CS 536 (Park) **Assignment VIII** **Due: Oct. 31 (Fri.), 1997**

PROBLEM 1

Read "Congestion avoidance and control" by Van Jacobson (Proc. ACM SIGCOMM '88, pp. 314–329, 1988). Give a 1-page summary/critique of the paper. Comment on the main pros/cons of the paper from your perspective.

PROBLEM 2

Answer the following questions about TCP by conferring its RFC (Request for Comments), RFC 793. (1) What items are specified for the optional field in the TCP header? How is the MSS encoded? (2) Upon call collision, is a new random sequence number generated or is the previous one used? (3) How is `RST` used? Describe under what conditions it is generated. (4) What does the RFC say about how TCP is to interact with lower layer protocols?

In what RFC are amendments to TCP such as Nagle's algorithm and the Silly Window Syndrome discussed?

(*Note: The RFC's, for example, can be accessed via the URL http://www.isi.edu/rfc-editor/status-catagories.html.*)

PROBLEM 3

(a) This is an extension of Problem 3 in Assignment VII. To induce controlled dropping of packets (emulating a noisy channel), implement a router, call it `my_router`, that accepts UDP packets with the (payload) format

destination-IP-address payload

on port ABC (choose as you wish), examines the destination IP-address, then forwards the payload to the destination host. The router, before forwarding, however, is to drop the current packet with probability p which is a fixed parameter.

The structure of the router consists of a SIGPOLL handler which dequeues packets whenever they arrive, examines the header, then forwarding or dropping the packet.

Repeat the experiments carried out in Problem 3 of Assignment VII, now, with `my_router` running on a third host through which all traffic (data and ACK) is channeled. Use two values for p, 0.2 and 0.1, and compare their performance. Use the runs from Problem 3 in Assignment VII as an approximate sample point corresponding to $p = 0$. Give an interpretation of the results.

(b) As an extension of Problem 3 in Assignment VII, create an application called `ftp-tcp` which has the same command-line arguments as `ftp-greedy`, but instead of engaging in negative `ACK`ing, it just uses TCP with the `write` system call to transfer the file. Compare the performance (i.e., completion time) of `ftp-greedy` against that of `ftp-tcp`. Can you make `ftp-greedy` outperform `ftp-tcp` as you decrease the inter-send time S? Explain your results. Note that the application-layer router `my_router` is not to be used here.

PROBLEM 4

Problems 6.2, 6.8, and 6.11 from P & D.