Practicals on Computer Networks and Distributed Systems

Analysis of LAN traffic with the tcpdump sniffer

This practical explores how to interpret the fields that comprise two different types of datalink frames. The tcpdump software tool permits scanning a host's stack for sent and/or received datalink frames alongside their upper protocols' payloads . The first type of frame analyzed encapsulates an IP packet, and the second, encapsulates an WOL message (Wake-on-LAN) which allows a host to be powered up remotely.

Exercise 1. Hand computation of Lab B6 IP network number and the IP broadcast address

Every IP network must be allocated a block of consecutive IP addresses known as IP block. Henceforth, each host belonging to that network must be allocated a unique IP address from that IP block. The IP network in Lab B6 is allocated the following IP Block: 192.168.1.0/24 (See fig. 1). We'll delve into the topic of IP addressing in chapter 3; for the time being though we need to be able to handle IP addresses, as it were, at a basic level.

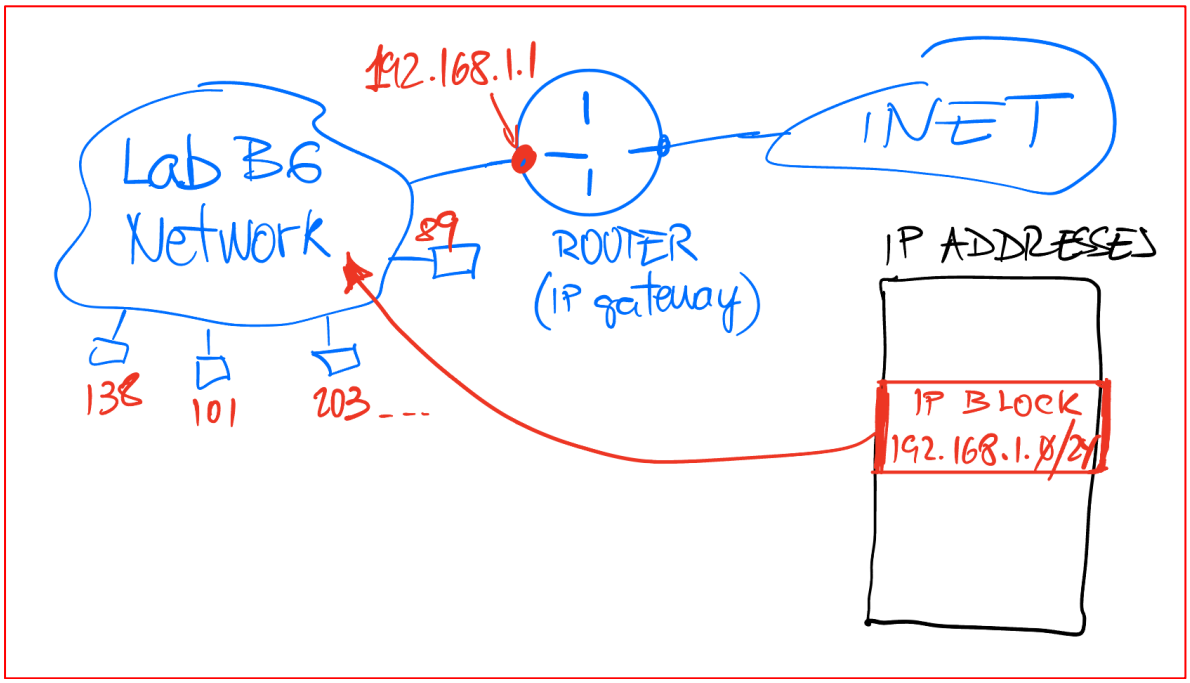

Figure 1. Allocation of 192.168.1/24 to Lab B6 network

a. Follow the instructions provided by the instructor on the board to compute Lab B6 IP Network Number:

b. Compute the Broadcast IP Address, that IP address which references all of hosts in Lab B6 network and which is common to all the hosts in the network:

c. Check that the -*internal*- address (192.168.1.1) given to the router does belong to the given IP Block:

