CS 536 (Park) **Assignment XII** **Due: Dec. 1 (Mon.), 1997**

PROBLEM 1

(a) Read Sections 4.2, 5.3, 5.4, 5.5 from P & D.

(b) Give a 2-page synopsis of your project idea. You may consult the course homepage to get some ideas; however, the items listed are not meant to be exhaustive. Your write-up should contain background, problem statement, and your proposed method of attack. It is important to delimit the scale of the project so that it is doable in the remaining time. Think of the project's workload as being comparable to that of a typical weekly problem set, however, now concentrating on a single problem rather than a set.

This write-up is due, separately, on hardcover, on Nov. 24 (Mon.). Put your e-mail addresses on the hand-in. If the write-up is ill-defined or too ambitious, feedback will be provided to that effect.

PROBLEM 2

(a) This is a variation of Problem 2 in Assignment XI using a sender-router-receiver set-up. Rather than invoke the help of the router for congestion control purposes, however, we will implement a purely end-to-end control mechanism. The traffic is still to be routed through the router as before which maintains a buffer of size $B = 30$ (packets).

The goal of the congestion control algorithm is to maximize throughput when sending $X = 1000$ packets (fixed size 1 kB) where *throughput* ν is defined to be

$$\nu = \frac{X - A}{T}$$

where $0 \leq A \leq X$ is the number of dropped packets (you can detect this at the receiver using sequence numbers) and T is the transfer time at the sender of all X packets. That is, T is the difference of the time that the last packet was sent from that of the first packet. Reliability is not an issue. Thus v *may* be maximized if no packets are dropped ($A = 0$) and T is minimal. Why is this not necessarily so? Explain.

The control law which dynamically adjusts the interpacket spacing parameter Λ is now as follows. Let `RTT` be the round-trip time of the most recent successfully received packet. The round-trip time is computed by having the sender put a time stamp of the send time on a packet (in addition to its sequence number) and having the receiver acknowledge the receipt of every k'th ($k \geq 1$) successfully received packet by sending an ACK packet containing the sequence number and sender's time stamp. Upon receive of an ACK packet, the sender takes another time stamp and by taking the time stamps' difference, computes the round-trip time. Let `RTT'` be the newly computed round-trip time. Then,

$$\text{new } \Lambda \leftarrow \text{old } \Lambda + \epsilon \cdot \text{sgn}(\texttt{RTT'} - \texttt{RTT})$$

if $|\texttt{RTT'} - \texttt{RTT}| > \theta$, and new $\Lambda \leftarrow$ old Λ (i.e., Λ stays unchanged) otherwise. Here $\theta > 0$ is a fixed threshold parameter and $\text{sgn}(\cdot)$ is the sign function; i.e., $\text{sgn}(s) = 1$ if $s \geq 0$, and $\text{sgn}(s) = -1$ if $s < 0$. What is this control law trying to do? Explain.

Test your implementation by running your system and measuring throughput ν for $\theta = 1$ ms, $\epsilon = 2$ ms, initial $\Lambda = 25$ ms, and $k = 5$. As part of the control law, enforce 5 ms $\leq \Lambda \leq$ 60 ms. The router (unlike in Assignment XI) services packets at a fixed (deterministic) interval of 30 ms. Give a time trace of Λ, `RTT`, and queue length, again, being careful not to use `printf`. Interpret your results.

Rerun your experiments for $\epsilon = 4, 6, 8, 10$ ms and show ν as a function of ϵ. What do you observe? (You may need to run each experiment 3–4 times to get a usable average performance value.)

Do the same for $\theta = 2, 3, 4, 5$ ms (with $\epsilon = 2$ ms). What do you observe?

What is the throughput when no control law is active (i.e., Λ is unchanged) and Λ is fixed at its initial value of 25 ms? Measure the throughput values for initial Λ of 10, 15, 20, 30, 35, 40 ms and plot ν as a function of initial Λ. In your opinion, how does the dynamic control law compare against the performance of static control?

(b) In an effort to improve the performance of the control law, adjust θ dynamically so that it better reflects the natural fluctuation associated with the `RTT` values. That is, upon computing a new `RTT` value at the sender side, set $\theta \leftarrow 2\sigma$ where σ is the standard deviation of the last 10 observed `RTT` values. If 10 `RTT` values have yet not been collected, set θ to some initial value θ_0. Setting $\theta_0 = 1$ ms, $\epsilon = 2$ ms, initial $\Lambda = 25$ ms, and $k = 5$, how does the performance compare against that of the corresponding case in part (a)? Plot the traces as in part (a).

What if, in addition, the control law is changed to

$$\text{new } \Lambda \leftarrow \text{old } \Lambda + \epsilon \cdot \text{sgn}(\texttt{RTT'} - \overline{\texttt{RTT}})$$

if $|\texttt{RTT'} - \overline{\texttt{RTT}}| > \theta$ where $\overline{\texttt{RTT}}$ is the average of the last 10 `RTT` values? If 10 `RTT` values have yet not been collected, set $\overline{\texttt{RTT}}$ to be the average of the already observed `RTT` values. Run the modified control law for the setting $\theta_0 = 1$ ms, $\epsilon = 2$ ms, initial $\Lambda = 25$ ms, $k = 5$ and compare the performance results and traces against the ones above and in part (a). Give an evaluation of your results.