Show All Answers Hide All Answers

6.02 Practice Problems: Reliable Data Transport Protocols

Problem 1. Consider the following chain topology:

A ---- B ----- C ---- D ---- E

A is sending packets to E using a reliable transport protocol. Each link above can transmit one packet per second. There are no queues or other sources of delays at the nodes (except the transmission delay of course).

A. What is the RTT between A and E?

Show Answer

B. What is the throughput of a stop-and-wait protocol at A in the absence of any losses at the nodes?

Show Answer

C. If A decides to run a sliding window protocol, what is the optimum window size it must use? What is the throughput achieved when using this optimum window size?

Show Answer

D. Suppose A is running a sliding window protocol with a window size of four. In the absence of any losses, what is the throughput at E? What is the utilization of link B-C?

Show Answer

E. Consider a sliding window protocol running at the optimum window size found in part 3 above. Suppose nodes in the network get infected by a virus that causes them to drop packets when odd sequence numbers. The sliding window protocol starts numbering packets from sequence number 1. Assume that the sender uses a timeout of 40 seconds. The receiver buffers out-of-order packets until it can deliver them in order to the application. What is the number of packets in this buffer 35 seconds after the sender starts sending the first packet?

Show Answer

Problem 2. Ben Bitdiddle implements a reliable data transport protocol intended to provide "exactly once" semantics. Each packet has an 8-bit incrementing sequence number, starting at 0. As the connection progresses, the sender "wraps around" the sequence number once it reaches 255, going back to 0 and incrementing it for successive packets. Each packet size is S = 1000 bytes long (including all packet headers).

Suppose the link capacity between sender and receiver is C = 1 Mbyte per second and the round-trip time is R = 100 milliseconds.

A. What is the highest throughput achievable if Ben's implementation is stop-and-wait?

Show Answer

B. To improve performance, Ben implements a sliding window protocol. Assuming no packet losses, what

should Ben set the window size to in order to saturate the link capacity?

Show Answer

C. Ben runs his protocol on increasingly higher-speed bottleneck links. At a certain link speed, he finds that his implementation stops working properly. Can you explain what might be happening? What threshold link speed causes this protocol to stop functioning properly?

Show Answer

Problem 3. A sender S and receiver R are connected over a network that has k links that can each lose packets. Link i has a packet loss rate of p_i in one direction (on the path from S to R) and q_i in the other (on the path from R to S). Assume that each packet on a link is received or lost independent of other packets, and that each packet's loss probability is the same as any other's (i.e., the random process causing packet losses is indepedent and identically distributed).

A. Suppose that the probability that a data packet does not reach R when sent by S is p and the probability that an ACK packet sent by R does not reach S is q. Write expressions for p and q in terms of the p_i's and q_i's.

Show Answer

B. If all p's are equal to some value $\alpha << 1$ (much smaller than 1), then what is p (defined above) approximately equal to?

Show Answer

C. Suppose S and R use a stop-and-wait protocol to communicate. What is the expected number of transmissions of a packet before S can send the next packet in sequence? Write your answer in terms of p and q (both defined above).

Show Answer

Problem 4. Consider a 40 kbit/s network link connecting the earth to the moon. The moon is about 1.5 light-seconds from earth.

A. 1 Kbyte packets are sent over this link using a stop-and-wait protocol for reliable delivery, what data transfer rate can be achieved? What is the utilization of the link?

Show Answer

B. If a sliding-window protocol is used instead, what is the smallest window size that achieves the maximum data rate? Assume that error are infrequent. Assume that the window size is set to achieve the maximum data transfer rate.

Show Answer

C. Consider a sliding-window protocol for this link with a window size of 10 packets. If the receiver has a buffer for only 30 undelivered packets (the receiver discards packets it has no room for, and sends no ACK for discarded packets), how bits of sequence number are needed?

Show Answer

Problem 5. Consider a best-effort network with variable delays and losses. Here, Louis Reasoner suggests that the receiver does not need to send the sequence number in the ACK in a correctly implemented stop-and-wait protocol, where the sender sends packet `k+1` only after the ACK for packet `k` is received. Explain whether he is correct or not.

Show Answer

Problem 6. Consider a sliding window protocol between a sender and a receiver. The receiver should deliver packets reliably and in order to its application.