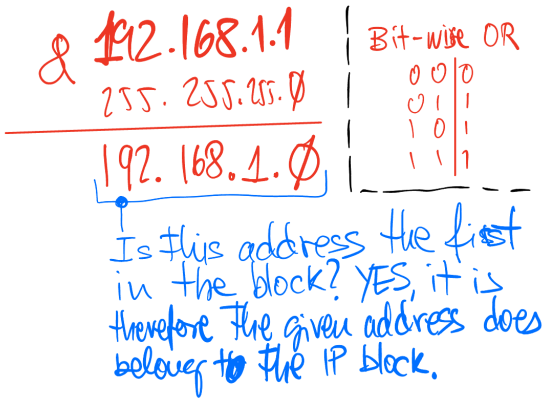

Figure 2. Checking that 192.168.1.1 belongs to 192.168.1/24

Exercise 2. Setting a correct (static) IP configuration in your Lab B6 host

- You can use either a Lab B6 PC or your own PC, assuming that the latter is connected to the Lab B6 network. Both types of physical connections are acceptable, Ethernet or WiFi
- The instructor will assign each student a unique number in the range 35..75. That number is the last decimal component of the IP address that you're going to configure in your NIC, today.
- Follow the following guidelines to set a correct IP configuration for your PC

a. Write down the unique number in range 35..75 that the instructor assigned to you, below:

Your Lab B6 IP address: 192.168.1._____

b. List and include the network device labels available in your Linux (Check the contents of directory /sys/class/net)

c. List all of the network devices physically connected to your host by issuing this command:

/usr/sbin/ifconfig -a

d. Today, we're using the Linux ip command to find out the network configuration. Execute the ip address command as illustrated in fig. 3. Then, identify which NIC label you're going to use for this exercise (Typically eno1, enp1s0 in our PCs):

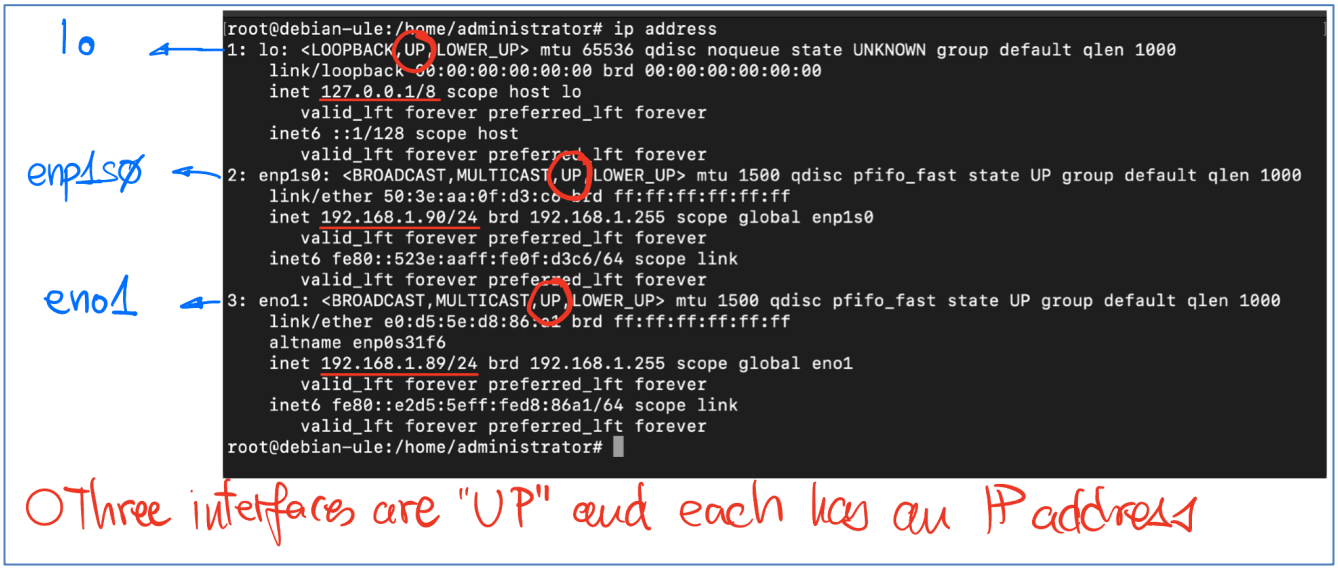

Figure 3. Example of using the ip address command to list net the interfaces in the UP state

e. Choose one of the physical interfaces in your system for this exercise, for example, in the case in fig. 3, we could choose enp1s0.

