

Computer Networks
Spring 2002
Assignment 6

Problem 18 (4 points):

Consider a scenario in which a host A wants to simultaneously send messages to hosts B and C . A is connected to B and C via a broadcast channel – a packet sent by A is carried by the channel to both B and C . Suppose that the broadcast channel connecting A , B and C can independently lose and corrupt messages (and so, for example, a message sent from A might be correctly received by B but not by C). Design a stop-and-wait-like error-control protocol for reliably transferring a packet from A to B and C , such that A will not get new data from the upper layer until it knows that both B and C have correctly received the current packet. Give FSM descriptions of A and B . (The FSM for C is essentially the same as for B .) Note that since A communicates with two hosts, just passing back an acknowledgement with a sequence number does not suffice any more.

Problem 19 (2 points):

Consider the Go-Back- N protocol with a sender window size of w and a (sufficiently large) sequence number range of N . 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 the message currently propagating back to the sender at time t ? Justify your answer.

Problem 20 (2 points):

Consider the TCP procedure for estimating RTT. Suppose that $0 < x < 1$. Let SampleRTT_1 be the most recent sample RTT, let SampleRTT_2 be the next most recent sample RTT, etc. Derive a formula for EstimatedRTT in terms of sample RTTs under the assumption that n sample RTTs have been obtained so far. Comment on why this averaging procedure is called an exponential moving average.

Problem 21 (2 points):

Consider the figure given in the lecture in which we demonstrated the fairness of TCP's additive increase, multiplicative decrease algorithm for two hosts. Suppose that instead of a multiplicative decrease, TCP decreased the window size by a constant amount. Would the resulting additive increase additive decrease converge to an equal share algorithm? Justify your answer using diagrams similar to the figure mentioned above and distinguishing between the following cases:

- (a) Hosts A and B have the same additive decrease value but start with a different rate.
- (b) Hosts A and B have different additive decrease values but start with the same rate.