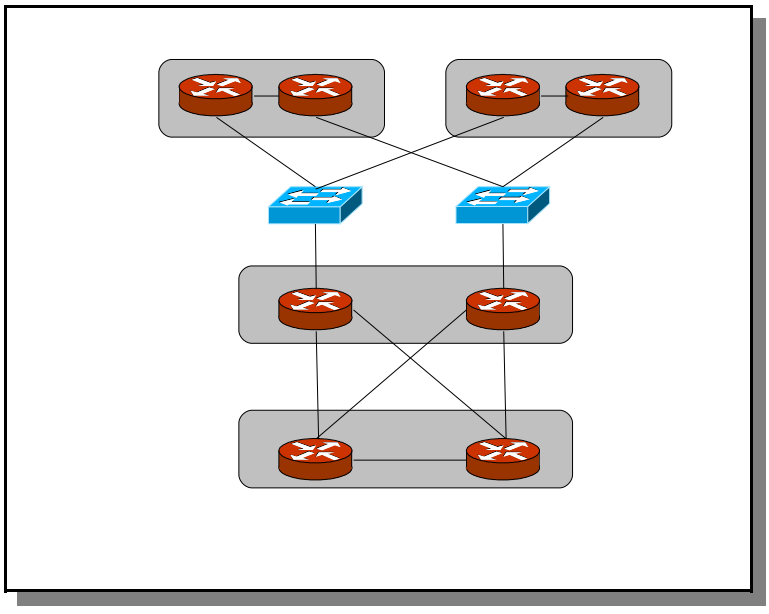
KTHNOC

Introduction to Routing



Introduction laboration to JunOS and dynamic routing protocols

Table of contents

1 Goals, background and overview

The goal of this lab is to give an introduction to dynamic routing and Juniper routers. After the lab, you should be able to configure the command-line interface of a Juniper router in order to make a simple dynamic routing set-up. The network is simple and not realistic, but provides an insight into several important routing issues.

The lab starts with a CLI tutorial and continues with the configuration of OSPF on a pair of routers representing a minimal interior network. The lab ends with configuring BGP between two such network islands.

The lab is five hours and it is important that you keep a good speed.

The lab consists of a relatively long introduction including a CLI tutorial. The “real” lab starts in Section 7 when OSPF and BGP is configured. Try to not spend more than two hours in the introduction part of the lab leading up to Section 7.

It is necessary to make the preparations before coming to the lab. Otherwise, it will be very difficult to complete the lab in time.

2 Preparation Questions

If you have not done the preparation questions you may 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 Section 11 as a command reference – it should contain most of the JunOS commands needed in this lab.
- Appendix A 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.
- You should have solved the routing recitation and homework.
- Section 12 describes how policy options are used in this lab and is useful when you design the policies in the last lab-part.
- 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/junos82.

2.1 Preparation questions

Please answer the following questions. You should find answers to most of the questions by reading this lab instruction.

1. On the Linux hosts, how do you configure address 192.168.3.226/27 on interface eth0?

   _____
   _____

2. On the Linux hosts, how do you configure a route to subnet 192.168.0.0/16 to next-hop 192.168.3.225?

   _____
   _____

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

   _____
   _____

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

   _____
   _____

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

   _____
   _____

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

   _____
   _____

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

   _____
   _____

8. Which command is used to show the whole routing table? For OSPF routes only?

   _____
   _____

9. How do you check which neighbours an OSPF peer has?

   _____
   _____

10. How do you monitor the OSPF protocol?

    _____
    _____

11. What is a router-id? On what interface should you normally set such an address?

    _____
    _____

12. Using the group A topology map in Appendix A, which two prefixes should be announced by AS 65001 to AS 65002?

_____
_____

13. What is an aggregate route?

_____
_____

14. How do you create an aggregate route in the CLI?

_____
_____

15. How do you check which routes are announced to another BGP peer?

_____
_____

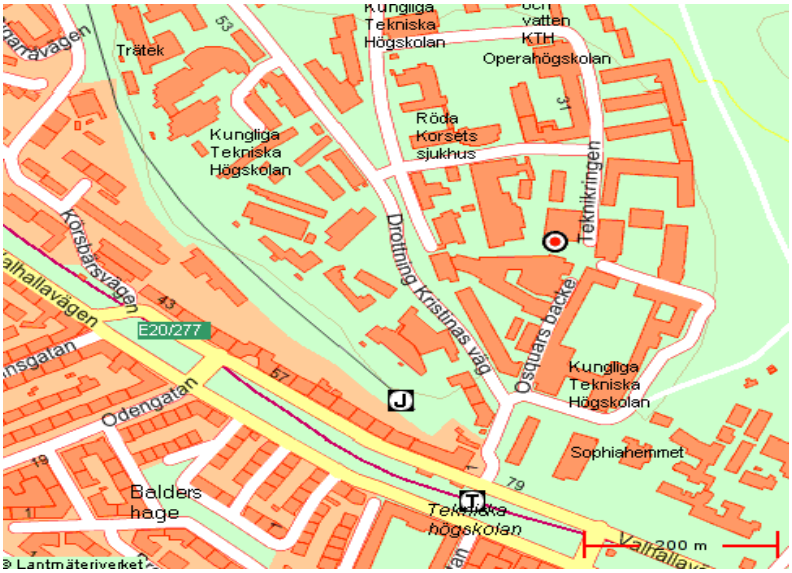
16. Try to complete the policy option in Section 9 for group A – the one used to announce the prefixes from AS 65001.

_____
_____
_____
_____
_____
_____

17. How do you activate/apply a policy option in BGP?

_____
_____

3 The router LAB – Teknikringen 14 / SAM



The router lab in SAM is used to conduct most of the labs at KTHNOC. SAM is located on the second floor at Teknikringen 14. The map above shows where Teknikringen 14 is located. There are several terminal rooms (Fylke, Merry, Pippin, etc). You will have to pass all other terminal rooms before you come to SAM.

You also need access to the “Fylke-salar” – if you have registered for the course you will have temporarily access to the rooms.

In SAM, there are five tables with four computers on each table. These terminals are placed in a closed rack outside SAM in the corridor. The terminals boot with windows and your KTH login should work to login.

The terminals are only used to connect to the hosts and routers of the lab. The hosts are *virtual*, meaning that they run as separate processes within a Xen-server. The routers are physically placed in the room “Mordor” outside SAM.

