# Universidad de León

## Bachelor Degree on Computer Engineering

## *Course on Computer Networks*

# Practical on Linux IP internetworking



## Ancillary documentation for this practical

- *1. Skim-read the following presentation about the internetwork used in Lab B6 on 2023. Commands and configuration guidelines can be found useful in solving the exercises included in this practical:*

  http://paloalto.unileon.es/cn/labs/CN-IP-LinuxBasic.pdf

- *2. IP space partitioning and internetwork diagram for the present academic year:*

  http://paloalto.unileon.es/cn/labs/Internetwork24.pdf

## Introductory notes

The network diagram for this year (See above link) consists of a ring of four Linux routers. Each is equipped with several network interfaces, physical in some cases and virtual in others. Physical and virtual, for the purposes of this practical is irrelevant altogether. Implementation of virtual interfaces, where it is necessary, is achieved by way of IEEE 802.1Q VLAN which is implemented by the Cisco switches in Lab B6.

The Linux hosts selected for routers R1, R2, R3 and R4 are all connected to Lab B6 base LAN (VLAN 1, having IP prefix 192.168.1.0/24) serving as sort of a *scaffold LAN* such that remote connectivity to each of R* is available at all times regardless of the network configurations applied to them, that is, taking for granted that, in no case, interface eno1 is reset from its dhcp configuration in /etc/network/interfaces. Please, don't set a value other than dhcp to interface eno1 in the routers. The scaffold net IP addresses to each router follow:

R1: 192.168.1.191
R2: 192.168.1.159
R3: 192.168.1.226
R4: 192.168.1.210

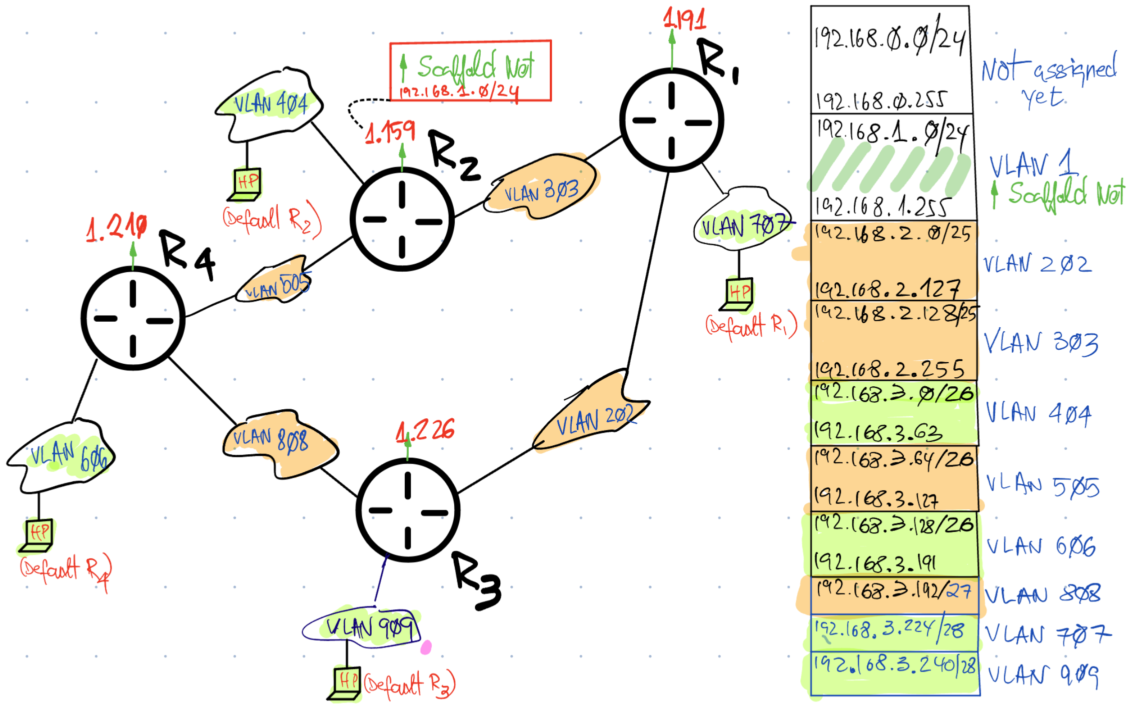


Figure 1. Internetwork, IP space breakdown and VLAN allocation

Each router provides service to a single stub network, in order to keep network complexity at bay. Service is provided via transit networks. Consider R1 as an example. Router R1 provides service to the customer hosts connected to VLAN 707, a stub network. It accesses the internetwork via transit networks 303 and 202. Same conceptual connectivity can be found at R2, R3 and R4, alongside their respective stub networks.

