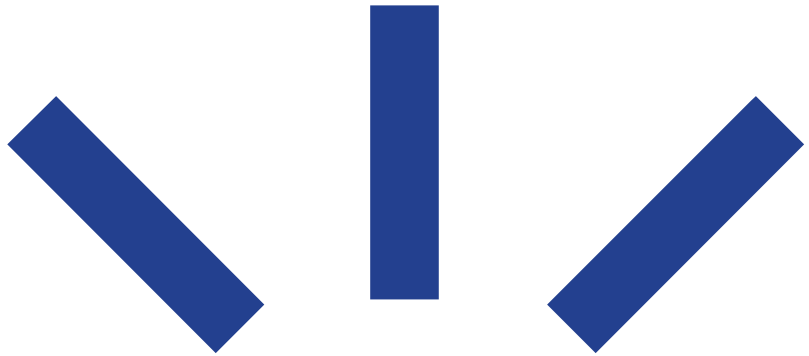


521150A Introduction to Internet

Exercise 2B

Welcome to calculation exercises

In this non-mandatory part of the course, you will learn to

- answer concrete, numerical questions on the subject matter
- model different network scenarios
- use the presented algorithms and methods

Contact info for issues with these exercises

- email: lauri.haverinen@oulu.fi
- room: TS387 (prefer email for contacting)

Before the session

Complete pre-exercises in Moodle

- you will be better prepared for the exercise
- you can earn up to **1 point for final grading** (in Moodle, the points are scaled by 10)

During the session

Follow, participate and solve

- revisiting pre-exercises
- solving example problems together
- describing and going through how to solve the advanced problems

After the session

Solve the rest of the problems presented in this document

- return a scanned version (or good photo) as PDF of your hand-written solutions **with your name on it** to Moodle **before the next exercise**
- you can earn up to **0.5 points for each solved problem**, so a maximum of 1.5 points per exercise (in Moodle, the points are scaled by 10)

Pre-assignments

1. **Which of the statements are true?**
   - Kuljetuskerroksen protokollat toteutetaan päätelaitteen käyttöjärjestelmässä - Transport layer protocols are implemented at operating system level
   - Kuljetuskerroksen protokollat yhdistävät päätelaitteiden prosesseja - Transport layer protocols interconnect application processes of hosts

2. **Which of the UDP related statements are false?**
   - UDP-otsikko on 8 bittiä pitkä - The size of a UDP header is 8 bits

3. **Which of the TCP related statements are FALSE?**
   - TCP-protokolla ei takaa pakettien saapumisjärjestystä - TCP does not guarantee that packets are delivered in the same order they were sent
   - Ruuhkanhallinnan ansiosta TCP voi aloittaa lähettämisen täydellä nopeudella kättelyn jälkeen - Because of congestion control, TCP can begin transmitting at maximum transfer rate after handshaking

Pre-assignments

4. **Round-trip delay time of the TCP connection is 10 ms, and the connection is not affected by congestion. The receiver window size is 24 kB and the maximum segment size is 2 kB. Assuming the TCP connection has already been established, and the senders congestion window is 1MSS, how long does it take with the TCP slow start until the first full window can be sent, if the protocol wouldn't move to the congestion avoidance phase?**
   - Initial CWND (congestion window) = 1 MSS (maximum segment size), CWND doubles every RTT
   - 10ms (1MSS=2kB) -> 10ms (2MSS=4kB) -> 10ms (4MSS=8kB) -> 10ms (8MSS=16kB) -> after which a full window would be sent, so **40 ms** without congestion avoidance
   - **with congestion avoidance**: SSTHRESH = RWND/2=12kB => 10ms (1MSS=2kB) -> 10ms (2MSS=4kB) -> 10ms (4MSS=8kB) -> 10ms (6MSS=12kB) -> 10ms (7MSS=14kB) -> 10ms (8MSS=16kB) -> 10ms (8MSS=18kB) -> 10ms (10MSS20kB) -> 10ms (11MSS=22kB) -> after which 24kB would be sent, so **90 ms with congestion avoidance**

5. **TCP congestion window size is set to 18 kB. Retransmission timer times out because timeout is experienced. What is the size of the congestion window after four subsequent segments are received successfully (i.e. ACK has been received)? The maximum segment size (MSS) is 1 kB, and assume that the ACK's arrive in the same order as segments have been sent.**
   - Timeout -> CWND = 1MSS, **each time ACK is received for a segment -> CWND=CWND+1MSS**, 8 kB after fourth window, not segment!
   - Once the next four segments have been sent and the ACK's have been received, the congestion window size is 5kB

Pre-assignments

6. **What is the maximum possible transmission rate (Mbps) that one can use to send 1300 byte TCP payloads (+ headers), without that the TCP sequence number wraps around when the maximum segment lifetime is 120 seconds?**

