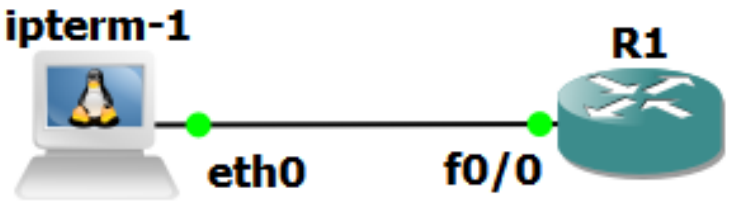
ELEC-H417: Lab 0 - Basics

Basics of GNS3, Wireshark and Cisco commands

Objectives

- Verify the installation
- Implement a first GNS3 topology
- Configure a CISCO router interface
- Use some basics UNIX command-line tools (ping, traceroute)
- Capture traces with wireshark

Topology for this lab



Device name	Type in GNS3	IP address	Netmask
ipterm-1	ipterm	192.168.1.2	255.255.255.0
R1	CISCO 3745	192.168.1.1	255.255.255.0

First of all, even if we work on the GNS3 software, you need to have the VM running in background.

Deploy the host and the router from the GNS3 and connect the links between them. Then
start all nodes (with the *play* button). Double-click on the router or the host to enter the
configuration console (terminal).

To add an host, select the ***End devices tab*** of the left-column (the third one with a picture of a screen). Then drag and drop the *ipterm* object in the center of the window. It will simulate a Linux computer in your network. **Do not use the *VPCS*, otherwise some linux commands won't work.**

To add a router, select the ***Router tab*** of the left-column (the first one with a picture of circle with four arrows). Then drag and drop the *CISCO 3745* object in the center of the window.

To link them together, click on the **last tab** of the left-column (with a picture of a cable).
Afterwards, click on the *ipterm-1* and select *eth0*, then click on the *R1* router and select *FastEthernet0/0*.

Configuration

Here are some information and examples on how to configure the Linux host and the CISCO Basics of GNS3, Wireshark, and CISCO router. You will have to adapt the examples to your scenario.

How to configure a Linux Host

- Display the host IP address with `ip address` or `ifconfig`

```
root@ipterm-1:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::800b:25ff:fe86:7c5  prefixlen 64  scopeid 0x20<link>
        ether 82:0b:25:86:07:c5  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7  bytes 586 (586.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- Set the IP address of `eth0`:

```
root@ipterm-1:~# ifconfig eth0 192.168.1.2 netmask 255.255.255.0 up
```

- Set your (default) gateway:

```
root@ipterm-1:~# route add default gateway 192.168.1.1
```

- Display your routing:

```
root@ipterm-1:~# ip route show
default via 192.168.1.1 dev eth0
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.2
```

or a more complete output:

```
root@ipterm-1:~# netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG        0 0          0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 eth0
```

Note that destination `0.0.0.0` is a notation for default gateway (in case no other option is found in the routing table).

How to configure a CISCO router interface

- Enter configuration mode:

```
R1# configure terminal
```

- Set the IP address and the netmask for interface *FastEthernet0/0* and activate the interface:

```
R1(config)# interface fastEthernet 0/0
R1(config-if)# ip address 192.168.1.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# end
R1#
```

- **Do not forget to save your configuration on a regular basis with the "write" commmand !**

```
R1# write
```

- Verification

```
R1# show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            192.168.1.1     YES manual up                    up
FastEthernet0/1            unassigned      YES unset  administratively down down
...
```

Tips: All CISCO commands can be abbreviated (if there is no ambiguity). For instance:
* **configure terminal** can simply be written as **conf t**
*** show ip interface brief** can simply be written as **sh ip int br**
Don't worry, you will learn to do this with time…

Let's move on the interesting stuffs

Validate

- Use the `ping` commands from the host (i.e. *ipterm-1*):

```
root@ipterm-1:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=255 time=35.2 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=255 time=10.9 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=255 time=7.19 ms
...
```

Tips: Stop the pinging process with **CTRL-C**

- Use the `ping` commands from the router (i.e. *R1*):

```
R1# ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/60/72 ms
```

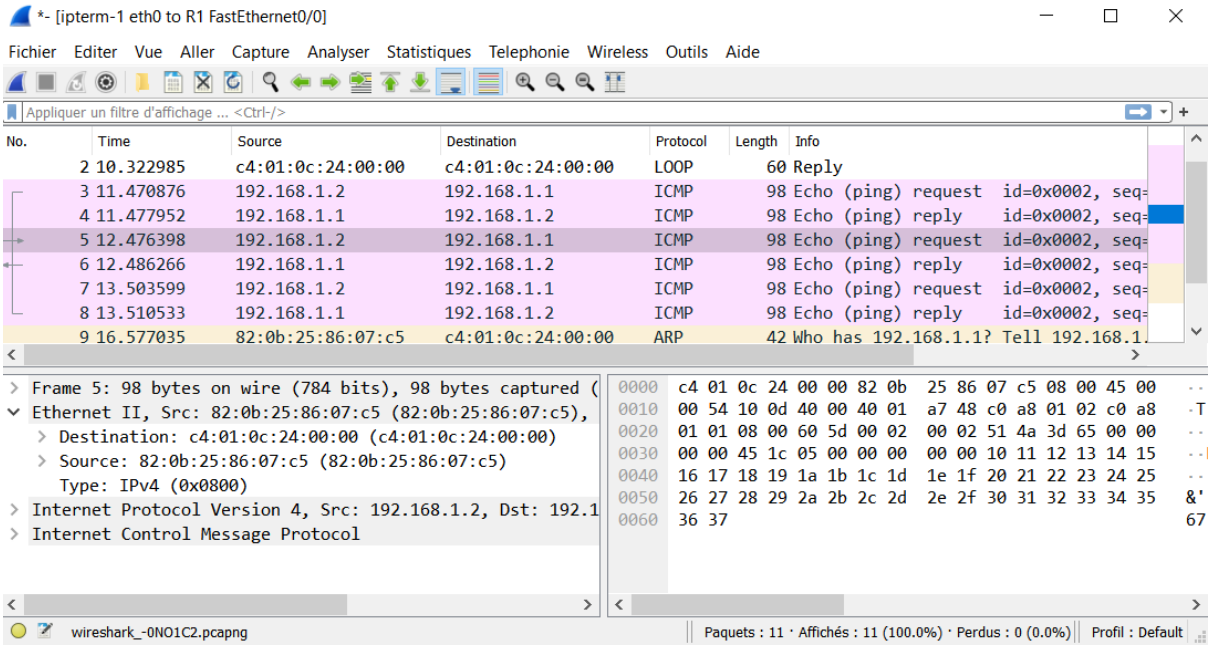
NB: In CISCO a `!` indicates a hit (i.e. successful ping) and `.` means a miss (i.e. no ping reply)

Packet sniffing

Right-click on the link between ipterm-1 and R1. Select "**start capture**". Wireshark will appear and you will be able to observe the packets passing by. Go to the host and ping 3 times the router (`ping -c 3 192.168.1.1`). Click on the stop button after the 3 pings.

Analyse the traffic, select a packet and take a look in the toggle lists. Observe that the headers are presented in the OSI order: Layer2, Layer3, Layer4 (i.e. Ethernet, IPv4, TCP/UDP, ...)

NB: note that wireshark decodes the packed for you (each field is presented in a human readable form)



Conclusion

Congratulations ! You now have all required expertise to start implenting and analyzing more advanced scenarios, inlcuding routing protocols, switching, VLAN, etc....

Do not hesitate to take a look at the ***Quickstart CISCO commands*** PDF file.