Connectivity tests are to be carried out by test hosts at stub networks. For example, a host is used for the tests at VLAN 707. Its IP address should be 192.168.3.226, at least, for 192.168.3.225 must be allocated to the router's interface at that network, following good practices in IP numbering (See fig. 1).

The internetwork has been already configured and tested. No reconfiguration is expected from students in this practical. The routers set the IP fwd kernel parameter at boot time, as well as the necessary routes to far networks are created (Understand *far networks* in the sense of networks that have no direct connection to the considered router, as is the case of VLAN 909 *vs.* router R1: That VLAN is not directly connected to router R1). The exercises below are aimed towards improving students grasping about IP and the IP LPM algorithm. The analysis tools we need to use here are tcpdump and ipcalc, only.

## Preconditions for doing the exercises

a. Power up all of the routers

b. Power up all of the PCs labelled as test hosts. There should be at least one at each stub network. The respective default router configured at these hosts should be the router each is directly connected to. Each student should sit in front of a test host.

c. Pings to the default router at your test PC should succeed at each test host at this time.

d. Open an ssh connection to your default router. For example, if you are seated at the test host in VLAN 707 (Default router is R1), do this:

```
$ ssh administrator@192.168.1.225

$ su

# /usr/sbin/ifconfig

Listing of interfaces
(Use the interface which IP is a member of VLAN 707; see Fig. 1)
```

e. In the preceding case, if you had to run tcpdump, you would compose the following command (Take note for later):

```
# tcpdump -i enp1s0.707 -ent -vvv icmp
```

f. Keep your local and remote sessions open during the resolution of the exercises that come next.

## Exercises for practical

1. **Draw a network diagram that represents router R1 interfaces (R2,** R3 or R4, can also be selected for this exercise; follow the instructions provided by the instructor for selecting a router and the corresponding test host).

    a. Execute /usr/sbin/ifconfig and obtain the listing of interfaces at use in R1, for example:

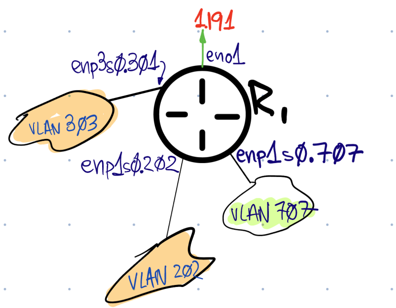


Figure 2. Network interfaces of R1

2. Open three separate ssh sessions to R1; in each, switch to superuser (Execute su) and launch the following instances of tcpdump (One per interface except for eno1):

    a. `# tcpdump -i enp1s0.707 -ent -vvv icmp`

    b. `# tcpdump -i enp1s0.202 -ent -vvv icmp`

    c. `# tcpdump -i enp3s0.303 -ent -vvv icmp`

3. **Router R1 protocol stack <u>accepts</u> an IP packet.** In this exercise, you use ping at the test host in VLAN 707 for sending an IP packet to R1's protocol stack, itself, no IP forwarding should be involved since the destination IP is the same as the receiving protocol stack's (192.168.3.225).

    a. You should see the icmp packet arriving at enp1s.707, in this case. No other interface should trace this packet: the ping packet should remain within the receiving stack (R1's). The icmp reply should be provided by R1 and you should receive it at the test host

    b. Check that the icmp request's IP packet sent by your ping, is forwarded on neither enp3s0.301 nor on enp1s0.202. This is of the essence at this time, since we aim to illustrate and test the concept of an IP packet being *accepted* in contrast to being *forwarded* by the receiving IP stack:

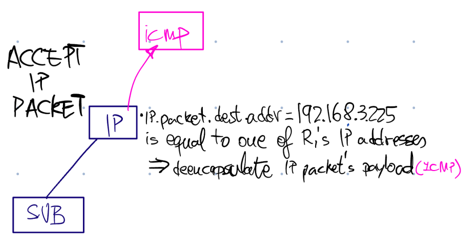


Figure 3. Received IP packet is accepted by R1's stack

c. Check the following tcpdump trace obtained in R1. Make sure that you understand it:

```
# [At test host in 7070] ping -c 1 192.168.3.225

# [At R1] tcpdump -i enp1s0.707 -entx -vvv icmp
tcpdump: listening on enp1s0.707, link-type EN10MB (Ethernet), snapshot length 262144 bytes
50:3e:aa:0a:71:2b > 50:3e:aa:0a:52:aa, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl
64, id 63764, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.3.227 > 192.168.3.225: ICMP echo request, id 1297, seq 1, length 64

50:3e:aa:0a:52:aa > 50:3e:aa:0a:71:2b, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl
64, id 13951, offset 0, flags [none], proto ICMP (1), length 84)
    192.168.3.225 > 192.168.3.227: ICMP echo reply, id 1297, seq 1, length 64
```