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

4 The routers

The central equipment in the router lab are 20 Juniper J4300 software routers. 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.

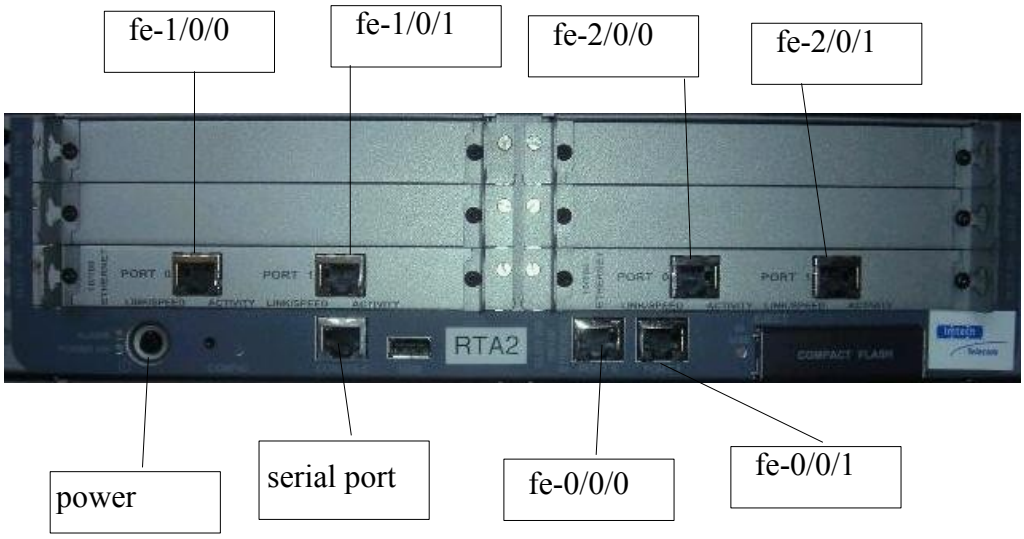


Figure 2: The front panel of one Juniper J4300 router

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:

Group	Routers	Terminal server	telnet port	user
A	RTA1-RTA4	192.71.24.7	2001-2004	netusr
B	RTB1-RTB4	192.71.24.7	2005-2008	netusr
C	RTC1-RTC4	192.71.24.8	2001-2004	netusr
D	RTD1-RTD4	192.71.24.8	2005-2008	netusr
E	RTE1-RTE4	192.71.24.8	2009-2012	netusr

5 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.

5.1 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.

5.2 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.

5.3 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.

5.4 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?

_____

5.5 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?

_____

_____

5.6 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?
_____

5.7 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: _____

Milestone 1: The CLI.
Signature: _____

6 The hosts

The virtual hosts run SUSE linux and are accessed using ssh. The username on the host is `laban`. The name of the hosts are

a1.lab.noc.kth.se, a2.lab.noc.kth.se, etc. The password will be provided to you at the time of the lab. To perform commands with super-user privileges, issue the `sudo` 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.

Configure eth0. Set the address to the IP number specified in the network topology handout. Do not forget the netmask!

You will need at least one static route on your workstation so that packets destined to the 192.168.0.0/16 network will be sent to the lab net (and not to the Internet). Use the `route add` command.

On the router, enter configure mode and configure an interface address on fe-0/0/0 on the router according to the network topology map.

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

7 Real lab start: Interface configuration

First start working in pairs, RTX1 with RTX4 and RTX2 with RTX3. You can use the commands in Section 13.4. Use the netmap topology in Appendix A for your group.

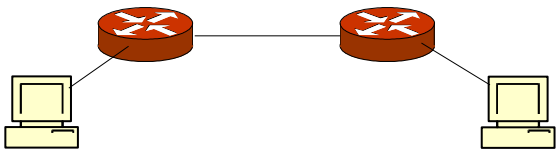


Figure 3: Router pair

First ensure that the state of your routers is clean. If not, use the `load override ~root/labconf` to load an empty state. Thereafter, set the correct hostname.

Connect and bring up the network between the workstation and the router. When you configure the router, *do not configure any more router interfaces than required for a pair*. Configure the workstation with the IP addresses according to the topology map. Do not configure the 10.* networks yet.

Your interface configuration should look something like the following:

```
interfaces {
    fe-0/0/0 {
        unit 0 {
            family inet {
                address XXXX/XX;
            }
        }
    }
    fe-1/0/X {
        unit 0 {
            family inet {
                address XXXX/XX;
            }
        }
    }
}
```

Ensure that you can ping the directly connected interfaces: The router from the workstation, and between routers.

Investigate the state of the router. In particular note all routes and be sure that you have understood which routes exists in the routing table. Explain each route and its fields:

1. Which types do they have (protocol)?
   _____
2. What is the difference between a local and direct route?
   _____
3. How long have the routes been installed?
   _____
4. What does '*' mean when it appears in the beginning of some routes?
   _____

Try to explain what happens with a ping packet that is sent from a workstation to the remote router. Use the routing tables to understand what happens to the packet. Is the packet dropped? If so, where?

_____
_____

Try to ping between the work-stations. Does it work? Why/why not?

_____

Milestone 2: Statically configured pair.
Signature: _____

8 Dynamic routing with OSPF

Start with setting the router-id. Use the address given in the router boxes in the netmap. For example, RTA1 should have the routerid 192.168.1.1.

Assign this address to the loopback interface (By default, OSPF will use the first lo0 address as router-id).

Turn on OSPF on the interfaces connecting you to your neighbour using area 0, the backbone. Also turn on OSPF on the interface to the workstation (fe-0/0/0). But since we do not expect an OSPF router to appear on the customer network – declare it as passive. That is, do not run the OSPF protocol, but announce it as an internal OSPF network. Also declare lo0 as passive.

Your OSPF configuration will look something like this:

```
routing-options {
    router-id XXXX;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface fe-1/0/X;
            interface fe-0/0/0 passive;
            interface lo0.0 passive;
        }
    }
}
```

Try to ping between your workstations. Also try to ping all the routers addresses, including the loopback addresses.

Check OSPF neighbours (`show ospf neighbor`) and interfaces (`show ospf interface`) to verify that OSPF runs correctly.

Examine the routing table again. What is the value of the route preference of the different routes? What does this value mean?

_____

_____

Try looking only on the OSPF routes (`show route protocol ospf`).

