

## Master Course Computer Networks

### Homework 1

(submission until November 5th into INBOX located in front of 03.05.052)

**Note:** Subproblems marked by \* can be solved without preceding results.

### Understanding encapsulation (Doing what Wireshark does)

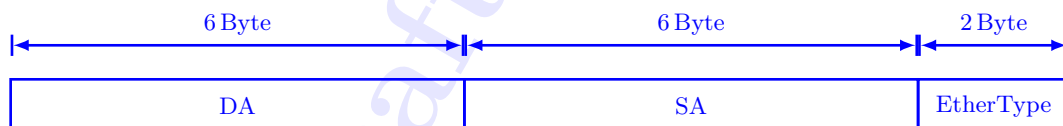
Figure 1 shows the hexdump of some frame captured on a wired network (Ethernet II frame format). The dump contains the whole frame (except its FCS) beginning with the target MAC address. Now we will figure out the contents ...

```

0000  00 25 90 57 1f dc 28 37  37 02 32 41 08 00 45 00
0010  00 42 99 a8 00 00 40 11  b6 9e 83 9f 14 59 83 9f
0020  0e cd d4 1e 00 35 00 2e  c2 25 c2 51 01 00 00 01
0030  00 00 00 00 00 00 06 73  6c 61 63 6b 79 03 6e 65
0040  74 02 69 6e 03 74 75 6d  02 64 65 00 00 01 00 01
  
```

Figure 1: Hexdump, leftmost column indicates the hex offset from the beginning of the frame.

a)\* Sketch the Ethernet II frame format, i.e. header fields and their length.



The header is prepended by a 7 Byte preamble and a 1 Byte start frame delimiter which are considered the physical layer header and thus not belongs to the frame as seen by the data link layer. The header is followed by up to 1500 Byte payload. At the end of the frame, there is the frame check sequence (FCS) which is also called trailer. Optionally, the header may include a 4 Byte VLAN tag field between SA and EtherType. The maximum frame length (without PHY header) is thus 1522 Byte with and 1518 Byte without VLAN tag.

b)\* What is the FCS being used for?

Error detection (not correction).

Here is a list of RFCs that might be helpful in decoding the frame:

- <http://www.ietf.org/rfc/rfc791.txt>
- <http://www.ietf.org/rfc/rfc768.txt>

- <http://www.ietf.org/rfc/rfc1034.txt>

The following two links help you figuring out which protocols are encapsulated by the Ethernet header:

- <http://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xml>
- <http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml>

c) Figure out everything about this frame you can!

```

▶ Frame 1: 80 bytes on wire (640 bits), 80 bytes captured (640 bits)
▼ Ethernet II, Src: Apple_02:32:41 (28:37:37:02:32:41), Dst: SuperMic_57:1f:dc (00:25:90:57:1f:dc)
  ▶ Destination: SuperMic_57:1f:dc (00:25:90:57:1f:dc)
  ▶ Source: Apple_02:32:41 (28:37:37:02:32:41)
  Type: IP (0x0800)
▼ Internet Protocol Version 4, Src: 131.159.20.89 (131.159.20.89), Dst: 131.159.14.205 (131.159.14.205)
  Version: 4
  Header length: 20 bytes
  ▶ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 66
  Identification: 0x99a8 (39336)
  ▶ Flags: 0x00
  Fragment offset: 0
  Time to live: 64
  Protocol: UDP (17)
  ▶ Header checksum: 0xb69e [correct]
  Source: 131.159.20.89 (131.159.20.89)
  Destination: 131.159.14.205 (131.159.14.205)
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
▼ User Datagram Protocol, Src Port: 54302 (54302), Dst Port: domain (53)
  Source port: 54302 (54302)
  Destination port: domain (53)
  Length: 46
  ▶ Checksum: 0xc225 [validation disabled]
▼ Domain Name System (query)
  Transaction ID: 0xc251
  ▶ Flags: 0x0100 Standard query
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  ▶ Queries
0000  00 25 90 57 1f dc 28 37 37 02 32 41 08 00 45 00  .%.W..(7 7.2A..E
0010  00 42 99 a8 00 00 40 11 b6 9e 83 9f 14 59 83 9f  .B...@.....Y..
0020  0e cd d4 1e 00 35 00 2e c2 25 c2 51 01 00 00 01  ...5...%.Q....
0030  00 00 00 00 00 00 00 73 6c 61 63 6b 79 03 6e 65  ....s.lucky.ne
0040  74 02 69 6e 03 74 75 6d 02 64 65 00 00 01 00 01  t.in.tum.de....

```

If you don't know what Wireshark is, it's time to figure it out. Play around with this tool. In case you are using a good OS, you might also want to have a look at `tcpdump` (might be useful for the project).