Observe that the ICMP reply has been generated by the same stack that received the request: The request's IP packet was accepted, and eventually handed to the ICMP protocol at R1, which thereafter generated the ICMP reply that you can see next to the request, on the preceding tcpdump trace. Observe, further, that the destination IP contained in the sent IP packet (**192.168.3.225**) is the same as R1's enp1s0.707's IP address printed out with ifconfig:

```
root@debian:/home/administrator# ifconfig enp1s0.707

enp1s0.707: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
inet 192.168.3.225  netmask 255.255.255.240  broadcast 192.168.3.239
inet6 fe80::523e:aaff:fe0a:52aa  prefixlen 64  scopeid 0x20<link>
ether 50:3e:aa:0a:52:aa  txqueuelen 1000  (Ethernet)
```

In summary, the packet is **accepted** by the stack, *i.e.,* it is not forwarded by IP's LPM.

4. **Router R1's protocol stack forwards an IP packet.**
In this exercise, we'll send ping from a host in VLAN 707 to a host in VLAN 404. Here, I assume that the IP address of the sending host (Check the IP partitioning, above) is 192.168.3.225. The destination IP address in VLAN 404 is assumed to be 192.168.3.2 (Make sure that the receiving host is up).

    a. Compute LPM upon the destination IP address like R1 should do. I'm using ipcalc to do the calculations:

        1. Obtain R1's FT (Forwarding table):

```
root@debian:/home/administrator# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 eno1
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eno1
192.168.2.0     0.0.0.0         255.255.255.128 U     0      0        0 enp1s0.202
192.168.2.128   0.0.0.0         255.255.255.128 U     0      0        0 enp3s0.303
192.168.3.0     192.168.2.130   255.255.255.192 UG    0      0        0 enp3s0.303
192.168.3.128   192.168.2.2     255.255.255.192 UG    0      0        0 enp1s0.202
192.168.3.224   0.0.0.0         255.255.255.240 U     0      0        0 enp1s0.707
192.168.3.240   192.168.2.2     255.255.255.240 UG    0      0        0 enp1s0.202
```

2. Check 192.168.3.2 against each entry in turn; I'm checking here for the Longest Prefix Matching that *I know* that results in a match beforehand (The FT entry is highlighted and underlined):

Step 1. Check the destination IP: **192.168.3.2 & 255.255.255.192 = *192.168.3.0***

Step 2. Check the FT entry: **192.168.3.0 & 255.255.255.192 = *192.168.3.0***

Since the results from Step 1 and from Step 2 are equal, we have a match -in this case, this is the only and Longest Prefix Match, therefore, R1 will forward the packet onto the interface indicated in the matching entry (enp3s0.303).

Check another entry from the FT (That is, all of them except the last resort router, the default router, which is only used when no match has been found by LPM). In this case, that will not result in a match. Check it by hand, maybe with the help of ipcalc.

3. Now, use the ip command's get route option to compute the longest prefix match in a single operation, as it is implemented by IP in the kernel:

```
root@debian:/home/administrator# ip route get 192.168.3.2
192.168.3.2 via 192.168.2.130 dev enp3s0.303 src 192.168.2.129 uid 0
```

*The ip command line utility is telling us that IP packets which destination IP address is 192.168.3.2 would be forwarded by this router onto interface enp3s0.303 on their trip to next hop 192.168.3.2.*

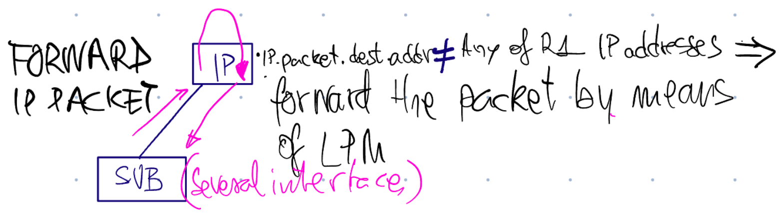


**Figure 4.** IP packet is forwarded

5. Start appropriate tcpdump traces at R1 that prove that your ping (Sent from test host in VLAN 707 to test host in VLAN 404) will ingress router R1 at **enp1s0.707** and leave through interface **enp3s0.303**.

6. Check that the ICMP reply ingresses R1 via enp3s0.303 and that leaves it via enp1s0.707.