Opgaveregning 9 (07-04-2020)

## Chapter 2

### Problem 30

**We have seen that Internet TCP sockets treat the data being sent as a byte stream but UDP sockets recognize message boundaries. What are one advantage and one disadvantage of byte-oriented API versus having the API explicitly recognize and preserve application-defined message boundaries?**

Byte-oriented:  
Når man sender store mængder af data, så behøver applikationen ikke at skulle dele dataene op i fragmenter, og holde styr på at sætte det sammen igen. Det samme gælder med ”reordering”, da man bare vil bruge byte ”sequence” nummeret for at aflevere det i den rigtige rækkkefølge

Communication preserving message boundaries:  
Mere control over hvornår applikations lageret får dataene, samt at der aldrig vil være to ”send” sammen med en ”recieve” samt den anden vej rundt.

## Chapter 3

### Review 14

**True or false?**

1. **Host A is sending Host B a large file over a TCP connection. Assume Host B has no data to send Host A. Host B will not send acknowledgments to Host A because Host B cannot piggyback the acknowledgments on data.**

Forkert, Host B behøver ikke data for at gøre dette.

1. **The size of the TCP rwnd never changes throughout the duration of the connection.**

Forkert, den kan godt ændre sig

1. **Suppose Host A is sending Host B a large file over a TCP connection. The number of unacknowledged bytes that A sends cannot exceed the size of the receive buffer.**

Rigtigt, dette er ikke muligt

1. **Suppose Host A is sending a large file to Host B over a TCP connection. If the sequence number for a segment of this connection is *m*, then the sequence number for the subsequent segment will necessarily be *m* + 1.**

Forkert, “payloadet” kan godt være mere end en byte, så dette passer ikke

1. **The TCP segment has a field in its header for rwnd.**

Rigtigt

1. **Suppose that the last SampleRTT in a TCP connection is equal to 1 sec. The current value of TimeoutInterval for the connection will necessarily be Ú 1 sec.**

Rigtigt

1. **Suppose Host A sends one segment with sequence number 38 and 4 bytes of data over a TCP connection to Host B. In this same segment the acknowledgment number is necessarily 42.**

Forkert, men det ville være det rigtige nummer for den ACK som B kunne sende tilbage.

### Review 15

**Suppose Host A sends two TCP segments back to back to Host B over a TCP connection. The first segment has sequence number 90; the second has sequence number 110.**

1. **How much data is in the first segment?**

20 bytes

1. **Suppose that the first segment is lost but the second segment arrives at B. In the acknowledgment that Host B sends to Host A, what will be the acknowledgment number?**

90, for at dataene vil blive gensendt.

### Problem 27

**Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.**

1. **In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number?**

Sequence nummeret er 207, det andet er det samme som ovenfor: altså source: 302, destination: 80.

1. **If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number, the source port number, and the destination port number?**

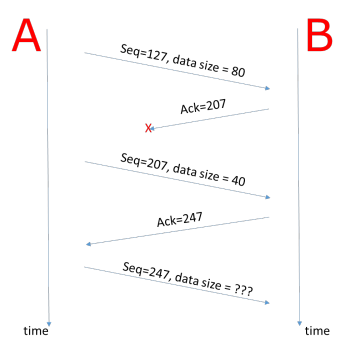
ACK nummer: 207, source: 302, destination: 80.

1. **If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?**

ACK 127, fordi den stadig mangler data fra 126-206.

1. **Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and acknowledgments sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the acknowledgment number.**

Yoinker bare den fra solutions, er doven.



### Problem 44

**Consider sending a large file from a host to another over a TCP connection that has no loss.**

1. **Suppose TCP uses AIMD for its congestion control without slow start. Assuming cwnd increases by 1 MSS every time a batch of ACKs is received and assuming approximately constant round-trip times, how long does it take for cwnd increase from 6 MSS to 12 MSS (assuming no loss events)?**

Det vil tage 6 RTT’s

1. **What is the average throughout (in terms of MSS and RTT) for this connection up through time = 6 RTT?**

(6+12) MSS / RTT

## Praktisk opgave

Part 1) Implement a very simple web-server using TCP sockets.

* You decide whether you want to use C or Javascript ( require('net');)
* The server should be able serve a get request for a single resource (eg "/index.html"), and send a "Hello World" HTML document back.
* Just store the html document as a fixed string in the server application
* Point your browser to the server, and view the resource.
* The server need only server a single client at a time.

Part 2) Implement a client application for your web-server,

* The client should get the resource, and print the html on the console/command line.