

Computer Networks – Practical session #3

Network interconnection, static and dynamic routing

M1 MOSIG

When booting the computers, choose “FreeBSD”, and then login with:

Username: “root”

Password: “root./”

Today we will focus on routing! First, *static routing* by manually entering routing information, and then *dynamic routing* using RIP to automate route computation.

Don't forget to send your report to your teacher at the end of the practical session! You can send one report for each group. Don't forget to mention the name of the other persons in the group.

1 Important remarks

This lab session can be quite complex, so here are some guidelines and a few tips to make it easier to manage.

Network interfaces In this session, you will have to create several IP networks, and some of your computers will be connected to two different networks. This will be achieved either through a USB network interface (ue0) or with a switch and tagged VLANs.

Addressing To make observation and debugging simpler, it is better not to take contiguous address ranges to prevent aggregation.

We ask you to use the following network addresses: 10.X.10.0/24, 10.X.20.0/24, 10.X.30.0/24 and 10.X.40.0/24, where X is your group number and 10 is the first network, 20 the second network, and so on.

Packet capture and debugging When you have two active network interfaces, run two instances of `wireshark` side-by-side, each listening on a single interface. This allows to more easily understand what's going on in each network.

General organisation The best way to control and observe what is going on in the entire network is the following:

- have only one command running per terminal, hence you have to open one terminal per command;

- layout the terminals so that you can see them all at the same time: resize and move them appropriately;
- have one `wireshark` running per configured interface;
- run all commands and daemons in the foreground (option `-d` for `routed`).

2 Static routing with tagged VLANs

You will start by configuring each forwarding table statically, using tagged VLANs to obtain several virtual interfaces on the router.

2.1 Target topology

In this scenario, we will use `pc2` as a router that will forward packets between two distinct Ethernet networks, as shown in the following figure:

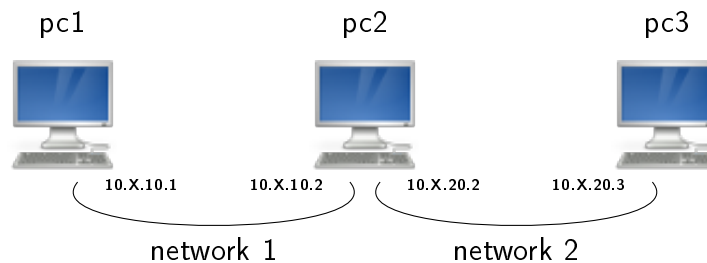


Figure 1: Target topology with `pc2` acting as a router.

2.2 Setup with tagged VLANs

Since `pc2` needs two interfaces, we will use tagged VLANs and a switch, as shown in Figure 2. If you don't remember how to configure VLANs, refer to the previous practical session.

The resulting topology with the name of the network interfaces is shown in Figure 3.

- ☞ Connect three computers to the switch as shown in Figure 2.
- ☞ Connect a fourth PC to the switch, and use it to access the management interface of the switch (using `telnet` towards `192.168.0.254`, or with a serial console).
- ☞ On the switch, configure the port connected to `pc3` so that it belongs to VLAN 2 (untagged).
- ☞ Configure IP addresses on `pc1` and `pc3`.

For `pc2`, you will need to create the two virtual interfaces `vlan1` and `vlan2`, and associate them with the appropriate VLAN tag. As a reminder, on FreeBSD, to create a virtual interface `vlanX` associated to a tagged VLAN X:

