UNIVERSITY OF OSLO

Faculty of mathematics and natural sciences

Examination in	INF3190/INF4190 — Data Communications
Day of examination:	2nd June, 2004
Examination hours:	9.00 – 12.00

This problem set consists of 6 pages.

Appendices:	None
Permitted aids:	All printed and written material, calculator

Please make sure that your copy of the problem set is complete before you attempt to answer anything.

Problem 1

A frame-oriented data communications system operates at a transmission rate of 512 kb/s with a frame length of 512 bytes over a long-distance link which produces a propagation delay of 20 ms. A flow control system is required using a window mechanism. Determine the minimum window size which allows for optimum throughput.

Problem 2

A point-to-point satellite transmission link connecting two computers uses a stop-and-wait ARQ strategy and has the following characteristics:

- Data transmission rate = 64 kb/s
- Frame size, n = 2048 bytes
- Information bytes per frame, k = 2043 bytes
- One-way propagation delay, td = 180 ms
- Acknowledgement size, a = 10 bytes
- Two-way processing delay, tp = 25 ms

Determine the throughput and link utilization.

Problem 3

A cyclic code (CRC) has a generator polynomial $x^3 + x + 1$. Information bits consisting of 1100 are to be coded and transmitted. Determine:

i. the transmitted codeword

(Continued on page 2.)

ii. the remainder obtained at the receiver if the transmission is error free (show the computation)

iii. the remainder obtained at the receiver if an error occurs in bit 4 (counted from the left)

Problem 4

A message consisting of 2400 bits is to be passed over an internet. The message is passed to the Transport layer which appends a 150-bit header, followed by the Network layer which uses a 120-bit header. Network layer packets are transmitted via two networks, each of which uses a 26-bit header. The destination network only accepts packets up to 900 bits long. How many bits, including headers, are delivered to the destination network?

Problem 5

The computers connected to the small network shown in figure 1 are started at the same time with empty routing tables. They will use original *Distance Vector Routing* and send routing table updates to their neighbors every 50 ms. The numbers at the edges of the graph in figure 1 are the delays between the nodes that are connected through the edge.

i. If you assume that the first routing table update is sent by each node 0 ms after the start of the nodes, what does the routing table at node E look like at time 110 ms after the start?

ii. What is the final routing table of node E?

iii. What happens to the routing table of node E if node B crashes 500 ms after the start? (Explain)

Make the (unrealistic) assumption that a node knows its own distance to a neighbor as soon as it has received the first message from it, but not earlier. Use ∞ to symbolize unknown and infinite distance.

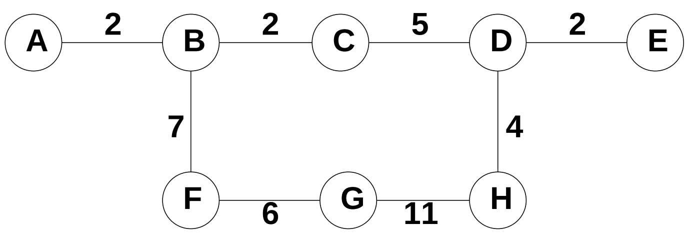

Figure 1: Distance Vector Routing Example

Answer

i. The only trick here is to figure out that the DVR routing updates can not have gone further than 2 hops within 110 ms. Therefore, E can only know about the distance of C, D and H.

(Continued on page 3.)

Destination	Line	Estimated delay
A	*	∞
B	*	∞
C	D	7
D	D	2
E	E	0
F	*	∞
G	*	∞
H	D	6

The fields marked with a * can contain nothing or D. E doesn't know how to route to the destinations in question, but since it has only one outgoing line, it knows that only D is possible.

ii. This is nearly a trick question because the actual route is always going through D. It is not really a trick question because a DVR routing table contains distances.

Destination	Line	Estimated delay
A	D	11
B	D	9
C	D	7
D	D	2
E	E	0
F	D	16
G	D	17
H	D	6

Note: F is reached in 16 ms via E-D-C-B-F.

iii. When B crashes, we get a count-to-infinity problems for the nodes A and B. What happens is that C and D will exchange messages every 50 ms. In every exchange, C will believe that it's distance from A and B, respectively, is that reported by D plus 5 ms, and D will believe that it is C's distance plus 5 ms. So every 100 ms, the assumed distance to A and B increases by 10 ms. Via D, E will receive updated distances for A and B every 50 ms, and the distance will always grow by 5 ms until a predefined finite value is reached that is interpreted as ∞. The text does not say anything about a finite value that is considered ∞, so it is not possible to determine how long it takes before E believes that A and B not reachable. Additionally, the route to F is updated. It is 23 ms.

Problem 6

A networking interface of a machine has the IPv4 address 9.228.12.18 and the netmask 255.255.255.128.

i. What is the class of the network that this IP address belongs to?

ii. What is the address of the subnetwork that this IP address belongs to?

iii. Why is 255.255.128.128 not a legal netmask?

(Continued on page 4.)