   - Segment payload size = 1300B
   - Headers (without options) = 58B (Ethernet 18B; IPv4 20B; TCP 20B), also accepted if 64B was used for headers (size of empty Ethernet frame)
   - Sequence numbers are 32-bit numbers, so the amount of sequence numbers = 2^32 = 4,294,967,296
   - Sequence number is the byte stream number of the first byte in the data segment, so the amount of segments = (2^32)/1300B
   - Headers as overhead for each segment = 58B*((2^32)/1300B)
   - We can send 2^32 B of data with added headers for each payload size, so **packet** size (bits) = ((2^32)+58B*(2^32)/1300B)*8

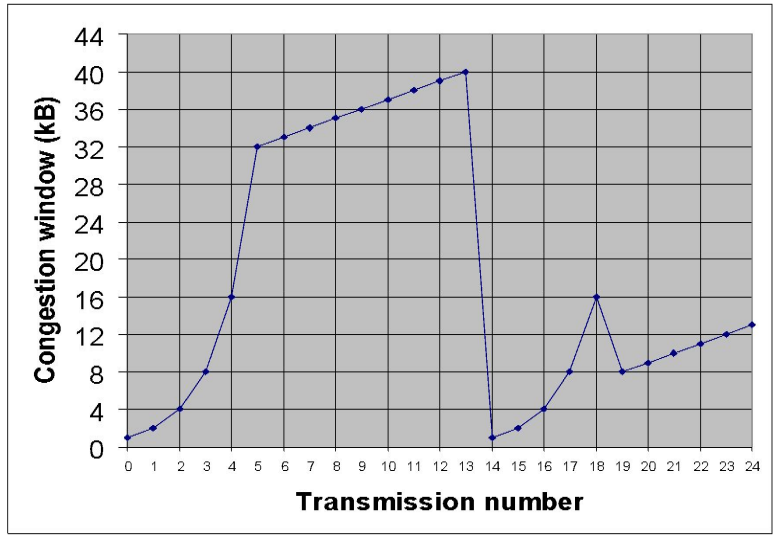
   - Packet should be transmitted during segment lifetime, so required transmission rate < ((2^32)+58B*(2^32)/1300B)*8 / 120 = 299105927.588 ~=**300Mbps** (or 300427455.987 **~301Mbps** with 64kB headers)

Pre-assignments

7. **Client receives data from the server as TCP segments; each segment has payload of 1460 bytes. If TCP acknowledges every other segment, what is the minimum uplink bandwidth required to download payload data at 1 Mbps from the server? Ignore overhead caused by layers below L3 (network layer). Options are not used with TCP and IP.**
   - (1Mbps / (2*1460B*8))*(40B*8) = ~13,7 kbps

8. **A client sends an request to the server using TCP and receives a data object, whose size is 100 kB. The link transmission rate is 10 Mbps bidirectionally. The MSS is 536 bytes and RTT is 100 ms. Assuming that TCP window size does not limit the transmission rate, what is the minimum time required from sending the request to until the the data object is received? Options are not used with TCP and IP. Ignore header overhead and other latencies caused by layers below L3 (network layer). There is no congestion, packets are not dropped and TCP slow start is not in use, i.e. TCP can send at full speed right after connection has been established. Express your answer in milliseconds without decimals rounding to the nearest.**
   - Handshake, request
   - Transmitting the file over the link: (100000B + 100000/536*40B)*8 / 10000000 = 0.08597014925 s
   - => ~286 ms

Problem #C



a) Given the plot of the size of the TCP congestion window to the right, please create and fill the following table (0-5 & 5 is already filled):

Transm. #	MSS (kB)	Congestion window	Slow start threshold	Description
0-5	1	start: 1MSS end: 32MSS	32	CWND (congestion window size) is growing exponentially
5	1	32	32	SST (slow-start threshold) is reached
5-13	1			
13	1			
13-14	1			
14	1			
14-18	1			
18-19	1			
19-24	1			

b) You are hired to design a reliable byte-stream protocol that uses sliding windows (just like TCP). The protocol will run over 1 Gbps network, the RTT of the network is 140 ms, the maximum segment lifetime is 60 seconds and the maximum segment size is 64 kB.

i) How many bits would you allocate for the window size field in the protocol header and why? Explain your solution in few words.

ii) How many bits would you allocate for the sequence number field in the protocol header and why? Explain your solution in few words.

iii) Compare these to actual TCP header field sizes! Would TCP have a problem under these circumstances? Are there solutions?

