College of Computer & Information Science — Fall 2004
Northeastern University — Handout 4
CSU640: Network Fundamentals — 2 October 2004

Problem Set 2 (due at the start of class October 18)

Each problem is worth 10 points.

1. Performance of web caching

Problem 9 of Chapter 2, page 173.

2. Stop-and-wait for broadcast

This problem is the same as Problem 15 of Chapter 3 of text, with some clarifications.

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). Assume that there are no out-of-order message deliveries. Design a stop-and-wait-like protocol for reliably transfering 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 finite-state-machine descriptions for A and B.

Hints: Extend the rdt3.0 protocol studied in class and described in the text. The FSM for B and C are essentially the same; they are also essentially the same as for the receiver in the rdt3.0 protocol.

3. Maximum sequence number in GBN

Consider the GBN protocol with $N = 3$, with no out-of-order arrivals and no bound on the sequence numbers. In the following, assume that the packets and acknowledgments are numbered from 0, and that ACK[i] is a cumulative acknowledgment indicating that packets 0 through i have been received.

(a) Show that if the receiver window contains 6, then PKT[0] cannot arrive at the receiver. Generalize this argument to show that if a number i is in the receiver window, then PKT[$i-6$] or any older data cannot arrive at the receiver.

(b) Show that if the sender window contains 6, then ACK[2] (or earlier) cannot arrive at the sender. Generalize this argument to show that if a number i is in the sender window, then ACK[$i-4$] or any older acknowledgment cannot arrive at the receiver.

(c) Using (a) and (b), argue that it is sufficient to use sequence numbers 0 through 5 for the GBN protocol with $N = 3$.

(d) **Bonus problem:** Generalize your answer for part (c) to arbitrary N. That is, show that it is sufficient to use sequence numbers 0 through $2N-1$ for GBN for arbitrary N. Also give an example that illustrates that the range $[0-4]$ will not suffice for $N = 3$.

4. Sequence number and receive window size in TCP

You are hired to design a reliable byte-stream protocol that uses a sliding window (like TCP). This protocol will run over a 100 Mbps network. The RTT of the network is 100ms, and the maximum segment lifetime is 60 seconds. How many bits would you include in the "Sequence number" and "Receive window" fields of your protocol header?

5. RTT estimation and timeout in TCP

Recall that the Jacobson-Karels algorithm for RTT estimation and timeout calculation is given as:

$$\begin{aligned}
\texttt{DevRTT} &= (1-\beta)\cdot\texttt{DevRTT} + \beta\cdot|\texttt{SampleRTT} - \texttt{EstimatedRTT}|,\\
\texttt{EstimatedRTT} &= (1-\alpha)\cdot\texttt{EstimatedRTT} + \alpha\cdot\texttt{SampleRTT},\\
\texttt{TimeoutInterval} &= \texttt{EstimatedRTT} + 4\cdot\texttt{DevRTT}.
\end{aligned}$$

Suppose that `EstimatedRTT` is 400ms at some point and subsequent `SampleRTTs` are all 100ms. Assume an initial `DevRTT` value of 50ms; use $\alpha = .125$ and $\beta = .25$.

How long does it take before the `TimeoutInterval` value falls below 200ms? If the same measurements and calculations continue *ad infinitum*, what will be the value of `TimeoutInterval` in the limit?

6. TCP congestion control

Parts (a)-(d) and (h) of Problem 27, pages 291-292.