f. Build the following static IP configuration in your host. If you're using one of Lab B6 PCs, you must edit file /etc/network/interfaces, appropriately and in super-user mode (**$ su**). Set the IP address assigned to you above to the device chosen in the preceding step, enp1s0:

192.168.1.<Your exclusive Number 35..75>

g. Set the Netmask as: 255.255.255.0

h. The default gateway must be set to IP address 192.168.1.1 as illustrated above. The gateway specification in /etc/network/interafces is attained by the use of the following clause:

gateway 192.168.1.1

i. The DNS (Domain Name System) nameserver is specified by the following clause:

dns-nameservers 192.168.1.1

j. Check the interfaces file syntax once again and reboot your system

Exercise 3. IP connectivity checks after rebooting your system

a. Verify that your host has connectivity to localhost ($ ping 127.0.0.1)

b. Execute command the /usr/sbin/route -n to print out your host's IP routing table. Observe the line identified with label 0.0.0.0; that line identifies your host's default gateway as specified in /etc/network/interfaces. What's its IP address? (It appears in the column titled Gateway)

c. Verify that you have connectivity to your default gateway ($ ping 192.168.1.1)

d. If the preceding step was successful, test connectivity to a host located elsewhere in Internet. For example, test IP address 141.101.90.98: $ ping 141.101.90.98

Exercise 4. Using the tcpdump sniffer for observing the ICMP packets as they cross your stack (ICMP – IP – SUB)

In practices 1 and 2, we used the ping utility in various exercises. The ping program, or utility, is used for a variety of purposes, most of them related to the ICMP protocol. Today, again we are to illustrate that it's mandatory for every Internet protocol stack to implement ICMP. Before proceeding to the exercises, observe the protocol stack in fig.4, it is remarkable that the ping application sends ICMP echo messages, and therefore, ping uses the service interface of IP (An instance of a PF_PACKET raw socket. Furthermore, it is of import here that you observe as well that the tcpdump packet sniffer uses exactly the same IP service interface for scanning the packets that are crossing your protocol stack (Packets that are being sent and/or received by the application programs running on top of your stack).

