From CN to DS

- IP protocol
  - PDU is *packet*
  - SRC IP
  - DST IP
  - Protocol (Multiplexing key)

- UDP datagram
  - PDU is *datgram*
  - SRC Port
  - DST Port (Multiplexing key)

Exercise 1: Datagram capture

- 1. Install tcpdump if necessary
  - # apt-get update
  - # apt-get upgrade
  - # apt-get install tcpdump

Exercise 1: Datagram capture

- 2. Initiate capture with tcpdump
  - Specify interface with **–i** option; for example:

    `$ tcpdump –i eno1 …`

```
$ sudo tcpdump -c 1 -X -vvv -n udp dst port 50001
Password:
tcpdump: data link type PKTAP
tcpdump: listening on pktap, link-type PKTAP (Apple DLT_PKTAP), capture size 262144 bytes
13:33:16.347207 IP (tos 0x0, ttl 64, id 17907, offset 0, flags [none], proto UDP (17),
length 41)
    192.168.2.109.1022 > 193.146.101.46.50001: [udp sum ok] UDP, length 13
      0x0000:  7056 81c4 dfba 2837 3703 3042 0800 4500  pV....(77.0B..E.
      0x0010:  0029 45f3 0000 4011 4afb c0a8 026d c192  .)E...@.J....m..
      0x0020:  652e 03fe c351 0015 d2ae 4865 6c6c 6f20  e....Q....Hello.
      0x0030:  776f 726c 6421 0a                        world!.
```

Exercise 1: Datagram capture

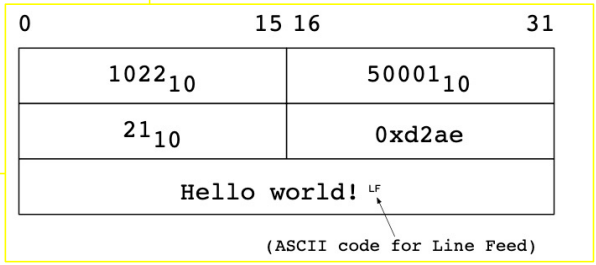
- 3. Send one UDP datagram with the netcat command (nc) on a separate terminal
  - Install nc if necessary
  - Use source port higher than 1024 (Lower ports are reserved)
  - Check which UDP ports are available in your machine with:

```
$ netstat –a –n –udp
…
$ echo “Hello world!” | nc –u –p 1026 paloalto.unileon.es 50001
```

Exercise 1: Datagram capture

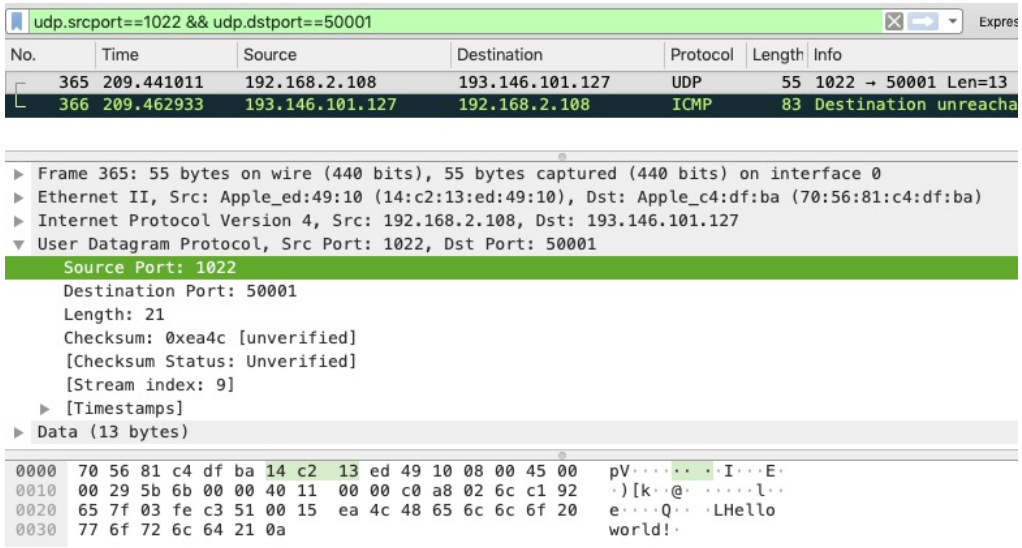
- 4. Interpretation of the results
  - Check the contents of the received datagram are consistent with what you sent with nc
  - In the example below, the source port was **1022**!
  - The dest port was **50001**, etc

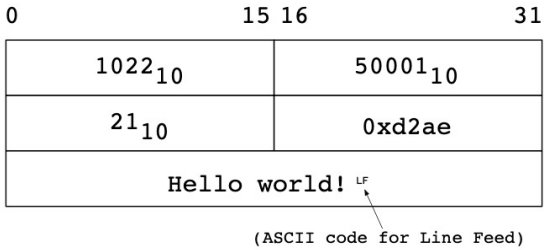
```
$ sudo tcpdump -c 1 -X -vvv -n udp dst port 50001
Password:
tcpdump: data link type PKTAP
tcpdump: listening on pktap, link-type PKTAP (Apple DLT_PKTAP), capture size 262144 bytes
13:33:16.347207 IP (tos 0x0, ttl 64, id 17907, offset 0, flags [none], proto UDP (17),
length 41)
    192.168.2.109.1022 > 193.146.101.46.50001: [udp sum ok] UDP, length 13
      0x0000:  7056 81c4 dfba 2837 3703 3042 0800 4500  pV....(77.0B..E.
      0x0010:  0029 45f3 0000 4011 4afb c0a8 026d c192  .)E...@.J....m..
      0x0020:  652e 03fe c351 0015 d2ae 4865 6c6c 6f20  e....Q....Hello.
      0x0030:  776f 726c 6421 0a                        world!.
```



Exercise 1: Datagram capture

- 5. Repeat the sending and the capture with Wireshark and check the results are the same as those obtained with tcpdump



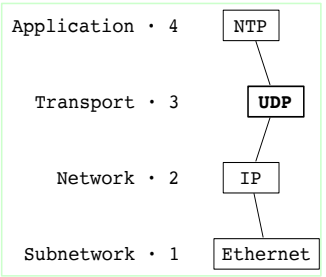


Exercise 2: ntp protocol stack

- 1. Obtain the protocol stack of the ntp (Network Time Protocol)
- 2. Start the capture and wait for your system to attempt *clock synchronization* with its time server (You can use Wireshark, also)

```
root@protocol:/home/networks# tcpdump  -c 2 -n -vv 'udp port 123'
```

- 3. Check that the protocol stack of ntp is this one:



- 4. Transcribe the results to your LabBooks.