

Lab 1: IPv6

Reti di Calcolatori II – Università di Trieste – Martino Trevisan

In this lab, you will practice with IPv6, assigning configuration parameters to a set of virtual hosts. You will use a software called Mininet, which makes it playing with virtual networks rather easy.

First, visit the Mininet website (<http://mininet.org/>) and check how it works. Then, install it. If you use Ubuntu, just run the following command: `sudo apt-get install mininet`

Mininet works by creating virtual hosts using Linux containers. Containers are connected one with each other using a virtual links and virtual switches. It is possible to customize the network topology to create arbitrary complex networks.

Once the network is created, you can run any command on the virtual hosts, such as ping, curl, etc. Hosts are automatically configured with private IPv4 addresses, but you can modify their configuration with the normal Linux tools.

Start Mininet

Mininet can emulate various network topologies. Start a simple topology with four hosts connected with a single switch with:

```
sudo mn --topo single,4
```

Then, you are prompted to an interactive shell, where you can execute commands on any host. When you type a command on the shell, the first word must indicate the virtual host where you want to execute the command. The remaining part constitutes the bash command.

For example, to check the IP address on h1, you should run the `ip addr` Bash command on host h1, with:

```
h1 ip addr
```

To ping h2 from h3, type:

```
h3 ping h2
```

Assigning IPv6 addresses

Check, using the proper commands, if the host already have any IPv6 address assigned. Can you already ping a host using IPv6 from any other host? You can do this, by specifying an IPv6 address as argument of the `ping` command, such as:

```
h1 ping fd00::4
```

Now, assign to the four hosts an IPv6 address from the `fd00::/8` subnet. To add an IPv6 address to an interface, use the command:

```
ip -6 addr add <ip>/<class> dev <interface>
```

Then, ping a host using IPv6. Does it work?

Check the generated traffic and compare it with the traffic you get when running a ping on IPv4. Does it vary? Which protocols are used to obtain the MAC address of the destination host given its IP address?

Hint 1: when you start Mininet, various virtual interfaces are automatically created. Run Wireshark on those interfaces to intercept the traffic between virtual hosts.

Hint 2: To clean the ARP cache, run the command:

```
ip -s -s neigh flush all
```

The two nodes when pinging using IPv4, they use ARP requests to get the MAC address and besides this the two hosts transmit ICMP requests and replies to each other (PING).

When using IPv6:

- Ipv6 headers sent are larger than the IPv4 ones
- IPv6 uses the Neighbouring Discovery Protocol for obtaining the MAC address of a secondary node, and this NDP packet is sent on multicast and NOT broadcast.

Duplicate IP addresses

Assign the same IPv6 address to two hosts. What happens when you ping the duplicate IP address from another host? How many Neighbour Solicitations the host receive?

Try the same with IPv4 (use the `ip` command to have duplicate addresses). How many ARP replies the host receive?

Hint: check the traffic a host generates when you assign a new IPv6 address to it. Check deeply the `ip addr` command output.