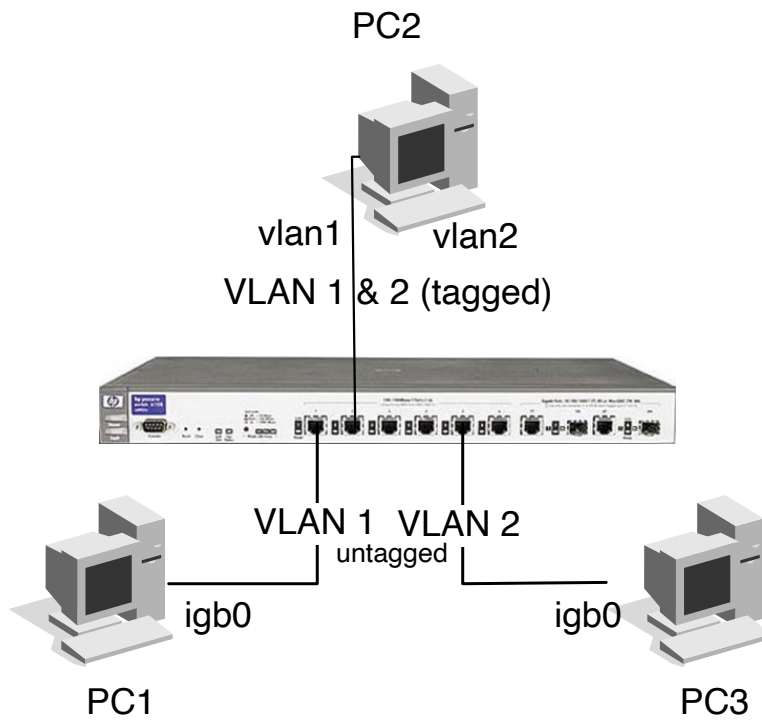


Figure 2: Tagged VLANs

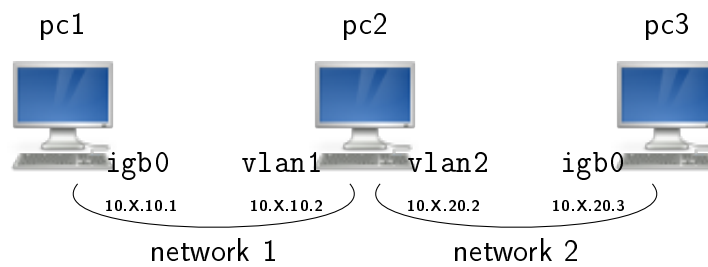


Figure 3: Target topology with interface names

```
ifconfig vlanX create
ifconfig vlanX vlan X vlandev bge0
ifconfig vlanX 10.A.B.C/24 up
```

- ☞ Create the virtual interfaces `vlan1` and `vlan2` on `pc2`, and assign them IP addresses (do not forget the netmask) so that they are in the same subnet as `pc1` and `pc3` respectively. Also disable hardware support for vlans, with `ifconfig bge0 -vlanhwtag`.
- ☞ On the switch, configure the port connected to `pc2` so that it allows VLANs 1 and 2 (as **tagged** VLANs).

Q 1 — *Start a packet capture on `igb0` on `pc2`, and try to ping between `pc1` and `pc2`, and between `pc3` and `pc2`. Verify that the pings work. What do you see in the packet capture of `PC2`? Check the headers of an incoming packet from `PC1` or `PC3`. Why do we talk about “VLAN tags”?*

At this point, you have successfully setup two separate networks and `pc2` is connected to both networks using tagged VLANs. Congratulations!

2.3 Static routing

It is now necessary to instruct `pc2` to relay packets (by default, FreeBSD doesn't). Relaying packets between networks is called *packet forwarding* or *IP forwarding*.

- ☞ Enable packet forwarding on `pc2` using `sysctl net.inet.ip.forwarding=1`.

Q 2 — *Try to ping between `pc1` and `pc3`. Does it work?*

Q 3 — *What route should be added on `pc1` to be able to reach `pc3`? Do you need to add a route in the routing table of `pc2`? Why?*

To add or delete a route, the `route` command should be used:

```
route add|delete <destination> <gateway>
```

Example usage:

```
# route add 192.168.20.0/24 192.168.30.4
```

To display the routing table:

```
# netstat -rn -f inet
```


- ☞ Configure the routing table of `pc1` to be able to reach `pc3`, and likewise on `pc3` to be able to reach `pc1`.


Q 4 — *Verify that `pc1` and `pc3` can now communicate through `pc2`.*

- ☞ Run `tracertoute` (using `tracertoute -n -P icmp`) from `pc1` to `pc3` and vice-versa.

Q 5 — *How many routers do you see in the path? Is it consistent?*

Q 6 — *What is the IP address of pc2 that appears in each traceroute? Is it the same? Interpret.*

 On pc1, pc2 and pc3, launch `arp -ad` to flush the ARP tables. On pc2, start a packet capture on both configured interfaces.

 Ping pc3 from pc1. Analyze the traffic captured on pc2, especially the Ethernet and IP addresses of the ICMP packets.

Q 7 — *Describe and explain the temporal sequence of exchanged packets (ARP and ICMP). Explain in details the work done by the router pc2.*

3 Dynamic routing with RIP

You will now experiment with one of the first Internet routing protocols: RIP (*Routing Information Protocol*). This protocol will allow to automatically configure the routing table of hosts.

Dynamic routing is handled by a routing daemon¹. This daemon communicates with its neighbours running in the network to exchange routing information, compute routes and configure the routing table on the system.

3.1 The routed daemon

There are various RIP implementations. We will use the `routed` daemon provided with FreeBSD. It supports two different modes:

- *supply* mode (option `-s`), to announce routing information to other computers in the network
- *quiet* mode (option `-q`), to receive routing information, without supplying any routing information to other computers.

