Department of Information Engineering (DEI)
Master degree on ICT for Internet and Multimedia Engineering (MIME)

INTERNET – LAB Experience 2: Static Routing

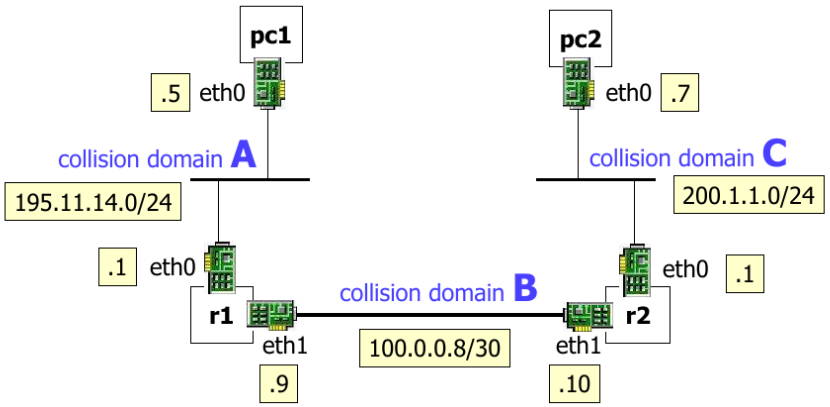

Figure 1: Network topology configuration details.

Exercise 1: *Start the four virtual machines as described in Fig. 1. Test the connectivity between* `pc1` *and the interface* `eth0` *of* `r1` *using the* `ping` *command. Then, test the connectivity between* `pc1` *and interface* `eth1` *of* `r1`*. Report the routing table of* `pc1`*, which can be listed with the* `route` *command.*

Directly connected networks are automatically inserted in the routing table when the corresponding interface is brought up. This is a common behavior of all real-world routers. Other networks, instead, are not automatically added in the routing table, so they need to be manually inserted. To do this, we can use two strategies: set the default gateway/router or configure a static route.

Default Gateway

The default gateway indicates the gateway (interface) through which you can reach all the other networks. On `pc1` execute `route add default gw 195.11.14.1` while on `pc2` execute `route add default gw 200.1.1.1` Using these settings, `pc1` will forward all packets directed to networks other than 192.11.14.0/24 through the interface 195.11.14.1. Analogously, `pc2` will send packets directed to unknown networks through interface 200.1.1.1.

Exercise 2: *What interfaces can be reached by* `pc1`*? And by* `r1`*?*

Static Routes

Interfaces on `r2` seem unreachable by `pc1`. Sniff on interface `eth1` of `r2` `tcpdump -i eth1` to check if echo request packets reach `r2`. You should see the request packets but not their answers. So, `pc1` knows how to reach `r2` but the opposite is not true. In fact, `r2` only knows where to forward packets directed to 100.0.0.8/30 and 200.1.1.0/24 networks, while network 195.11.14.0/24 is unknown to it. We then need to manually instruct `r2` on how to forward packets to `pc1` through the addition of a static route: `route add -net 195.11.14.0 netmask 255.255.255.0 gw 100.0.0.9 dev eth1`

Exercise 3: *Deploy a similar configuration on* ***r1*** *and report the command you used for that.*

Exercise 4: *On* ***pc1*** *we could have set one or more static routes instead of the default gateway. Report the commands to set these routes in order to make* ***pc1*** *able to communicate with the rest of the network without setting a default route. You can test these commands by first removing the default route on* ***pc1*** *(use* ***man*** *to learn how to do that).*

Exercise 5: *In the network of this lab routers* ***r1*** *and* ***r2*** *can be also configured to exclusively use default routes. Try such a configuration and report the routing table of* ***r1****.*

Advanced excercise

Create the network depicted in Fig 2. Since all machines are connected to a single hub `h1`, all interfaces are connected to a single collision domain. Remember that in Kathará you do not have to manually add hubs to the network.

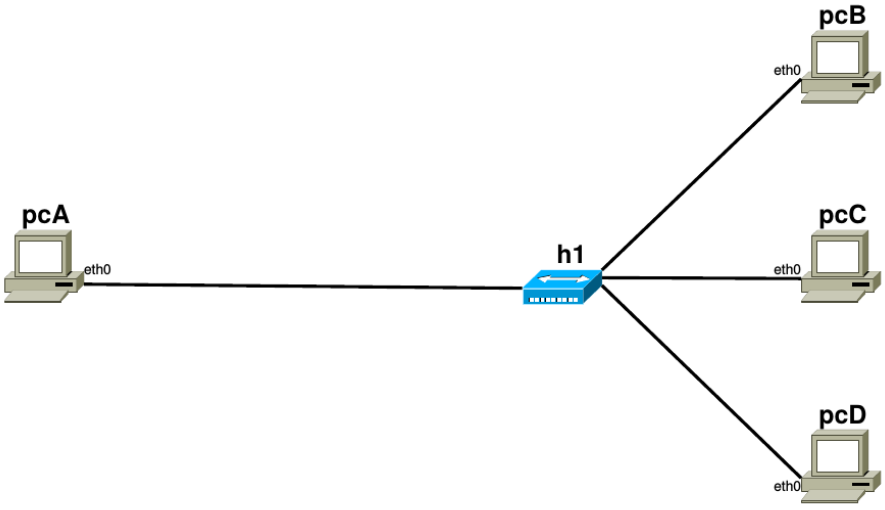

Figure 2: Network topology configuration details in the advanced exercise.

Set the IP addresses as described in Table 1. Note that the four hosts are all connected to the same collision domain but have different netmasks.

Machine	IP address	Network
`pcA`	192.168.20.1/24	192.168.20.0/24
`pcB`	192.168.20.193/28	192.168.20.192/28
`pcC`	192.168.20.206/28	192.168.20.192/28
`pcD`	192.168.20.254/24	192.168.20.0/24

Table 1: IP addresses in the advanced exercise.

For each host, check if it can communicate with any other machine in the network. Add the needed entries to the routing tables of `pcA`, `pcB`, `pcC`, and `pcD` so that they can reach any other machine in the network.

Exercise 6: *Write down the resulting routing tables for `pcB`.*