Examine the OSPF internal routing table (`show ospf route`). What is the difference with the previous command?

Check the routing table (`show route protocol ospf`). What is the difference with the previous command? Do you see any OSPF routes? Why/ why not?

_____

_____

Traceroute or use ping record route between the workstations. Verify that the path is correct.

_____

_____

Also examine which router is DR (Designated Router), and which is BDR (Backup DR) on the shared network between the routers.

_____

Milestone 3: Show a working pair in OSPF.
Signature: _____

8.1 Monitor OSPF*

When debugging and monitoring a protocol, you first set traceoptions for that protocol, and then view the traces with the monitor command.

The following example creates a file (ospfdbg) with a 5Mbyte limit, makes detailed OSPF routing traces to that file, and then views the trace from the terminal.

Example:
```
student@RTA3# set protocols ospf traceoptions file ospfdbg size 5M
student@RTA3# set protocols ospf traceoptions flag route detail
student@RTA3# run monitor start ospfdbg
```

When monitoring, it is a good idea to create several windows: one for monitoring and one where you make CLI commands.

Instead of doing an active monitoring you can also examine the log-file:
```
student@RTA3# run show log ospfdbg
```

8.2 Adding more networks

To make the lab more interesting, you will now include more routes. Use the 10.X.Y.0/24 networks that are given in the netmap.

Configure these network as secondary addresses on fe-0/0/0. They will appear automatically as OSPF stub networks.

```
interfaces {
    fe-0/0/0 {
        unit 0 {
            family inet {
                address 10.X.X.1/24;
                address 10.X.X.1/24;
                address 10.X.X.1/24;
                address 10.X.X.1/24;
            }
        }
    }
```

When this is done, look in the other router and investigate that these routes appear.

Milestone 4: More OSPF.
Signature: _____

9 BGP: Stub AS

You should now have configured IGP on both pair of routers. This represents two AS:s. You will now use BGP to interconnect these two domains.

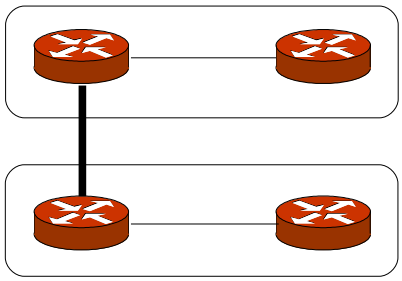


Figure 4: A pair of AS:s connected by one eBGP peering

Connect and configure the link between RTX1 and RTX2 as shown in the figure. Run OSPF (as before) internally in both ASs.

Set your AS number as given in the topology map using `routing-options autonoumous-system`.

Configure an external BGP group on the border routers RTX1 and RTX2 as shown in Figure 4.

Your basic EBGP configuration should look something like the following:

```
protocols {
      bgp {
            group EXTERN {
                  type external;
                  family inet {
                        any;
                  }
                  peer-as XXX;
                  export MYNETWORK; # later
                  neighbour 192.168.X.X;
            }
      }
}
```

Ensure that the link between the AS:s is distributed into OSPF. Why?

_____

_____

The easiest way is to declare the interface in OSPF as passive.

Now you need to define which prefixes to export to the other AS. These prefixes should cover your whole AS, not only your router. The basic idea is to use policy-options and export them with BGP.

One way to do this is to create aggregate routes for all prefixes you wish to export (you should export two prefixes):

```
set routing-option rib inet.0 aggregate route X.X.X.X/X
set routing-option rib inet.0 aggregate route X.X.X.X/X
```

Then you need to define policies to export the aggregate routes. When you apply the rule in BGP, ensure that you apply it to the specific group or neighbours where it should apply. That is, try to place the export statement at the correct level in the BGP configuration tree.

The export policy looks something like:

```
policy-options {
      policy-statement MYNETWORK {
            term 1 {
                  from {
                        protocol aggregate;
                        route-filter X.X.X.X/X exact;
                  }
                  then accept;
            }
      }
}
```

When you have done this, ensure that you can communicate between all addresses on your peer. Check which routes are are announced and received with your peer.

But you still cannot communicate externally with the router not running BGP. Why?

_____

_____

You now need to announce a default route with the IGP, so that your inner routers not speaking BGP can communicate with the outside world.

Create a new aggregate *default* route and export this via OSPF. Set an appropriate metric as well. You need also to create a policy for this and to apply it to OSPF. This policy option looks something like the following:

```
policy-options {
      policy-statement DEFAULT {
            term 1 {
                  from {
                        protocol aggregate;
                        route-filter 0.0.0.0/0 exact;
                  }
                  then {
                        metric YY;
                        accept;
                  }
            }
      }
}
```

Remember to activate the policy into OSPF by using the `export` statement.

When you are done, check that you can ping between all interfaces, including the customer's addresses.

Investigate the routing tables. Explain all routes from the different protocols.

Milestone 5, Stub BGP

Signature: _____

10 Cleanup

To reset the machines to their initial state, follow the following steps for all routers.

1. Reset the configuration by loading (using override) the ~root/labconf configuration.
2. Logout from the routers.
3. Logout from the virtual hosts.

11 REFERENCE: JunOS / Linux Commands

11.1 Introduction

This document lists a number of specific routing protocol commands that can be useful in the routing labs. However, it is not at all a complete reference guide. For this, please consult Juniper web-sites, where 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 Reference: Virtual hosts

The hosts simulate customers of the networks. They run SUSE Linux and needs to be configured as well. Although not routers, the reference commands for the workstation are included here for completeness.

11.2.1 Configuring network access

`ifconfig <interface> [up|down]`
Activates/deactivates an interface, for example:
`ifconfig bge0 up`.

`ifconfig <interface> <address> netmask <netmask>`
Assigns an IP number to an interface, e.g.
`ifconfig bge1 193.11.23.12 netmask 255.255.255.0`

`route add -net <network> netmask <netmask> gw <dst ip number>`
Tells the host to route packets to a certain destination. To assign the default route to use next-hop 193.11.23.1:
`route add -net 0.0.0.0 netmask 0.0.0.0 gw 193.11.23.1`,
to route all traffic to the lab network via 192.168.8.X:
`route add -net 192.168.0.0 netmask 255.255.255.0 gw 192.168.8.X`

`netstat -nr`
Shows the routing tables in the hosts.

`sysctl -a`
Show all system configuration variable settings.

`sysctl <variable>=<value>`
Assign a specific system configuration variable.

