POLITECNICO di TORINO

Dipartimento di Elettronica e Telecomunicazioni

COURSE SYLLABUS
NETWORK MEASUREMENT LABORATORY

Marco Mellia

E-mail: mellia@polito.it

September, 2017

Table of content

Networking Lab 1 – Basic ping

The goal of this laboratory is to get used to the tools and command line mechanisms to properly configure a PC. Students will work in groups of 3 people. Each group will use three PCs, one Ethernet switch, three UTP cables and three pen drives with installed Linux. PCs must be bootstrapped from the laboratory pen drive.

1. Linux and useful commands

Linux is a UNIX based O.S. Most of the lab will require working with the Linux command line. Here is a brief list of useful commands. It is not intended to be a complete tutorial. In general, it is always useful to check the online manual that can be obtained by using the `man` command. There are plenty of tutorials you can find in the Internet to get more information.

`man <command>`: show the man page of the <command>. For example

`man man` shows the man page of the `man` command.

You can go up and down with the cursor arrows or page up/down keys. To exit, press **"q"** as quit. To look for some keyword, use the **"/"** – slash- then insert the string you are looking for.

The laboratory pen drive has been configured so that the user is logged in as a normal user, with account name "**laboratorio**". This is a non-privileged user, i.e., she has no right to change the system configuration. Thus, you have to run them as "**superuser**", or **"root"** to run most of the commands that require privileged access. This can be done by prepending the command "`sudo`" – `do` as `su`peruser- to the command you would like to run. The system will ask for the user password before granting access as super user. The password is "**studente**". For example

```
sudo ifconfig eth0 172.16.1.1 netmask 255.255.255.0
```

runs `ifconfig` as *superuser* instead of *laboratorio* user.

2. Networking related commands

- **`ethtool`** - Display or change Ethernet card settings. This command allows you to check and change the **physical layer configuration** of an Ethernet interface. It has a lot of options, which depends on the actual hardware and driver of the Ethernet interface.
- **`ifconfig`** - configure a network interface. Some useful options
  - `ifconfig` : show the configuration of the interfaces that are actually up and running
  - `ifconfig –a`: show the configuration of **all** interfaces in the system, including those that are down (not currently active)
  - `ifconfig <ethX>`: show the configuration of interface X, X=0,1,2,3,… depending on how many interfaces the system has
  - `ifconfig <ethX> <IP_ADDRESS> [netmask <MASK>] [broadcast <BROADCAST>] [MTU <MTU>]`: configure the interface `ethX` with `IP_ADDRESS`, netmask `MASK`, broadcast `BROADCAST` and MTU `MTU`.
  - [same as before] `ifconfig <ethX> <IP_ADDRESS/MASK> [broadcast <BROADCAST>] [MTU <MTU>]`: configure the interface `ethX` with `IP_ADDRESS`, netmask `MASK`, broadcast `BROADCAST` and MTU `MTU`.
  - `ifconfig <ethX> up|down` : put the interface up and running (active) or down (not active).

- **route**: show / manipulate the IP routing table. `Route` manipulates the kernel's IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an interface after it has been configured with the `ifconfig` program. When the `add` or `del` options are used, route modifies the routing tables. Without these options, route displays the current contents of the routing tables.
  - `route add default gw <GW_ADDR>`: add the default gateway at `GW_ADDR`
  - `route add –net <NET_ADDR> Netmask <MASK> gw <GW_ADDR>`: add the `NET_ADDR/MASK` route via `GW_ADDR`.
- **arp**: manipulates or displays the kernel's IPv4 network neighbor cache. It can add entries to the table, delete one or display the current content.
  - `arp –n`: show the arp table state without resolving addresses
  - `arp –d IP_ADDR`: delete the entry in the arp table referring to `IP_ADDR`
- **netstat:** Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships
  - `netstat –t –n`: show the TCP (-t) connections status using numerical (-n) addresses instead of trying to determine symbolic host, port or user names
  - `netstat –u`: show the UDP (-u) connections status
  - `netstat –l`: show only listening sockets. (These are omitted by default.)
- **ping:** send ICMP ECHO_REQUEST to network hosts `IP_ADDRESS`. It uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams (``pings'') have an IP and ICMP header, followed arbitrary number of ``pad'' bytes used to fill out the packet.
  - `ping –s <size> -c <count> -i <interval> IP_ADDR`: send `count` ICMP `ECHO_REQUEST`s separated by `interval` seconds, each of size `size` to `IP_ADDR`
- **traceroute** `<IP_ADDR>`: print the route packets trace to network host `IP_ADDR`. It tracks the route packets taken from an IP network on their way to a given host. It utilizes the IP protocol's time to live (`TTL`) field and attempts to elicit an `ICMP TIME_EXCEEDED` response from each gateway along the path to the host.