Problem #D

a) Consider the delay introduced by the TCP slow-start phase. Consider a client and a server directly connected by one link of data rate R. Suppose the client wants to retrieve an object whose size is exactly equal to 15S, where S is the maximum segment size (MSS). RTT denotes the round-trip time between the client and the server, which is assumed to be constant. Ignoring protocol headers, assuming congestion-free and error-free transmission and including TCP connection establishment, determine the time to retrieve the object when:

i) 4S/R > S/R + RTT > 2S/R
ii) 8S/R > S/R + RTT > 4S/R
iii) S/R > RTT

b) Consider requesting and transmitting a 20,000 byte object over a 10 Mbps link where RTT is 10 ms. Maximum segment size is 536 bytes. Take TCP connection establishment and TCP/IP headers, but not L2 (Ethernet) headers into account in your calculations. You can assume error-free transmission.

i) Compute the minimum latency ignoring TCP slow start, i.e. TCP can blast away at full speed right after the connection is established.
ii) Compute the latency taking TCP slow start into account. Assume that the slow start threshold is a very large value that will not be reached.
iii) Compute the latency as in ii), but with a slow start threshold of 4,288 bytes

Problem #E

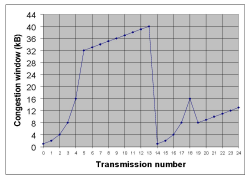
In this problem, you study the performance of TCP in high-speed long-distance networks, assuming a 1 Gbps cross-continental connection with RTT of 60 ms.

a) **What is the maximum window size (in bytes) for TCP without window scaling? Calculate the maximum possible throughput of this link when window scaling is not in use (in Mbits/s).**
b) **Determine the minimum window size that would let TCP achieve maximum possible throughput in this connection using window scaling. What is the window scale factor in that case?**
c) **With slow-start in use, how long it will take to reach maximum window size after a timeout, when the size of the congestion window was 16*MSS when the timeout occurred, and the MSS is** i) 536 bytes **or** ii) 16kB**?**

Problems 2B

Problem #C:

a) **Given the plot of the size of the TCP congestion window, please create and fill the table:**



Transm. #	MSS (kB)	Congestion window	Slow start threshold	Description
0-4	1	Starting with 1MSS, end -> 16MSS	32	Congestion window size is growing exponentially until SST is reached
5	1			
6-12	1			
...	...	...	...	...
19-24	1			

b) **You are hired to design a reliable byte-stream protocol that uses sliding windows (just like TCP). The protocol will run over 1 Gbps network, the RTT of the network is 140 ms, the maximum segment lifetime is 60 seconds and the maximum segment size is 64 kB.**

i) How many bits would you allocate for the window size field in the protocol header and why? Explain your solution in few words.

ii) How many bits would you allocate for the sequence number field in the protocol header and why? Explain your solution in few words.

iii) Compare these to actual TCP header field sizes! Would TCP have a problem under these circumstances? Are there solutions?

Notes for #C

a) what loss events are happening in the figure? Check lecture 10 slides for TCP congestion control

b) start by checking TCP headers, what limitations exist because certain header sizes and how this could be improved for this specific scenario

Problem #D:

a) **Consider the delay introduced by the TCP slow-start phase. Consider a client and a server directly connected by one link of data rate R. Suppose the client wants to retrieve an object whose size is exactly equal to 15S, where S is the maximum segment size (MSS). RTT denotes the round-trip time between the client and the server, which is assumed to be constant. Ignoring protocol headers, assuming congestion-free and error-free transmission and including TCP connection establishment, determine the time to retrieve the object when:**

i) 4S/R > S/R + RTT > 2S/R

ii) 8S/R > S/R + RTT > 4S/R

iii) S/R > RTT

b) **Consider requesting and transmitting a 20,000 byte object over a 10 Mbps link where RTT is 10 ms. Maximum segment size is 536 bytes. Take TCP connection establishment and TCP/IP headers, but not L2 (Ethernet) headers into account in your calculations. You can assume error-free transmission.**

i) Compute the minimum latency ignoring TCP slow start, i.e. TCP can blast away at full speed right after the connection is established.

ii) Compute the latency taking TCP slow start into account. Assume that the slow start threshold is a very large value that will not be reached.

iii) Compute the latency as in ii), but with a slow start threshold of 4,288 bytes

Problem #E: In this problem, you study the performance of TCP in high-speed long-distance networks, assuming a 1 Gbps cross-continental connection with RTT of 60 ms.

a) What is the maximum window size (in bytes) for TCP without window scaling? Calculate the maximum possible throughput of this link when window scaling is not in use (in Mbits/s).

b) Determine the minimum window size that would let TCP achieve maximum possible throughput in this connection using window scaling. What is the window scale factor in that case?

c) With slow-start in use, how long it will take to reach maximum window size after a timeout, when the size of the congestion window was 16*MSS when the timeout occurred, and the MSS is

i) 536 bytes

ii) 16kB

Notes for #D

a) - you can draw a timing diagram to get a better understanding, with only one segment the delay would be RTT+RTT+S/R