`sysctl net.inet6.ip6.accept_rtadv=1`
Enable IPv6 router advertisements.

11.3 Basic commands

This section describes some general commands.

11.3.1 Operation commands

`configure`
`edit`
Go to configure mode

`file`
Range of commands about files on the system.

`help topic`
Show on-line help about general subjects.

`help reference`
Show on-line help on commands.

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

`show interfaces [terse|bried|detail|extensive]`
Show information about the interfaces on the router.

`show route`
Show IP route information.

`show route protocol [static|direct|local|ospf|...]`
Show IP route information of specific protocol.

`show system`
Underlying operating system status.

`show storage`
Show filesystem information.

`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

`request system halt`
Halts the system

`ping`
Ping another host

`ping [record-route | source]`
Ping another host using record-route| specify a source address.

`traceroute`
Traceroute to another host

11.3.2 Configure commands

`set`
Set a configuration parameter

`delete`
Delete a configuration parameter

`edit`
Edit a sub-element

`top`

Go to top of configure mode

`exit`

Leave configure mode and go to operations mode

`load override <filename>`

Load a configuration from file. Replace the old configuration with the one on the file.

`load merge <filename>`

Load a configuration from file. Merge the file with the current configuration.

`load merge terminal [relative]`

Load a configuration from the terminal. In this way, you can cut and paste configurations without using the `set` command.

`rollback [<number>]`

Rollback configuration to last step (or specified number of steps).

`set file <filename> interactive-commands any`

Log all CLI commands on file.
steps).

`save <filename>`

Save a configuration to file

`commit [check]`

Activate the configuration. If the check option is used, all run-time sanity checks will be made, but the configuration will not actually be committed.

`commit [confirmed]`

Activate the configuration for a limited time (10 minutes). If you do not make a new commit within this time, a rollback to the old state is performed.

`show`

Show the current configuration. May be non-committed. Keywords can be any part of the configuration, such as `system`, `protocols`, etc.

`show | compare [filename]`

Show the difference between the current configuration being edited and the committed state of the router.

`show | display set`

Show the configuration using `set` commands. Useful if you want to cut and paste commands.

`set interfaces <ifname> unit 0 family inet address <prefix>`

Set an interface address

`set routing-options static route <prefix> next-hop <ip-address>`

Create a static route

`set routing-options router-id <address>`

Set id of this router. Recommended to be a routable address.

`set routing-options forwarding-table export <name>`

Apply a policy on FIB (can be used for load-balancing)

`set policy-options policy-statement <name> then load-balance per-packet`

Set a policy for load-balancing to install all next-hops with equal cost. Need also to apply this to RIB->FIB translation

`set forwarding-options hash-key family inet layer-3`

Force load-balancing (if enabled) to use only IP addresses

`set forwarding-options hash-key family inet layer-4`

Force load-balancing (if enabled) to use also layer -4 info, including ports and incoming interfaces. You must also specify layer-3 for this to take effect.

11.4 OSPF Commands

11.4.1 Operation commands

`show ospf database`
: Show the link-state database. Useful extensions are `router`, `network` and `extern`, for example. In the output, (*) indicates that the LSA is generated by your router. Also, you can append the usual externsions: detail or extensive if you want to have more information.

`show ospf database advertising-router <id>`
: Show which LSA:s are announced by a specific router.

`show ospf database | match router | count`
: Count the number of routers

`show ospf neighbor`
: Show which routes OSPF advertises to its neighbours

`show ospf interface`
: Show which routes OSPF advertises to its neighbours

`show ospf statistics`
: Show counters and other statistics related to OSPF

`show ospf route`
: Show OSPF routes (with OSPF info)

`show route protocol ospf`
: Show OSPF routes in the routing table

`restart routing gracefully`
: Restart the routing (in a nice way)

`clear ospf neighbour`
: Clear the OSPF neighbour adjacencies

`clear ospf database purge`
: Clear the OSPF database (locally)

11.4.2 Configure commands

`set routing-options router-id <address>`
: Set id of this router. Should be a routable address.

`set protocols ospf area <nr> interface <interface>`
: Turn on OSPF on an interface.

`set protocols ospf area <nr> interface <interface> interface-type <type>`
: Change interface behaviour, (nbma, p2mp, p2p).

`set protocols ospf area <nr> interface <interface> passive`
: Do not run OSPF on an interface, but announce its network.

`set protocols ospf area <nr> stub`
: Set an OSPF stub area.

`set protocols ospf area <nr> stub default-metric <nr>`
: Inject default route into stub area.

`set protocols ospf area <nr> stub no-summaries`

Stop LSA-3 into area. (totally stub)

`set protocols ospf area <nr> nssa`

Set an OSPF not-so-stubby area.

`set protocols ospf area <nr> nssa`

Set an OSPF not-so-stubby area.

`set protocols ospf traceoptions file <name>`

Create a log file for tracing of OSPF events

`set protocols ospf traceoptions flag hello detail`

Turn on detailed tracing of OSPF. Other traceoptions include `error` and `lsa-update`.

`set policy_options policy_statement <name> then metric <nr>`

Set metric in a policy. Useful when exporting external routes, for example.

11.5 BGP Commands

11.5.1 Operation commands

`show bgp summary`

List all BGP peers with summary information.

`show route <prefix> detail`

Display detailed information for a given prefix.

`show route receive-protocol bgp <address>`

Display routes received by a peer before policy is applied.

`show route advertising-protocol bgp <address>`

Display routes advertised to a specific peer.

`show route receive-protocol bgp <address>`

Display routes received by a peer before policy is applied.

`clear bgp neighbor <ip> [soft]`

Display routes received by a peer before policy is applied.

`monitor [start|stop] <file>`

Start/Stop monitor output from file to terminal.

11.5.2 Configure commands

`set routing-options router-id <ip>`

Set router id.

`set routing-options autonomous-system <ASN>`

Set AS Number.

`set protocols bgp local-as <ASN>`

Set AS Number.

`set protocols bgp export <policy>`

Set export rules.

`set protocols bgp group <group> peer-as <ASN>`

Set peer AS Number.

`set protocols bgp group <group> type [internal|external]`

Set external/internal type.

`set protocols bgp group <group> neighbor <peer-ip>`

Set peer ip.

`set protocols bgp group <group> cluster <cluster-id>`

Configure route reflection.