Answer

i. Class A

ii. 9.228.12.0

iii. This is a pretty hard, maybe unfair question. We should probably give extra credits when its answered, but make it possible to get 100% if it is not answered. Reasonable answers:

- Because this netmask must start with an unterinterrupted sequences of ones, end with an uninterrupted sequences of zeros, and can't have anything in between.
- Because a bitmask can always be alternatively represented by the number of starting 1-bits. After these one bits, only 0-bits follow. Such a representation is not possible for 255.255.128.128.

Problem 7

TCP's congestion control mechanism is the most important one in the Internet. Consider a newly established TCP session that is used to transmit 532 kbytes in total and that is terminated after that. Assume that the round-trip time is a constant 10 ms, the timeout time constant 25 ms, the message size is 1 kbytes, and the congestion window threshold is 64 kbytes.

i. You have no packet loss at all in either direction. Draw the development of the TCP congestion window at the sender for the entire session. Use time on the X-axis, and the number of packets that are sent at the same time on the Y-axis. Make sure that the exact number of packets is visible in your drawing. Ignore connection setup and connection teardown.

ii. One timeout occurs for the first packet that is sent 70 ms after sending of the first packet of the connection. Explain what happens and draw the transmission of the entire 532 kbytes.

iii. If your sender has 10 Gbytes of data to send, and the bandwidth of the network between sender and receiver is as good as infinite (meaning that there is no loss and no additional delay), the speed for sending this amount of data would still be limited. Which other mechanisms besides congestion control can slow the speed of a TCP transmission down?

Answer

i. The expected solution is shown in figure 2. The congestion window size must start with 1 (packet or kbyte, both is correct), and it must double every round-trip time (10 ms) until it reaches 64. From 64, one packet must be added every round-trip time. 532 kbytes have been transmitted successfully after 120 ms.

(Continued on page 5.)

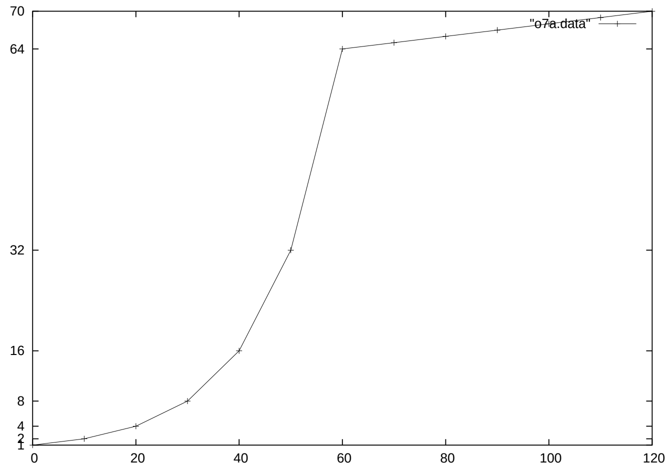

Figure 2: TCP congestion window development for 7(i)

ii. The expected solution is shown in figure 3. The most important details are (1) that the congestion window size restarts at 1 after the timeout, (2) that the transmission after timeout begins with a slow start phase, and (3) that the transmission after timeout leaves the slow start phase. Things to consider:

- The congestion window threshold should be halved after a timeout. Therefore, the congestion window should grow exponentially after the timeout until time 155 ms. Afterwards TCP should leave slow start and the congestion window should grow linearly. If someone gets wrong when he leaves slow start, it should not be heavily punished.
- The text says that the timeout is 25 ms. It should not matter if students get that wrong. One way of getting this wrong would be sending 1 packet at time 80 instead of time 95.
- When the threshold is halved after sending 65 packets, the students can choose 32 or 33 as the new value. 33 is right, but the other option should be allowed.
- The text says that there is a timeout for the first packet that is sent at 70 ms. This implies in fact that no packets of that sequence arrives at the receiver. Therefore, the total number of bytes sent should be 532+65=597. It should not matter if students get this wrong.

iii. Reasonable answers that I can think of right now are:

- Flow control (complete answer)
- Sliding window size (nearly complete answer)
- Limited number of available sequence numbers (nearly complete answer)
- Credit mechanism (nearly complete answer)

(Continued on page 6.)

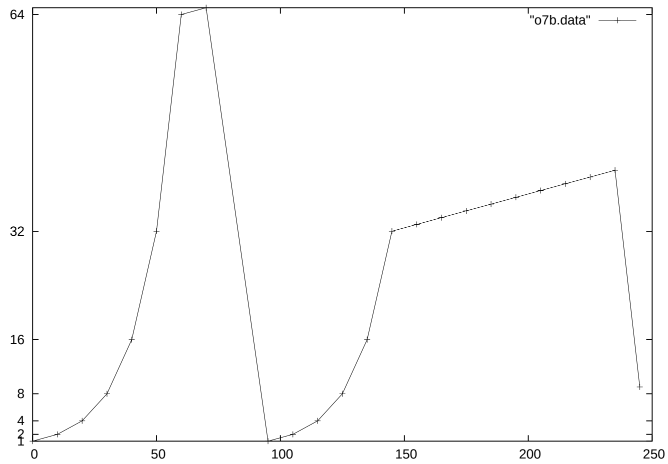

Figure 3: TCP congestion window development for 7(ii)

- Receiver speed and available sequence numbers (complete answer with extras)