The sender correctly maintains the following state variables:

`unacked_pkts` -- the buffer of unacknowledged packets

`first_unacked` -- the lowest unacked sequence number (undefined if all packets have been acked)

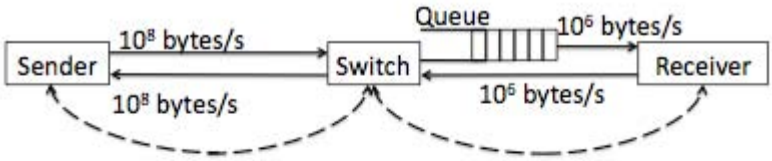
`last_unacked` -- the highest unacked sequence number (undefined if all packets have been acked)

`last_sent` -- the highest sequence number sent so far (whether acknowledged or not)

If the receiver gets a packet that is strictly larger than the next one in sequence, it adds the packet to a buffer if not already present. We want to ensure that the size of this buffer of packets awaiting delivery *never exceeds* a value $W \geq 0$. Write down the check(s) that the sender should perform before sending a new packet in terms of the variables mentioned above that ensure the desired property.

Show Answer

Problem 7. Ben decides to use the sliding window transport protocol we studied in 6.02 and implemented in the pset on the network below. The receiver sends end-to-end ACKs to the sender. The switch in the middle simply forwards packets in best-effort fashion.



Max queue size = 30 packets
Packet size = 1000 bytes
ACK size = 40 bytes
Initial sender window size = 10 packets

A. The sender's window size is 10 packets. Selecting the best answer from the choices below, at what approximate rate (in packets per second) will the protocol deliver a multi-gigabyte file from the sender

to the receiver? Assume that there is no other traffic in the network and packets can only be lost because the queues overflow.

a. Between 900 and 1000.
b. Between 450 and 500.
c. Between 225 and 250.
d. Depends on the timeout value used.

Show Answer

B. You would like to double the throughput of this sliding window transport protocol running on the network shown on the previous page. To do so, you can apply one of the following techniques alone:

a. Double the window size.
b. Halve the propagation time of the links.
c. Double the speed of the link between the Switch and Receiver.

For each of the following sender window sizes, list which of the above techniques, if any, can approximately double the throughput. If no technique does the job, answer "None". There might be more than one answer for each window size, in which case you should list them all. Note that each technique works in isolation. Explain your answers.

1. W = 10.

   Show Answer

2. W = 50.

   Show Answer

3. W = 30.

   Show Answer

Problem 8. Consider the sliding window protocol described in lecture and implemented in the pset. The receiver sends "ACK k" when it receives a packet with sequence number `k`. Denote the window size by W. The sender's packets start with sequence number 1. Which of the following is true of a correct implementation of this protocol over a best-effort network?

A. Any new (i.e., previously unsent) packet with sequence number *greater than* W is sent by the sender if, and only if, a new (i.e., previously unseen) ACK arrives.

Show Answer

B. The sender will never send more than one packet between the receipt of one ACK and the next.

Show Answer

C. The receiver can discard any new, out-of-order packet it receives after sending an ACK for it.

Show Answer

D. Suppose that no packets or ACKs are lost and no packets are ever retransmitted. Then ACKs will arrive at the sender in non-decreasing order.

Show Answer

E. The sender should retransmit any packet for which it receives a duplicate ACK (i.e., an ACK it has received earlier).

Show Answer

Problem 9. In his haste in writing the code for the exponential weighted moving average (EWMA) to estimate the smoothed round-trip time, `srtt`, Ben Bitdiddle writes

```
srtt = alpha * r + alpha * srtt
```

where `r` is the round-trip time (RTT) sample, and `0 < alpha < 1`.

For what values of alpha does this buggy EWMA over-estimate the intended `srtt`? You may answer this question assuming any convenient non-zero sequence of RTT samples, r.

Show Answer

Problem 10. A sender S and receiver R communicate reliably over a series of links using a sliding window protocol with some window size, W packets. The path between S and R has one bottleneck link (i.e., one link whose rate bounds the throughput that can be achieved), whose data rate is C packets/second. When the window size is W, the queue at the bottleneck link is always full, with Q data packets in it. The round trip time (RTT) of the connection between S and R during this data transfer with window size W is T seconds. There are no packet or ACK losses in this case, and there are no other connections sharing this path.

A. Write an expression for W in terms of the other parameters specified above.

Show Answer

B. We would like to reduce the window size from W and still achieve high utilization. What is the minimum window size, Wmin, which will achieve 100% utilization of the bottleneck link? Express your answer as a function of C, T, and Q.