`set protocols bgp traceoptions file <file>`
File for bgp monitoring.
`set protocols bgp traceoptions flag [update|open|all|...] [detail]`
Set traceoptions flags for monitoring bgp details.
`set protocols bgp advertise-inactive`
Make BGP advertise routes even if they are not active in the local routing table (because IGP is preferred).

11.5.3 Policy options

`edit policy-options`
All following commands in this edit mode.
`set prefix-list <name> <prefix>/<length>`
Create a prefix list.
`set policy-statement <name> from <match> then [accept|reject]`
Create simple policy statement.
`set policy-statement <name> term <name> from ...`
Create policy statement.
`set policy-statement <name> term <name> then ...`
Continue policy statement.
`set as-path <name> ...`

12 Policy-options

Policies are central when configuring protocols in the Juniper routers. With policies you control how routes are redistributed between protocols and routing tables. The routing protocols are somewhat different in their default behaviour: some export direct routes by default for example, while others have to be explicitly configured.

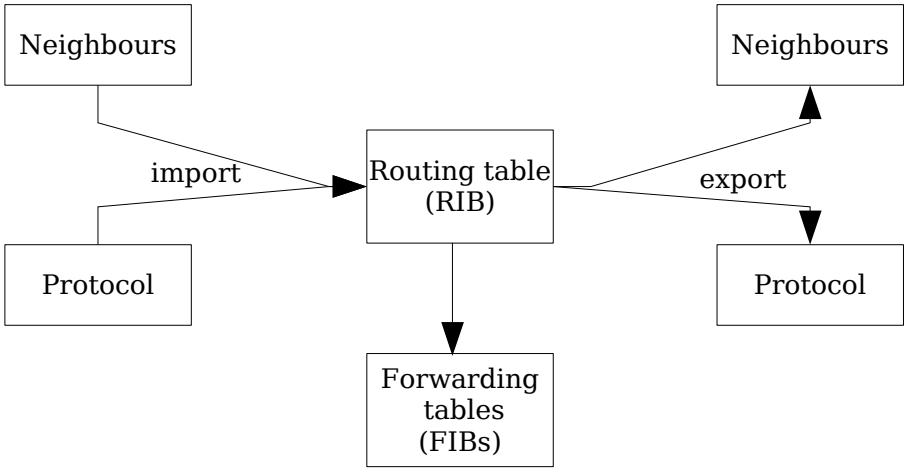


Figure 5 Exporting and importing routes: basic model

First of all it is important to understand the underlying model of importing and exporting routes. Figure 1 shows the underlying policy model in a Juniper router. We will primarily be concerned with the export of routes to a routing protocol.

The routes are grouped according to “protocol”, as follows:

- direct – directly connected networks
- static – manually configured routes
- aggregate – route aggregates configured in the router
- local – local system addresses
- rip – routes learned by RIP
- ospf – routes learned by OSPF
- isis – routes learned by ISIS
- …

By default, only active routes may be exported – that is, routes that are in the routing table. Also, in order to export routes, you need to specify a routing policy, which specifies a set of rules that applies to the routes. Third, you need to apply this policy to your protocol. To override this default rule, the special advertise-inactive command must be issued.

12.1.1 Policy-statements

You create a policy by entering *policy-statements*. A policy-statement has a *match-part* and an *action* part. The match-part specifies which routes the policy applies to, while the action part specifies what actions to perform for these routes.

The following example is a simple policy called “allstatic” that specifies that all static routes should be accepted:

Example :
```
policy-options {
    policy-statement allstatic {
        from protocol static;       # match
        then accept;                # action
    }
}
```

The example above is single-termed. More complex policies can be *multi-termed* so that a chain of policy terms can be expressed. In other words, one policy-statement contain several terms that are linked together.

Further, more elaborate expressions can be expressed with *route-filters*. Route-filters are used to express subset of routes that the policies applies to.

The following example shows a policy statement with a route-filter:
```
policy-options {
    policy-statement multi {
        term 1{
            from {
                  protocol static;
                  route-filter  192.168.0.0/16  prefix_length-range  /
30-/32 ;
            }
            then accept;
        }
    }
}
```
The policy accepts all static routes that are /30-/32 prefixes in the 192.168.0.0/16 range.

Another example shows a policy that accepts one specific route only (a directly connected network):
```
policy-options {
    policy-statement exact {
      from {
            protocol direct;
            route-filter 192.168.0.0/24 exact;
      }
      then accept;
    }
}
```

12.1.2 Exporting policies

Finally, the policy needs to be explicitly applied to a protocol, which is made by the `export` command.

Example:
```
protocol {
      rip {
            export multi;
      }
}
```

This means that the RIP protocol will export all static routes (as specified by the multi policy) into RIP, and advertise those routes to its neighbours.

13 Firewalls

Firewall rules contains packet filters and traffic policing. Defining firewall rules is similar to the policy_statments described in Section 7.

You use packet filter rules to identify a class of traffic, and then to apply actions to all packets belonging to the class. When the rule is defined, it is activated by associating it with an input or output filter rule in an interface.

You can associate the rules either to traffic being forwarded using external interfaces. Alternatively, you can protect the route processor from incoming traffic (e.g., DOS attacks) by associating filters to the loopback address.

13.1 Interface syntax

An example of how the association of a filter rule to an interface is shown below:

```
interfaces eth-0/0/3 {
      unit 0 {
            family inet {
                  filter {
                      input rule1;
                      output rule2;
                  }
             }
      }
}
```

In the example, all incoming IPv4 traffic to eth-0/0/3 will be filtered according to rule1. Outgoing traffic will be filtered according to rule2. If instead lo0 was specified instead of eth-0/0/3, rule1 would protect the RP from incoming traffic.

13.2 Match conditions

Packet filters are written as policy statement with a “from” part and a “then” part. Note that “from” in this context means “match out of all packets”, it does not indicate a direction.

The match conditions include:

- destination-address
- source-address
- address
- destination-port
- source-port
- protocol
- dscp
- icmp-code
- packet-length
- interface-group
- fragmentation-offset
- fragment-flags

- first-fragment
- is-fragment
- ip-options
- tcp-flags
- tcp-established
- tcp-initial

Most conditions have an argument and can be combined into complex rules. For more detailed documentation about each match condition, see the reference manual[1] or the on-line JunOS documentation.