- 15S = S+2S+4S+8S

b) calculate amount of segments and headers

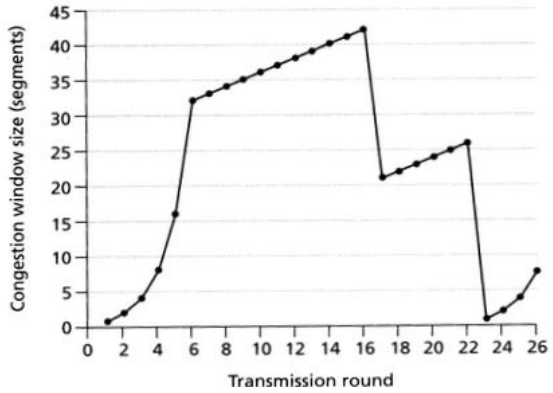
Notes for #E

- without window scaling, TCP window size header is 2B and SEQ 4B

- window scaling can increase the max. window size from 65535B to 1 GB

- TCP reaches max. throughput when wait time is 0, so CWND/R = RTT

Examples



Problems

a) Figure in the top-right corner shows the behaviour of connection utilizing TCP Reno protocol. Describe the events.

b) You are hired to design a reliable byte-stream protocol that uses sliding windows. Assume the following:
- Protocol will run over 56 kbps network
- RTT of the network is 300 ms
- maximum segment lifetime is 1 minute
- maximum segment size is 536 B

How many bits do you need for i) window size ii) sequence number fields in the protocol header?

c) Consider the delay introduced by the TCP slow-start phase, and a client and a server directly connected by one link of data rate R. The client is retrieving an object whose size is exactly equal to 7S, where S is the maximum segment size (MSS). Round-trip time between the client and the server is assumed to be constant. Ignoring protocol headers, assuming congestion-free and error-free transmission and including TCP connection establishment, determine the time to retrieve the object when 4S/R > S/R + RTT > S/R.

d) Consider a 1 Gbps cross-continental connection with RTT of 100 ms.
- i) What is the maximum throughput in this link without TCP window scaling?
- ii) What is the minimum window size required for TCP to achieve maximum possible throughput using window scaling? Solve the scale factor.

Solutions

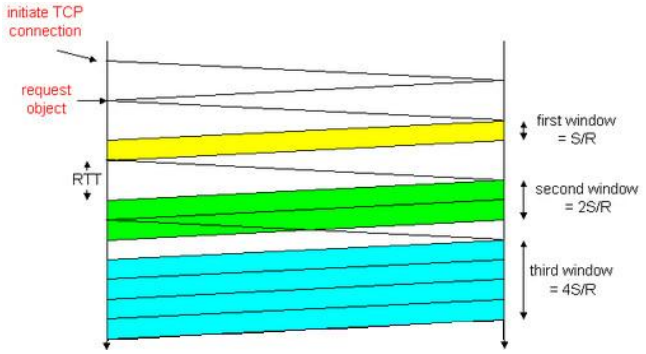
a) [1,6]: TCP slow-start, CWND grows exponentially until SST (32) is reached
[6,16]: SST is reached, CWND grows linearly to avoid congestion
[16,17]: packet loss recognized by triple duplicate ACK, SST set to half of current CWND (42->21)
[17,22]: CWND grows linearly to avoid congestion
[22,23]: packet loss recognized by timeout, CWND set to 1, SST set to half of current CWND (26->13)
[23,26]: TCP slow-start, CWND grows exponentially until SST is reached

b)

i) window size must be RTT*R, so $300\ ms \cdot \frac{56\ kbps}{8} = 2{,}100\ B \rightarrow \log_2(2{,}100) = 11.04 \approx 12$ bits required

ii) we can send $60\ s \cdot \frac{56\ kbps}{8} = 420{,}000\ B \rightarrow \log_2(420{,}000) = 18.68 \approx 19$ bits required

c) 7S=S+2S+4S
initialising TCP connection and requesting the object: RTT + RTT
(S/R+RTT)+(S/R+RTT)+(4S/R) = 6S/R+4RTT



d)

i) $\frac{65{,}535\ B \cdot 8}{100 \cdot 10^{-3}\ s} = 5{,}242{,}800\ bps \approx 5.2\ Mbps$

ii) $\frac{WINDOWSIZE}{1 \cdot 10^9\ bps} = 100\ ms \rightarrow WINDOWSIZE = 100\ ms \cdot 1{,}000{,}000{,}000\ bps = 100\ Mb = 12.5\ MB$

Scale factor: log2(12,500,000)=23.575 ≈ 24 bits => required window size header 24 - default window size header 16 = 8