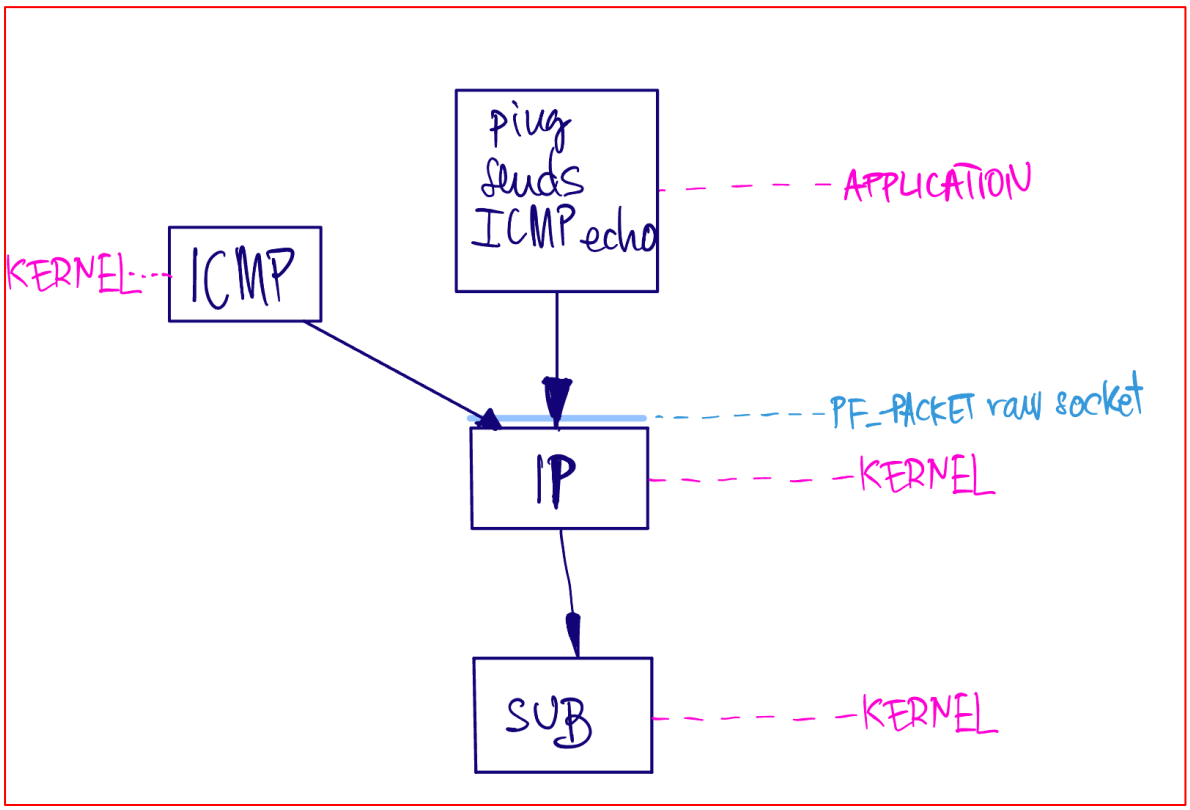

Figure 4. The ping application sends ICMP echo messages

a. According to the protocol stack in fig. 4, what is the program responsible for handling the reception of ICMP echos and for sending back the corresponding ICMP Replies? Observe further that the responder is not the ping utility, that is, the ping utility running on the server side.

b. Request a new terminal window which we'll call Term 2 to view the packets sent and received by the ping application as they cross your host's stack (The ping utility is started in Term 1 in step c below). To this end, run the tcpdump sniffer as root user by issuing these two commands in succession:

```
$ su
# /usr/bin/tcpdump -i <net device chosen above> -vvv -n -eX icmp
```

c. On Term 1, verify that you have connectivity beyond your local net by using ping with option -c (Count). An IP address you could test connectivity to is 193.146.96.2, or 193.146.96.3. Send a single echo packet by including the "-c 1" option of ping, which limits the number of packets sent to 1, so that the resulting tcpdump trace is easier to interpret:

```
$ ping -c 1 193.146.96.3
```


d. Use the illustration in Fig. 4 to comprehend the results that you have obtained. Check out the following aspects from fig. 4. What's the software entity responsible for providing the echo replies? Draw its protocol stack, below:

- tcpdump prints out a block of readable text at the heading which contains a summary of the activated protocol stack, this time including the Subnet layer frame (Note that we included the -eX options in this case)
- Next to that block of text, tcpdump prints out, in the HEX base (Base 16), the full encapsulation hierarchy beginning with the Subnet layer (Layer 1), followed by layer 2, etc.
- Each row contains 16 bytes, indexed by the numbers from the first column (0x0000). The second row's index is 0x010. Check that you understand why.

