COM-208: Computer Networks - Homework 3

1 Application Layer

1. (P22) Consider distributing a file of $F = 15$ *Gbits* to N peers. The server has an upload rate of $u_s = 30$ *Mbps*, and each peer has a download rate of $d_i = 2$ *Mbps* and an upload rate of u. For $N = 10$, 100, and 1000 and $u = 300$ *Kbps*, 700 *Kbps*, and 2 *Mbps*, prepare a chart giving the minimum distribution time for each of the combinations of N and u for both client-server distribution and P2P distribution.

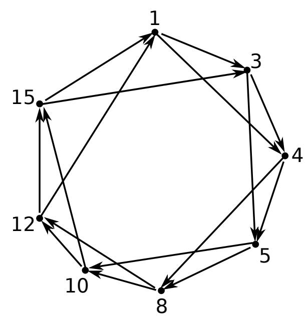


Figure 1: DHT setup for Problem 2.

2. Consider the circular DHT with shortcuts in Figure 1, where each node in the DHT also keeps track of (i) its immediate predecessor, (ii) its immediate successor, and (iii) its second successor (i.e., the successor of the node's immediate successor). Also, consider the following assumptions:

   - Each node keeps track of its predecessor ONLY to know which files it is responsible for. Unlike in lectures, where the name of the node was not anyhow related to the files it is responsible for, in this exercise we assume that every node with ID n and predecessor m contains the files with file ID in the range $[m + 1, n]$.
   - Note that the arrows in Figure 1 are unidirectional. A node cannot send a message to its predecessor. It knows it only to determine which files it is responsible for. For instance, the node 15 would be responsible for files in the range $[13, 15]$.
   - Once a node receives a request for which it is indeed responsible for, assume that it can directly send the file to the node that initiated the request (because the request also contained the source transport address).

a. Suppose that peer 1 wants to learn where file with content ID 9 is stored. Write down the sequence of DHT protocol messages that the nodes exchange until peer 1 discovers the location of the file.
b. Suppose that peer 3 learns that peer 5 has left. How does peer 3 update its successor state information?
c. Now consider that the DHT nodes do not keep track of their second successor. Suppose that a new peer 6 wants to join the DHT and peer 6 initially only knows the IP address of peer 15. What steps are taken?

- (P30) As DHTs are overlay networks, they may not necessarily match the underlay physical network well in the sense that two neighbouring peers might be physically very far away; for example, one peer could be in Asia and its neighbour could be in North America. If we randomly and uniformly assign identifiers to newly joined peers, would this assignment scheme cause such a mismatch? Explain. And how would such a mismatch affect the DHT's performance?

2 Transport Layer

1. (P3) UDP and TCP use 1s complement for their checksums. Suppose you have the following three 8-bit bytes: 00100011, 01001110, 01010100.
   a. What is the 1s complement of the sum of these 8-bit bytes? (Note that although UDP and TCP use 16-bit words in computing the checksum, for this problem you are being asked to consider 8-bit sums). Show all work. *Note: wrap around if the sum overflows.*
   b. With the 1s complement scheme, how does the receiver detect errors? Is it possible that an 1-bit error can go undetected? How about a 2-bit error?
2. (P11) Consider the rdt2.2 receiver in Figure 3.14 and the creation of a new packet in following two self-transitions[1]: *Wait-for-0-from-below* and *Wait-for-1-from-below*.
   a. Would the protocol work correctly if *udt_send(sndpkt)* were removed from the self-transition in the *Wait-for-1-from-below* state? Justify your answer.
   b. What if *udt_send(sndpkt)* were removed from the self-transition in the *Wait-for-0-from-below* state?
   [*Hint*: In this latter case, consider what would happen if the first sender-to-receiver packet were corrupted.]
3. (P12) The sender side of rdt3.0 simply ignores (that is, takes no action on) all received packets that are either in error or have the wrong value in the acknum field of an acknowledgment packet. Suppose that in such circumstances, rdt3.0 were to immediately retransmit the current data packet (instead of waiting for a timeout). Would the protocol still work?
   [*Hint*: Consider what would happen if there were only bit errors; there are no packet losses but premature timeouts can occur. Consider how many times the nth packet is sent, in the limit as n approaches infinity.]

[1] A self-transition is a transition from one state back to itself.

4. (P14) Consider a reliable data transfer protocol that uses only negative acknowledgments. Note that such a protocol is different than the protocol that combines NAK and ACK together. The NAK-only protocol uses the following rules:

   - Receiver sends NAK N if packet N arrived corrupted.
   - Receiver also sends NAK N when it receives packet $N + 1$ without previously receiving packet N.

   Suppose the sender sends data only infrequently.

   a. Would a NAK-only protocol be preferable to a protocol that uses ACKs? Why?

   b. Now suppose the sender has a lot of data to send and the end-to-end connection experiences few losses. In this case, would a NAK-only protocol be preferable to a protocol that uses ACKs? Why?

5. (P15) Consider the cross-country example shown in Figure 3.17. How big would the window size have to be for the channel utilization to be greater than 97 percent? Suppose that the size of a packet is 1200 bytes, including both header fields and data.

6. (P22) Consider the GBN protocol with a sender window size of 4 and a sequence number range of 1024. Suppose that at time t, the next in-order packet that the receiver is expecting has a sequence number of k. Assume that the medium does not reorder messages. Answer the following questions:

   a. What are the possible sets of sequence numbers inside the sender's window at time t? Justify your answer.

   b. What are all possible values of the ACK field in all possible messages currently propagating back to the sender at time t? Justify your answer.

7. (P23) Consider the GBN and SR protocols. Suppose the sequence number space is of size k. What is the largest allowable sender window that will avoid the occurrence of problems such as that in Figure 3.27 for each of these protocols?