To sum up, we generally execute `routed` with the option `-s` on routers, and with the option `-q` on hosts. Furthermore, we will use the following additional options:

`-Pripv2` to force using version 2 of RIP;

`-Pno_rdisc` to prevent router auto-discovery with ICMP;

`-i` to allow `rtquery` to obtain the routing table;

`-d` to keep the process in the foreground instead of operating as a daemon. This allows to easily interrupt the program with `Ctrl-c`, and see some debugging messages on the terminal.

The typical command-line will look like:

```
# routed -Pripv2 -Pno_rdisc -i -d [-s|-q]
```

¹Autonomous process running in the background and taking care of a service of the system.

3.2 Additional practical guidelines

When you change the physical network topology or if you want to make a new experiment, make sure that you have:

1. stopped the routing daemons, `ctrl-C` in the terminal is enough if you used option `-d`, otherwise `killall routed` may help;
2. flushed the forwarding tables, using `route flush` or manually removing all entries using `route delete <destination>`.

3.3 RIP analysis with two networks

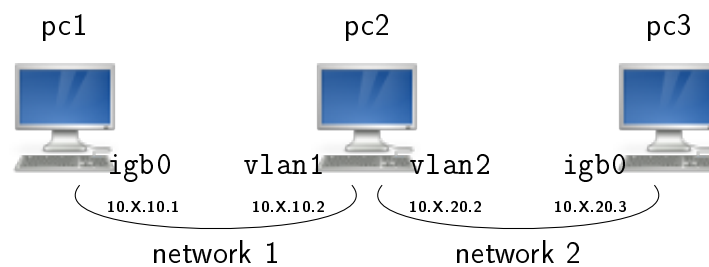


Figure 4: Initial network set-up

- ☞ Keep the previous setup with the switch and tagged VLANs on PC2. Logically, the topology is the one shown in Figure 4 (three computers, one acting as a router with two virtual interfaces: the switch is not represented in the figure). Then:
 - ☞ – disconnect all computers from the ENSIMAG network (`ifconfig em0 down`);
 - ☞ – flush the routing table on pc1 and pc3;
 - ☞ – run `routed` in *quiet* mode on pc1 and pc3;
 - ☞ – run `routed` in *supply* mode on pc2.

Q 8 — *Check the forwarding table of each computer. What can you conclude?*

Q 9 — *Check connectivity between pc1 and pc3.*

- ☞ While `routed` is still running on all computers, perform a packet capture on each interface of pc2. Don't forget to start two `wireshark` instances for the capture, one for each interface.

Q 10 — *What is the destination IP address of RIP packets?*

Q 11 — *Based on the packet capture, what kind of information is transmitted by RIP? Do router and hosts behave the same with respect to sending RIP information?*

Q 12 — *At which time interval does a RIP daemon transmit information to other computers in the network?*

☞ Query the routing table of each RIP daemon by using: `# rtquery -n`.

Q 13 — *What does the metric represent in the output of `rtquery`?*

☞ Check regularly the routing and forwarding tables on `pc1`, using `netstat -rnf inet` and `rtquery -n`. Then disconnect `pc1` from network 1 (unplug the cable) in order to simulate a network failure.

Q 14 — *What happens after around 3 minutes?*

☞ Connect `pc1` back on the network.

Q 15 — *What happens after around 30 seconds?*

Q 16 — *Based on your observations, explain how a `routed` daemon updates its routing table (shown by `rtquery -n`) and the forwarding table of the system (shown by `netstat -rnf inet`).*