Which of the previous commands must be run as superuser?

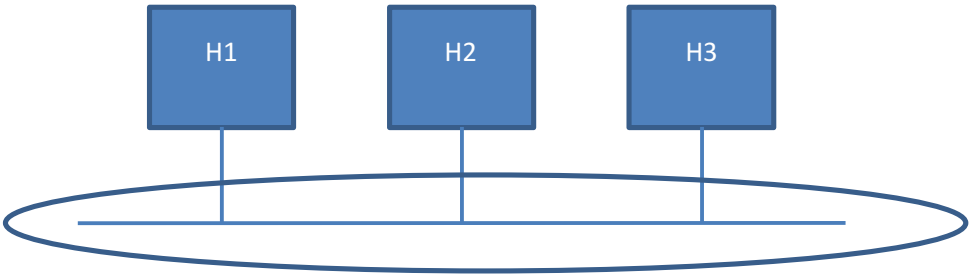
3. Configuration of the testbed

Each group must configure its testbed, i.e., the three PCs must be properly configured and interconnected to create and setup a LAN.

1. **Disconnect** the PCs from the laboratory LAN **before booting** them
2. Insert the pendrive and boot the PC
3. Connect the PCs to the switch using the cables
4. Check that the physical links are working by looking at the switch LEDs.
5. Assign IP addresses to each host in the LAN. **Use private addresses from the CLASS B range**, and choose the strictest Netmask that is required to host no more than **62 hosts**.

WARNING: the Ubuntu Linux Network configuration manager will periodically try con (re)-configure your LAN. Disable it from the top left menu bar (remove the tick from "Enable Networking").

In the following, we will refer to each host as `H1`, `H2`, `H3`.



SAME LAN and SAME SUBNET

1. Report the configuration of the testbed

In the laboratory workbook, report the physical layer configuration of the testbed. Sketch both the physical, datalink and network layer topologies.

1) Use `ethtool` to check the status of the physical layer and datalink layer configuration of the linecard. How much of the provided information can you understand? Report the configuration of a linecard and summarize its state. What changes if you unplug the cable from the linecard?
2) Use `ifconfig` to check the status of the networking layer. How much of the provided information can you understand? Report the configuration of a linecard and summarize its state. What is the difference between "`UP`" and "`RUNNING`" for an interface?
3) Use the command `route` to check the status of the routing table. What happens when you configure a linecard using the `ifconfig` command?
4) Put the linecard "`down`". What happens to the routing table? Is the IP configuration of the linecard been discarded (suggestion: use `ifconfig -a` to see all interfaces…)?
5) Put the linecard "`up`". What happens to the routing table? What happens to the IP configuration of the linecard?
6) How can you remove an IP address that has been assigned to a linecard?

2. Connectivity check – basic ping

Using the ping command, check if it is possible to reach other hosts in your LAN. Run the command

```
H1: ping H2 -c 4
PING H2 (H2) X(Y) bytes of data.
Z bytes from H2: icmp_seq=1 ttl=64 time=18.6 ms
Z bytes from H2: icmp_seq=2 ttl=64 time=0.127 ms
Z bytes from H2: icmp_seq=3 ttl=64 time=0.127 ms
Z bytes from H2: icmp_seq=4 ttl=64 time=0.125 ms

--- H2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.125/4.758/18.656/8.024 ms
```

1) What are the numbers **X,Y,Z** ?
2) How can the ping program report the `icmp_seq`? Why there is such a header? What could happen if such header were not present?
3) What is the **ttl** value? Does it refer to the packets sent, or received? Who decides to use 64?
4) What does it change if you change the size of the requests using the `-s<size>` option? Which is the minimum size that allows ping to measure and report the "`time`" field? Why this is happening? If you were the programmer working on the implementation of a `ping` command, how would you implement the "`time`" measure?
5) What happens if `H1` tries to ping a host that is not active but belonging to your subnet, e.g., `H1: ping H4`
   Is there any packet that is sent on the LAN (check the LEDs on the switch)? Which packets are those?
6) What happens if `H1` tries to ping a host that is not active and does NOT belong to your subnet, e.g., `Y1`? Is there any packet that is sent on the LAN (check the LEDs on the switch)? Which packets are those?