13.3 Actions

The action part specifies actions to be performed on the identified packets. The actions include:

- accept: Accept the packet and send it to its destination
- discard: Silent discard
- reject: Drop and send an ICMP error message to the source.
- alert: Log an alert for the packet.
- count: Count the packets
- sample: Sample traffic
- log/syslog: packet header is logged.
- output-queue: Assign the packet to an output-queue
- loss-priority: Set packet loss priority (PLP)

It is important to remember that all filter rules have an *implicit deny rule*. That is, if a packet does not match any rule, it will be silently discarded.

13.4 Example

The following example shows a firewall rule that only accepts packets from two (private) prefixes, and discards and logs everything else:

```
firewall {
    family inet {
        filter rule1 {
            term allow {
                from {
                    source-address {
                        192.168.0.0/16;
                        10.0.0.0/8;
                    }
                }
                then accept;
            }
            term reject{
                then {
                    log;
                    discard;
                }
            }
        }
    }
}
```

13.5 Policing

If a policer is associated to an interface, it rate-limits the traffic to adhere to a *token bucket* specifying average bandwidth and maximum burst size.

When the threshold is exceeded, the traffic is either discarded, its loss-priority is set, or it is placed in a specific output queue.

The following example limits the traffic coming in on eth-0/0/3 to 500kbps:

```
interfaces {
    eth-0/0/3 {
        unit 0 {
            family inet {
                policer {
                    input p500k;
                }
            }
        }
    }
}
firewall {
    family inet {
        policer p500k {
            if-exceeding{
                 bandwidth-limit 500k;
            }
            then{
                 log;
                 discard;
            }
        }
    }
}
```

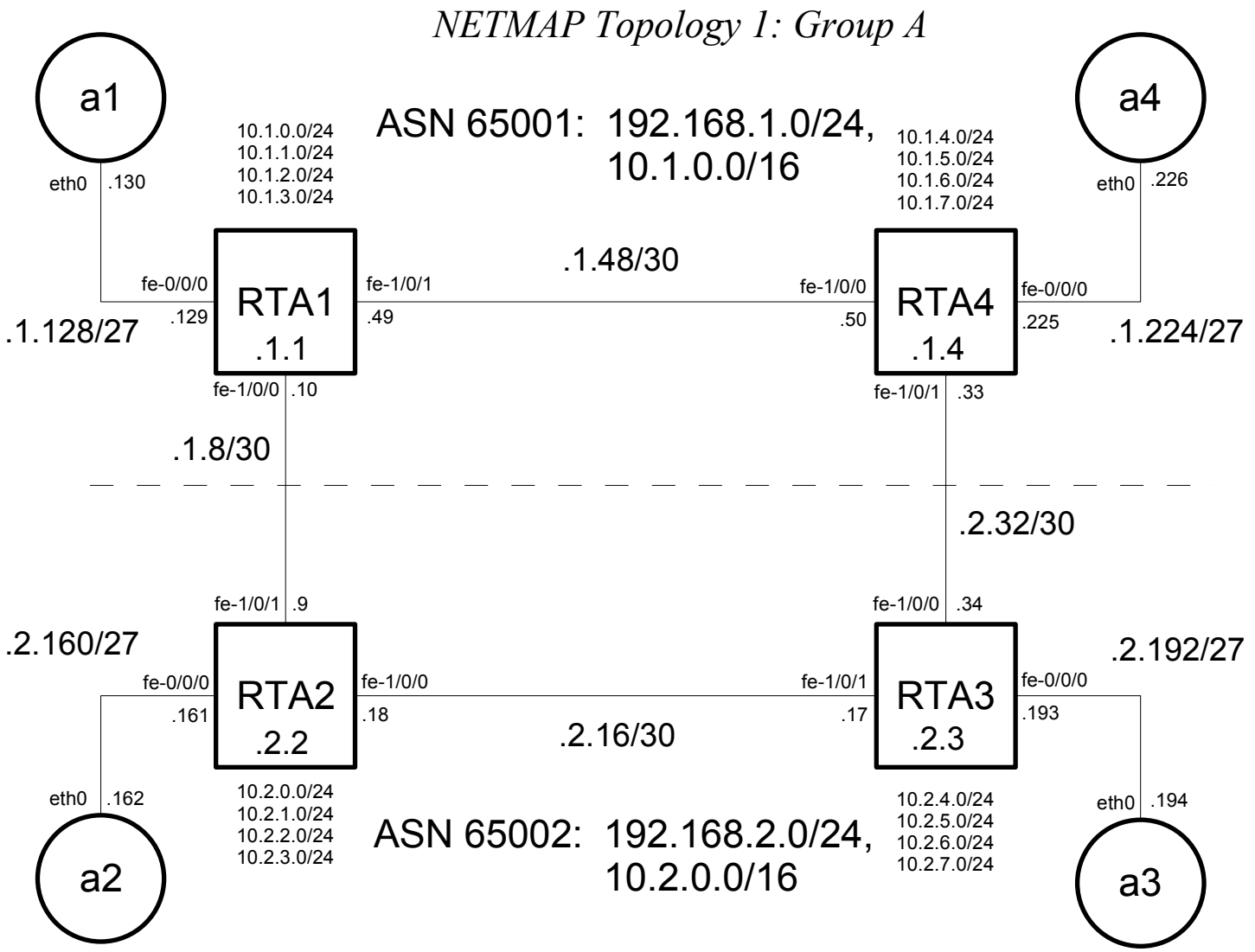
More useful is a combination of filtering rules and policing. As an extension to the filter actions, a policer may be invoked. In this way, packet filter rules identify a traffic class and the policer limits this traffic to a pre-defined threshold. This is, for example, useful to limit traffic to the RP.