8. (P29) It is generally a reasonable assumption when the sender and receiver are connected by a single wire that packets cannot be reordered within the channel between the sender and receiver. However, when the "channel" connecting the two is a network, packet reordering can occur. One manifestation of packet reordering is that old copies of a packet with a sequence or acknowledgment number of x can appear, even though neither the sender's nor the receiver's window contains x. With packet reordering, the channel can be thought of as essentially buffering packets and spontaneously emitting these packets at any point in the future. What care must be taken to guard against such duplicate packets?

9. End-host A is connected to end-host B over a communication channel that has propagation delay d_{prop} seconds and transmission rate R bytes/sec (in both directions). The channel is reliable and does not drop or reorder packets. A wants to send a file of size $size_F$ bytes to B. We define the "duration" of the file transfer as the amount of time that elapses from the moment A transmits the first bit of the file until the moment B receives

the last bit of the file (the duration of the file transfer does not include the amount of time it takes for A to receive the last ACK from B).

(a) What is the minimum possible duration of the file transfer (regardless of which transfer protocol we use)? Assume that the entire file can fit in a single packet.

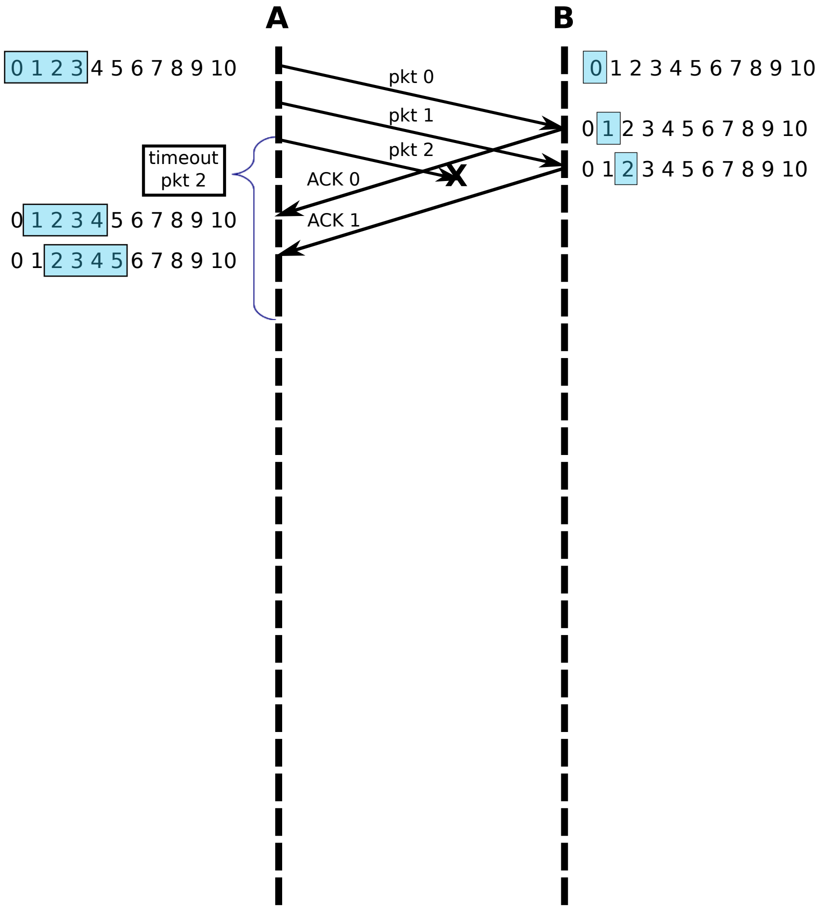
(b) Suppose that the file transfer application uses a stop-and-wait protocol (also referred to as $rdt3.0$ in the book). What is the duration of the file transfer? Assume that the file transfer protocol uses packets of size $size_P$ bytes, and $size_P$ perfectly divides the size of the file, $size_F$ (i.e. $\frac{size_F}{size_P}$ is an integer). Both the size of packet headers and the size of ACKs are insignificant.

(c) Suppose that the file transfer application uses a Go-Back-N (GBN) protocol with window size N. What is the duration of the file transfer? How does your answer compare to the result you found for the stop-and-wait protocol? Assume that the window size N perfectly divides the number of packets that A transmits during the file transfer.

(d) Suppose we want to choose between a GBN and a Selective Repeat (SR) protocol to perform the file transfer over the communication channel described above. Which one is a better choice? Justify your answer.

10. End-host A is communicating with end-host B using a Go-Back-N (GBN) protocol with window size $N = 4$ and valid sequence numbers ranging from 0 to 10. The channel between the two end-hosts never reorders packets, but can drop packets. Suppose B has received and acknowledged packets with sequence numbers 0 and 1, so it is expecting packet with sequence number 2. Instead, it receives packets with sequence numbers 4, 5, 2, and 3 (in this order). Which packets have been lost? What is the response of B when it receives each of the packets, 4, 5, 2, and 3?

    Complete the sequence diagram below by showing all packet exchanges and actions between the sender and the receiver. You should include: *i*) lost packets and ACKs, *ii*) timeout events and *iii*) changes to the sender and receiver window, when packets (including ACKs) are received by either end-host. We have already filled in some of the information: packet 2 got lost, A timed out waiting for an acknowledgment to packet 2.



A
B
0 1 2 3 4 5 6 7 8 9 10
0 1 2 3 4 5 6 7 8 9 10
pkt 0
pkt 1
pkt 2
0 1 2 3 4 5 6 7 8 9 10
0 1 2 3 4 5 6 7 8 9 10
timeout
pkt 2
ACK 0
ACK 1
0 1 2 3 4 5 6 7 8 9 10
0 1 2 3 4 5 6 7 8 9 10