Show Answer

C. Now suppose the sender starts with a window size set to Wmin. If all these packets get ac- knowledged and no packet losses occur in the window, the sender increases the window size by 1. The sender keeps increasing the window size in this fashion until it reaches a window size that causes a packet loss to occur. What is the smallest window size at which the sender observes a packet loss caused by the bottleneck queue overflowing? Assume that no ACKs are lost.

Show Answer

Problem 11. A sender A and a receiver B communicate using the stop-and-wait protocol studied in 6.02. There are n links on the path between A and B, each with a data rate of R bits per second. The size of a data packet is S bits and the size of an ACK is K bits. Each link has a physical distance of D meters and the speed of signal propagation over each link is c meters per second. The total processing time experienced by a data packet and its ACK is T_p seconds. ACKs traverse the same links as data packets, except in the opposite direction on each link (the propagation time and data rate are the same in both directions of a link). There is

no queueing delay in this network. Each link has a packet loss probability of p, with packets being lost independently.

What are the following four quantities in terms of the given parameters?

A. Transmission time for a data packet on one link between A and B.

Show Answer

B. Propagation time for a data packet across n links between A and B.

Show Answer

C. Round-trip time (RTT) between A and B?. (The RTT is defined as the elapsed time between the start of transmission of a data packet and the completion of receipt of the ACK sent in response to the data packetâ€™s reception by the receiver.)

Show Answer

D. Probability that a data packet sent by A will reach B.

Show Answer

Problem 12.

Opt E. Miser implements the 6.02 stop-and-wait reliable transport protocol with one modification: being stingy, he replaces the sequence number field with a 1-bit field, deciding to reuse sequence numbers across data packets. The first data packet has sequence number 1, the second has number 0, the third has number 1, the fourth has number 0, and so on. Whenever the receiver gets a packet with sequence number s (= 0 or 1), it sends an ACK to the sender echoing s. The receiver delivers a data packet to the application if, and only if, its sequence number is different from the last one delivered, and upon delivery, updates the last sequence number delivered.

D. He runs this protocol over a best-effort network that can lose packets (with probability less than 1) or reorder them, and whose delays may be variable. Does the modified protocol always provide correct reliable, in-order delivery of a stream of packets?

Show Answer

Problem 13. Consider a reliable transport connection using the 6.02 sliding window protocol on a network path whose RTT in the absence of queueing is RTTmin = 0.1 seconds. The connection's bottleneck link has a rate of C = 100 packets per second, and the queue in front of the bottleneck link has space for Q = 20 packets.

Assume that the sender uses a sliding window protocol with fixed window size. There is no other connection on the path.

A. If the size of the window is 8 packets, then what is the throughput of the connection?

Show Answer

B. If the size of the window is 16 packets, then what is the throughput of the connection?

Show Answer

C. What is the smallest window size for which the connection's RTT exceeds RTTmin?

Show Answer

Problem 14. TCP, the standard reliable transport protocol used on the Internet, uses a sliding window. Unlike the protocol studied in 6.02, however, the size of the TCP window is variable. The sender changes the size of the window as ACKs arrive from the receiver; it does not know the best window size to use a priori.

TCP uses a scheme called *slow start* at the beginning of a new connection. Slow start has three rules, R1, R2, and R3, listed below (TCP uses some other rules too, which we will ignore).

In the following rules for slow start, the sender's current window size is W and the last in-order ACK received by the sender is A. The first packet sent has sequence number 1.

R1. Initially, set W ← 1 and A ← 0.
R2. If an ACK arrives for packet A+1, then set W←W+1, and set A←A+1.
R3. When the sender retransmits a packet after a timeout, then set W←1.

Assume that all the other mechanisms are the same as the 6.02 sliding window protocol. Data packets may be lost because packet queues overflow, but assume that packets are not reordered by the network.

We run slow start on a network with RTTmin = 0.1 seconds, bottleneck link rate = 100 packets per second, and bottleneck queue = 20 packets.

A. What is the smallest value of W at which the bottleneck queue overflows?

Show Answer

B. Sketch W as a function of time for the first 5 RTTs of a connection. The X-axis marks time in terms of multiples of the connection's RTT. (Hint: Non-linear!)

Show Answer

6.02 Introduction to EECS II: Digital Communication Systems

Fall 2012