In the following example, a simple filter has been extended with a refernce to the policer above that discards all icmp traffic that exceeds a 500kbps average load:

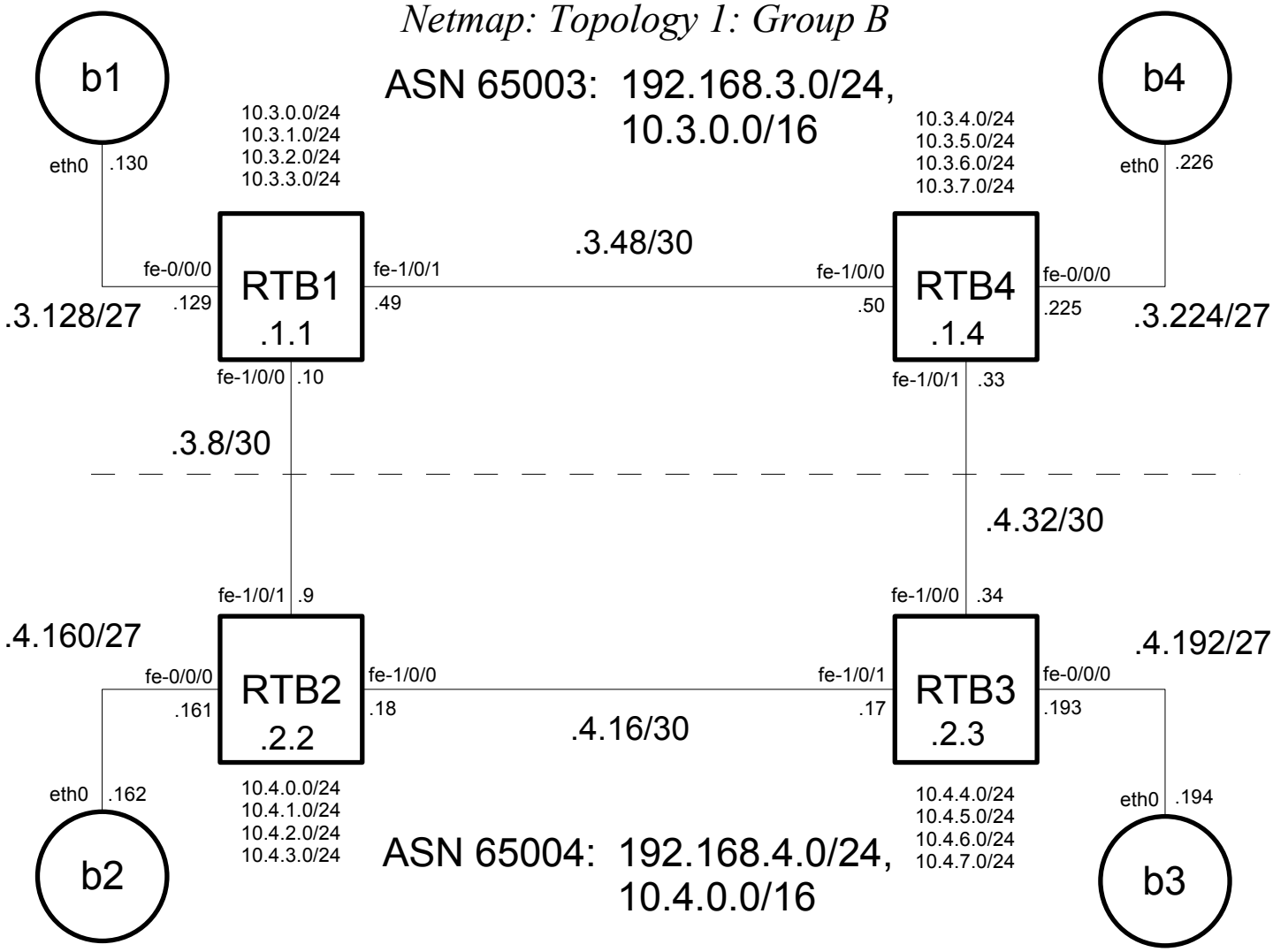
```
firewall {
    family inet {
        filter rate_icmp {
            term icmp {
                from protocol icmp;
                then {
                    accept
                    policer p500k;
                }
            }
        }
    }
}
```

Note that all other traffic will be discarded since the implicit deny rule will be activated for all traffic that does not match rate_icmp.

