Dr. Christian Scheideler
Johns Hopkins University

Baltimore, March 13 2002

Computer Networks

Spring 2002

Midterm Exam

Name:

Course number (344 or 444):

Scores:

1.	:
2.	:
3.	:
4.	:
5.	:
6.	:
7.	:
Total	:

Please write your solutions below the corresponding problems. If you do not have enough space, use the back of the pages. The exam consists of more problems than you may be able to handle in the given time. Consider the easiest ones first.

Good luck!

Problem 1: (10 points)
Give short answers to each of the following questions.

(a) Circle those layers that can (usually) only be found at the network edge:

physical layer, link layer, network layer, transport layer, application layer

(b) What is the difference between transmission and propagation delay?

(c) What is the difference between network applications and application-layer protocols?

(d) Give an example of a stateless application layer protocol.

(e) Give two examples of application layer protocols and their port numbers.

(f) Is it possible to issue more than one question with a single DNS message?

(g) What advantage does a persistent connection with pipelining have over a persistent connection without pipelining?

(h) How does the transport layer manage to perform application multiplexing and demultiplexing?

(i) What is the earliest segment in a TCP session that can carry application data?

(j) Why does TCP wait for three duplicate acknowledgements before retransmitting a packet?

Problem 2: (3 points)
Consider sending a large file of F bits from host A to host B. A and B are connected by a path of ℓ links, and the links are uncongested (that is, no queuing delays). Host A cuts the file into segments of S bits each (we assume here that F is a multiple of S) and adds h bits of header to each segment, forming packets of $S + h$ bits. Each link has a transmission rate of R bps. Find the value of S that minimizes the delay of moving the file from host A to host B. Neglect propagation delay.

Problem 3: (3 points)
Consider an unreliable channel that guarantees a propagation delay of at most d for every packet that gets through, but a packet is only successful with probability p. Furthermore, consider a transport layer protocol that works according to the stop-and-wait principle.

(a) Given that the transmission delay of a packet is t and apart from the propagation delay all other delays (including the transmission delay of its acknowledgement) are negligible, what timeout interval would be the smallest possible for the transport layer to ensure that only packets get retransmitted for which either the packet itself or its acknowledgement got lost?

(b) Using the timeout interval from (a), what is the average time needed for the sender to be sure that it successfully transmitted a packet to the receiver? Use the fact that the expected number of retransmissions of a packet is $(1-p)/p$.

(c) Suppose that $d = t = 1$ ms and $p = 0.8$, and suppose that the application layer generates a new packet every 4 ms. In this case, can the transport layer be expected to keep up with the packet arrival rate from the application layer?

Problem 4: (4 points)
Consider the following state diagrams of protocols for a sender and receiver attempting reliable data transfer over an unreliable channel in which packets may get corrupted but not lost.

(a) Why can they run into a deadlock? It suffices to give an example. (2 points)

(b) How can this flaw be repaired? (2 points)

Problem 5: (3 points)
Compute the checksum of the words 1011010011101000, 0110111011000111, and 1110011100111000 in the way this is done by TCP and UDP. Show all work.

Problem 6: (7 points)

(a) Consider the Go-Back-N and Selective Repeat protocols. Suppose that the sequence number space is of size k. What is the largest allowable window size for each of these protocols that will avoid sequence number problems? (2 points)

(b) What is the difference concerning the way acknowledgements are handled between the Go-Back-N method and the Selective Repeat method?

(c) Which of these two acknowledgement methods is used by TCP?

(d) Is it specified in the TCP RFC how to handle out-of-order segments? If so, how?

(e) Draw a picture for a typical exchange of messages between a client and a server when a TCP connection is closed. For each segment, specify the flags that are set to 1 and sequence numbers that are consistent with the prior segments. (Other aspects such as timeouts can be ignored.) (2 points)

Problem 7: (3 points)
Consider the original (Tahoe) strategy used by TCP to perform congestion control. Imagine a sender and a receiver are connected by a single link that can transmit up to 2^k segments per ms for some integer $k > 0$ and that has a round-trip time of 1 ms (representing here the time when the first bit of the segment is transmitted till its acknowledgement is received) . Furthermore, imagine the situation that TCP always starts with a single segment per ms, that it always uses a threshold value of $2^{k-1} - 1$, and that it backs off every time after it reached the situation that it transmits 2^k segments per ms and no segment ever gets lost. What is the average transmission rate (in the number of segments per ms) of TCP in this case?