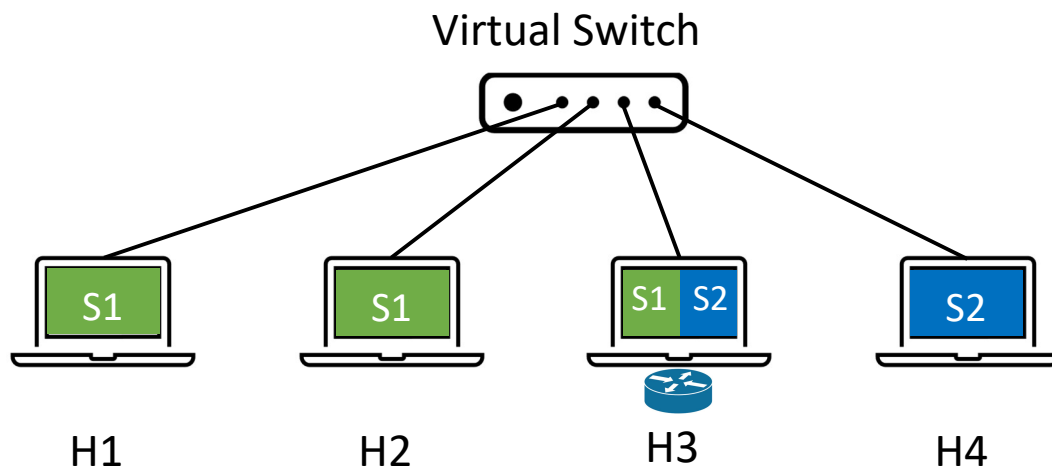
I assigned the same IPv6 address to h1 and h2. When executing the “ip addr” command I was able to see both IPv4 and IPv6 addresses for all hosts. After assigning to same address of h1 to h2, I noticed that the output of “h2 ip addr” was giving some error on the Ipv6 scope global address. The error is “dadfailed tentative” and it means that the Duplicate Address Detection sends a Neighbor Solicitation to see if the address is already in use and in this case, it is used by h1, so h2 do not assign that IPv6 address to the interface for regular use.

When sending a ping from h3 to the duplicated IPv6 address there will be Neighbor Solicitations and Advertisements just from the scope link IPv6 addresses of h1 and h3, since the duplicated address was assigned first to h1. Also “Router Solicitation” packets were present in the traffic.

When transmitting from h3 to duplicate IPv4 address (h1 and h2), h3 sends an ARP to resolve IP address and would get a double ARP reply, so one from h1 and one from h2.

Play with IPv4 Routing (optional)

Now divide the four hosts into two subnets S1 and S2. Two hosts belong to the first subnet S1, one host belongs to the second subnet S2. The last host belongs to two subnets, thus it has two IP addresses and acts as a router, like in the following picture.



First, decide the subnet addresses and assign them with the commands:

```
ip addr add <ip>/<class> dev <interface>
```

Notice that the above command adds an IP address to an interface. A single interface can have more than one IP address. If you need to remove an address from an interface, use the command:

```
ip addr del <ip>/<class> dev <interface>
```

You should also modify the routing table of the devices using the command:

```
h1 ip route add <subnet> via <next hop> dev <interface>
```

Think carefully how routing tables of different hosts should look like. Notice that the nodes that acts as a router shall have two IP addresses.

Moreover, to make a host to act as a router you should enable IP forwarding with the command:

```
sysctl -w net.ipv4.ip_forward=1
```

You should also prevent the router host from sending erroneous ICMP redirect with the command:

```
h3 echo 0 | tee /proc/sys/net/ipv4/conf/*/send_redirects
```

Once everything is configured, run a ping between two hosts on different subnets. You must report in the answer a schematic view of your network, including routing tables and IP address. Moreover, write all frames generated when you launch a ping, detailing source and destination MAC and IP addresses and the meaning of each packet.

Enlarge the box as needed

Installation notes

There are various possibilities to run Mininet

Native Ubuntu: The best way to use Mininet is to install it on an Ubuntu Machine. Just run : `sudo apt-get install mininet`

Using the VM: You can download the VM image available on the official website (<https://mininet.org/download/>) and run it with Virtual Box. However, the VM has IPv6 disabled by default, and you must enable it with the following steps:

- Edit the `/etc/sysctl.conf`

Delete the lines:

```
net.ipv6.conf.default.disable_ipv6 = 1
```

```
net.ipv6.conf.all.disable_ipv6 = 1
```

```
net.ipv6.conf.lo.disable_ipv6 = 1
```

- Edit `/etc/default/grub` and delete the `ipv6.disable=1` string in the `GRUB_CMDLINE_LINUX` and `GRUB_CMDLINE_LINUX_DEFAULT` lines.

- Run: `sudo update-grub`
- Reboot the machine

Using Windows subsystem for Linux: install mininet and bridge-utils with: `sudo apt-get install mininet bridge-utils` . Then, you must add the `--switch lxbr` option when running Mininet, like for example:

```
sudo mn --topo single,4 --switch lxbr
```