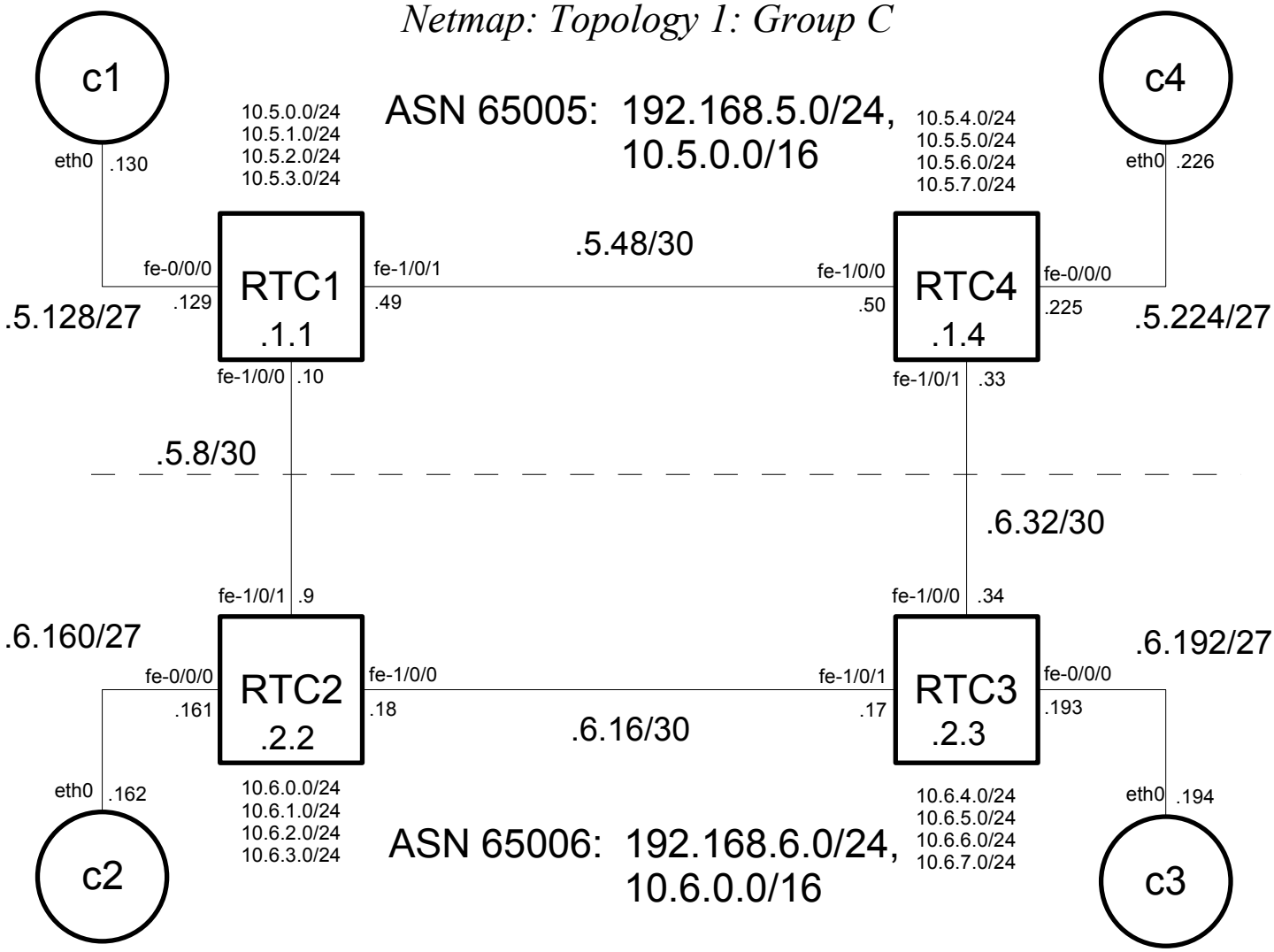
Appendix A: Netmap Topology

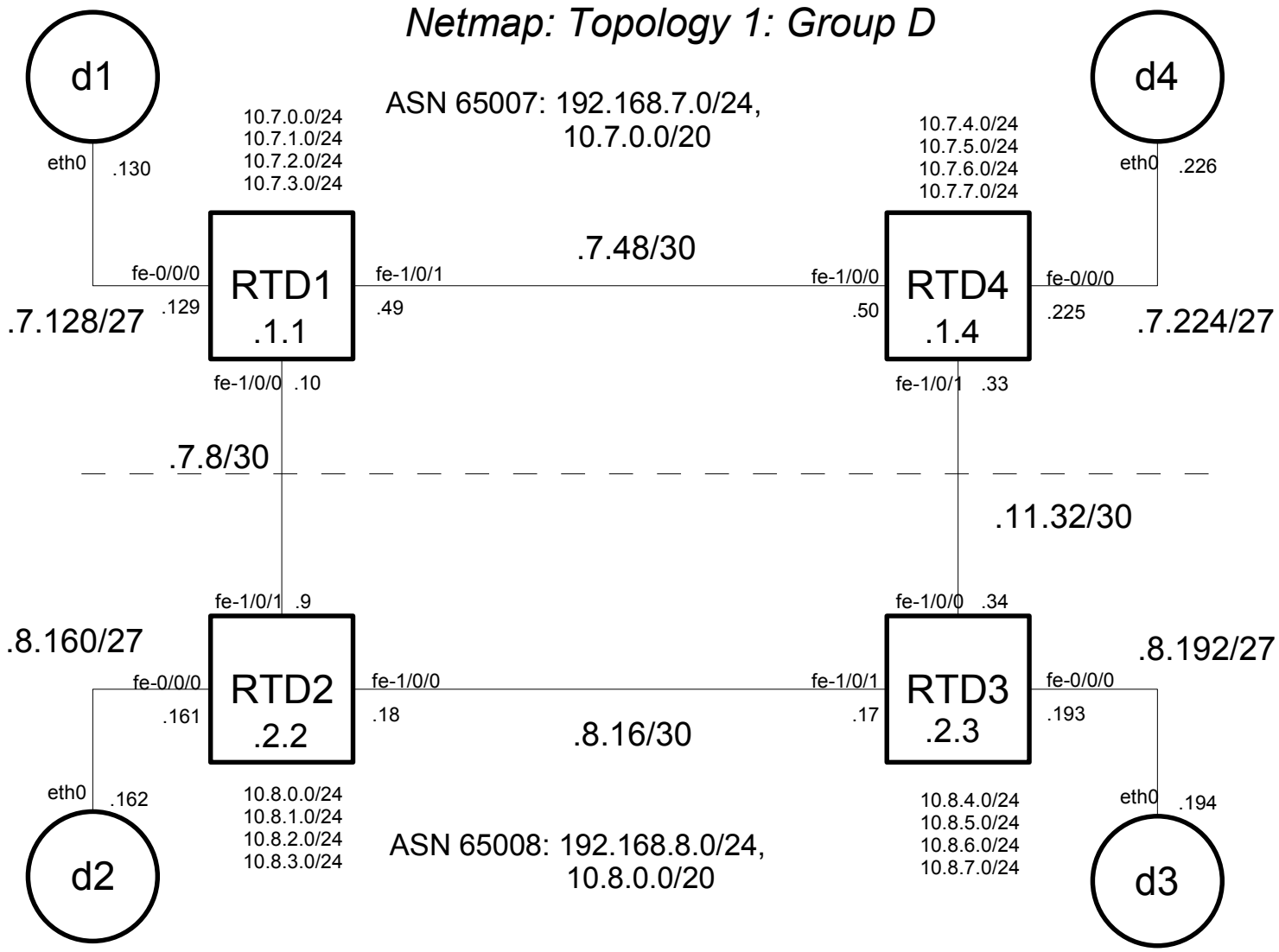


Netmap: Topology 1: Group B



Netmap: Topology 1: Group C




Netmap: Topology 1: Group D
d1
d4
ASN 65007: 192.168.7.0/24,
10.7.0.0/20
10.7.0.0/24
10.7.1.0/24
10.7.2.0/24
10.7.3.0/24
10.7.4.0/24
10.7.5.0/24
10.7.6.0/24
10.7.7.0/24
eth0
.130
eth0
.226
fe-0/0/0
.129
RTD1
.1.1
fe-1/0/1
.49
.7.48/30
fe-1/0/0
.50
RTD4
.1.4
fe-0/0/0
.225
.7.128/27
.7.224/27
fe-1/0/0
.10
fe-1/0/1
.33
.7.8/30
.11.32/30
fe-1/0/1
.9
fe-1/0/0
.34
.8.160/27
.8.192/27
fe-0/0/0
.161
RTD2
.2.2
fe-1/0/0
.18
.8.16/30
fe-1/0/1
.17
RTD3
.2.3
fe-0/0/0
.193
eth0
.162
10.8.0.0/24
10.8.1.0/24
10.8.2.0/24
10.8.3.0/24
ASN 65008: 192.168.8.0/24,
10.8.0.0/20
10.8.4.0/24
10.8.5.0/24
10.8.6.0/24
10.8.7.0/24
eth0
.194
d2
d3

Netmap: Topology 1: Group E