Problems: TCP and UDP

1. The TCP sliding windows are *byte-oriented*. What does this mean?
2. A TCP connection is using a window size of 10 000 bytes, and the previous acknowledgment number was 22 001. It receives a segment with acknowledgment number 24 001. Draw a diagram to show the situation of the window before and after.
3. A client uses UDP to send data to a server. The data are 16 bytes. Calculate the efficiency of this transmission at the UDP level (ratio of useful bytes to total bytes).
4. Redo Exercise 3. calculating the efficiency of transmission at the IP level using
   (a) IPv4 packets (assume no options)
   (b) IPv6 packets (assume no extension headers)
5. A client uses TCP to send data to a server. The data are 16 bytes. Calculate the efficiency of this transmission at the TCP level (ratio of useful bytes to total bytes, assume no options).
6. Redo Exercise 5., calculating the efficiency of transmission at the IP level using
   (a) IPv4 packets (assume no options)
   (b) IPv6 packets (assume no extension headers)
7. Comment on the results in Exercises 3.–6.. Give examples of applications where it is good to use UDP and TCP respectively.
8. The following is a dump of a UDP header in hexadecimal form:
   `06 32 00 0D 00 1C E2 17`
   What is the
   (a) Source port number
   (b) Destination port number
   (c) Total length of the UDP
   (d) Length of tha data
   (e) Considering that an IP frame can have a maximum total length of 65 535 bytes, what is the maximum length of the data in a UDP frame?

Solutions: TCP and UDP

1. It means that the sequence and acknowledgement numbers refer to bytes instead of segments. For example, the value of the ack-field in a segment defines the number of the next byte a party expects to receive.

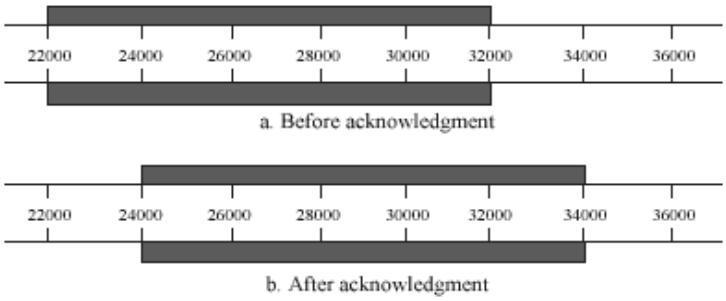

Figure 1: Sliding window of Exercise 2..

2. Presumptions: window size = 10000 bytes, previous acknowledgement 22001, receives acknowledgement 24001. The movement of the window can be seen in Figure 1.

3. Data are 16 bytes, length of UDP header is 8 bytes, so the ratio is $\frac{16}{16+8} = \frac{2}{3}$.

4. Data are 16 bytes, length of IP header (no options) + UDP header is 28 bytes, so the ratio is $\frac{16}{28+16} = 0.364$.

5. Data are 16 bytes, length of TCP header (no options) is 20 bytes, so the ratio is $\frac{16}{20+16} = \frac{4}{9}$.

6. Data are 16 bytes, length of IP header (no options) + TCP header (no options) is 40 bytes, so the ratio is $\frac{16}{16+40} = 0.286$.

7. UDP is a very simple protocol with minimal overhead. If a process wants to send a small message and does not care much about reliability, it can use UDP. Sending a small message using UDP takes much less interaction between sender and receiver than using TCP. UDP is used in multimedia and multicast applications, such as multiplayer games. If reliability is wanted on the other hand, TCP should be chosen. FTP and Telnet use TCP as transport layer protocol.

8. The UDP header has four parts, each of two bytes. That means we get the following interpretation of the header.

   (a) Source port number $= 0632_{16} = 1586$

   (b) Destination port number $= 000D_{16} = 13$

   (c) Total length $= 001C_{16} = 28$ bytes

   (d) Since the header is 8 bytes the data length is $28 - 8 = 20$ bytes.

   (e) The IP header is minimum 20 bytes, which gives the maximum payload 65515 bytes. To fit a UDP frame in this with header of 8 bytes we get data $65515 - 8 = 65507$ bytes.