3.4 Three networks

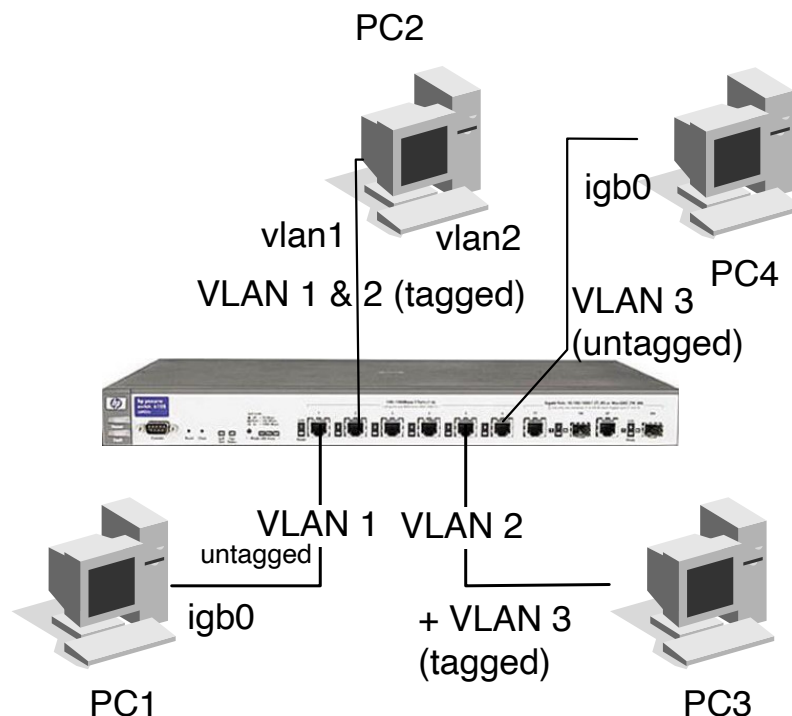


Figure 5: Tagged vlans for three networks configuration

- ☞ Connect pc4 to the switch and modify the router configuration as show in Figure 5
- ☞ Connect pc4 to the network in order to get the topology shown in Figure 6 where a third network is created. Don't forget to stop all the daemons and clean all the routing tables, as explained earlier.

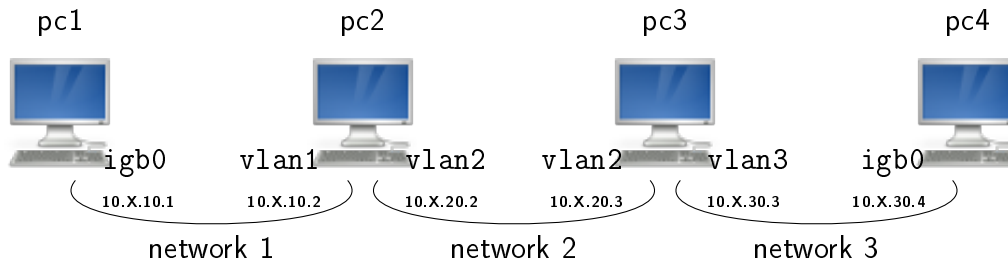


Figure 6: Three networks setup

- ☞ Configure IP addresses and use `routed` to ensure all computers can talk to each other. Don't forget to enable packet forwarding on pc3!

Q 17 — *What is the content of the RIP packets exchanged between pc2 and pc3? Detail what you find.*

3.5 Four networks

The purpose of this operation is to observe *dynamic reconfiguration* of routes. Use both `netstat` and `rtquery` to observe the forwarding table of the system and the routing table of the RIP daemon.

- ☞ Read again the end of Section 1 on the best way to control and observe what is going on in the entire network.
- ☞ Create a new VLAN that connects pc1 and pc4 in order to get the topology shown in Figure 7 where a fourth network is created. As before, stop all the daemons and clean all the forwarding tables.
- ☞ Run `routed` in supply mode on every station. Analyze the packets captured on both interfaces on pc1. Check the routing table of the RIP daemon.

Q 18 — *Which network path is used by pc1 to reach network 2? Why?*

Q 19 — *Similarly, which network path is used by pc2 to reach network 3?*

- ☞ Disconnect network 2 (by disabling the interfaces connected to this network, using `ifconfig interface down`). Look carefully at `rtquery` (notably on pc1) and at the RIP packets exchanged between routers.

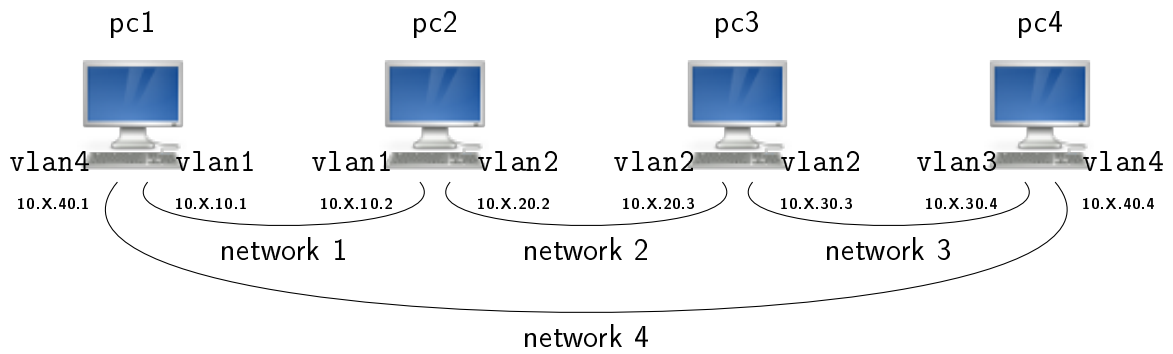


Figure 7: Four networks

Q 20 — *Is pc1 connected to network 2? By which process does it learn about the disconnection of network 2?*

Q 21 — *What is the path now used by pc2 to reach network 3?*

Q 22 — *Conclude on the advantage of using a dynamic routing protocol.*

Before leaving, make sure to put all cables and hardware back to their storage location.

To send your report at the end of the practical session, you may need to connect the computer back to the school network. This can be done using `ifconfig em0 delete` followed by `dhclient em0`.