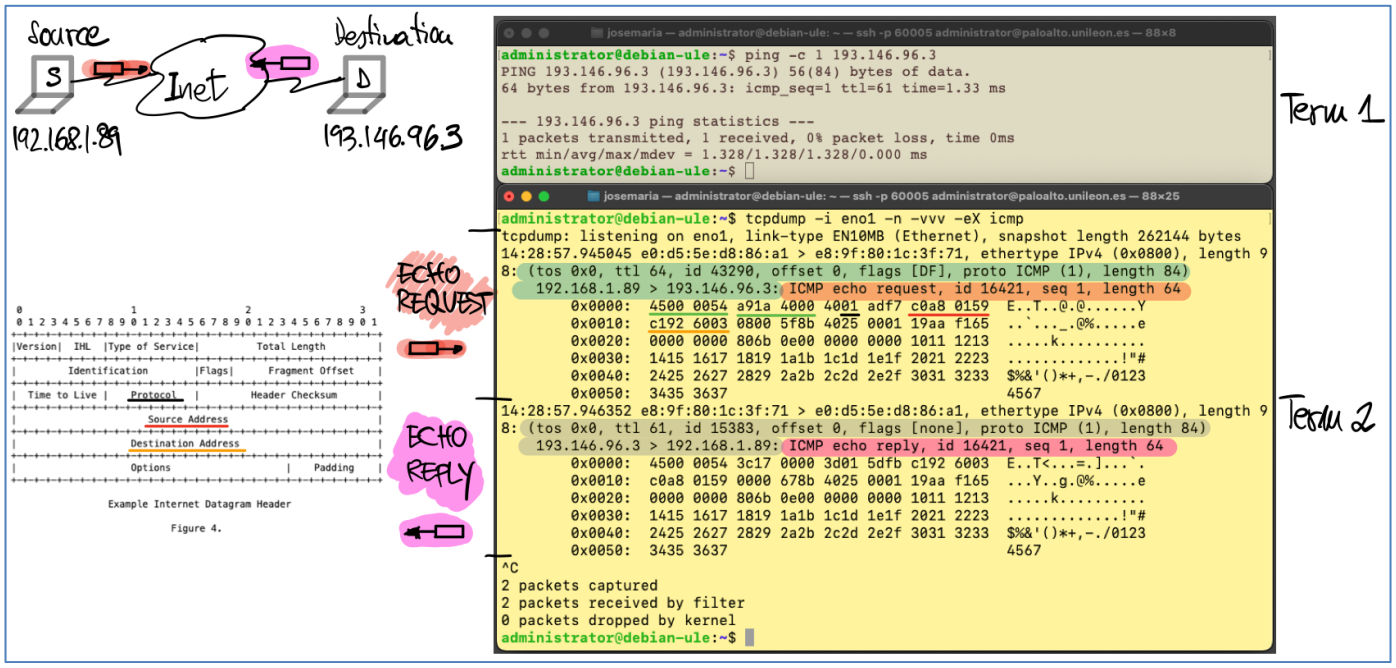

Figure 4. Decoding a reference tcpdump trace of an ICMP Echo/Echo Reply

e. [Home] Explain the tcpdump trace (Term 2) by following the explanation offered on the board and on the Lab B6 TV screen.

f. [Home] Draw the echo sender's protocol stack activated by its executing the ping utility:

g. [Home] Also, draw the protocol stack activated when the responder responds to the received echo requests

Exercise 5. Observing a WOL (Wake On LAN) packet sent from within Lab B6 network.

In this practice we use a program which name is the word 'magic'. This program, which we have written for use in these practices, sends a standard Ethernet frame known as *magic packet.* This frame, if correctly built and sent to the Ethernet broadcast address (ff:ff:ff:ff:ff:ff) is received by all of the hosts connected to the LAN to which the sender belongs and can power up one of those hosts (Wake that host up; in brief, have that host *wake on lan*). In an upcoming practice, we'll study how to build *magic* and we'll delve into the detailed structure of the magic packet. In this practice, we limit ourselves to running magic, observe what it sends and verify whether or not the intended hosts finally powers up.

a. Download the program from this URL: 192.168.1.89/cn/magic or paloalto.unileon.es/cn/magic

b. In Term 2, run the following tcpdump command (Check first the location of tcpdump in your concrete system. Use $ whereis tcpdump if necessary)

```
[Term 2] $ su

[Term 2] # tcpdump -i -n -vvv -eX enp1s0 ether proto 0x0842
```

c. In Term 1, run magic; for the time being, use an inexistent MAC address as destination (`12:34:56:78:90:af`), one that allows us to observe what the magic program sends:

```
[Term 1] $ su

[Term 1] # ./magic enp1s0 12:34:56:78:90:af
```

d. Collect all of the bytes captured by tcpdump and prove that they are conformant with the magic packet as explained above. Provide an explanation of your observations, below:

e. Coordinate with another classmate that is using one of those large tower PCs in Lab B6. Connect remotely to their PC and *copy* the MAC address of NIC eno1:

```
$ [Term 1] ssh administrator@192.168.1.<N of your classmate's PC>
(Enter password XXXcb1x2q%)

$ [At classmate's] /usr/sbin/ifconfig eno1
(Copy MAC address)

[At classmate's] su
(Enter password XXXcb1x2q%)

[At classmate's] /usr/sbin/shutdown -h now
(The remote system powers down, now)
```



```
(Send the magic packet to power up your classmate's PC]
$ [Term 1] ./magic enp1s0 <Classmate's eno1 MAC>
```

f. Was the PC powered up shortly after receiving the magic packet? After about 2 minutes, send a ping to your classmate's PC to check whether or not it successfully booted up. Record the responses below. You may want to retry sending the magic packet if the first one failed.

```
$ [Term 1] ping -c 2 <Classmate's IP address>
```

