CS 536 (Park) **Assignment IX** **Due: Nov. 7 (Fri.), 1997**

PROBLEM 1

Read Section 3.3 of "Data Networks" by Bertsekas & Gallager.

PROBLEM 2

(a) Derive the ruin probability q_z of the *Gambler's Ruin* problem filling in all the holes that were skipped over in the derivation carried out in class.

(b) We said in class that the success probability, $1 - q_z$, holds only for $t \to \infty$. Why is this so? For the special case when $p = 1/2$ (symmetric walk), formalize the counting argument described in class (you need to be careful here due to some subtleties). Show that as $t \to \infty$, the probability of never hitting the floor nor the ceiling converges to 0.

PROBLEM 3

(a) List all the assumptions that we made and that we *omitted* when deriving the Poisson process $X(t)$. Be as exhaustive as you can.

(b) Plot (using one of the graph plotting programs, e.g., `gnuplot`) the distribution of the Poisson process, $X_n(t)$, one, as a function of time for $n = 0, 1, 2, 5, 10$ (put all five graphs into a single plot), and two, as a function of n for $t = 0, 5, 10, 15, 100$. Use rate $\lambda = 10$.

If you were working as an analyst at an ISP (Internet Service Provider) and the log showing the number of connections arriving at the ISP server during its busiest period 8pm–11pm was observed to be Poisson with rate $\lambda = 10$, what would your recommendation be regarding the purchase of a new server to replace the old one? Assume the server's spec shows the speed of the server as "X concurrent tasks (assume the tasks supported by the server software is pretty homogeneous) handled with reasonable response time W but $X + 1$ tasks increases it to $100W$". Of course, the higher the spec mark X, the more expensive the server. Justify your recommendation and explain your reasoning, in part, based on the graphs plotted.

What are some of the assumptions that you had to make to advance a meaningful recommendation?

(c) Assume $\xi^1(t), \xi^2(t), \ldots, \xi^m(t)$ are Poisson processes (independent) with rates $\lambda_1, \lambda_2, \ldots, \lambda_m$, respectively. Consider the new process $\xi(t) = \xi^1(t) + \xi^2(t) + \cdots + \xi^m(t)$. Prove that $\xi(t)$ is a Poisson process with rate $\lambda = \lambda_1 + \lambda_2 + \cdots + \lambda_m$. First, clearly state the idea behind your proof. (*Hint: Go back to the set-up we used to develop the Poisson process.*)

Consider the "opposite" case, i.e., given a Poisson process $\xi(t)$ with rate λ, define m new processes $\xi^1(t), \xi^2(t), \ldots, \xi^m(t)$ obtained by splitting the arrivals of $\xi(t)$ into m separate queues with probability $p_1, p_2, \ldots, p_m$, respectively; i.e., $\xi^1(t), \ldots, \xi^m(t)$ represent the queues' lengths. What can you say about the split processes $\xi^1(t), \xi^2(t), \ldots, \xi^m(t)$?

Where might knowing the result of the two operations—merging/splitting—come in handy when talking about computer networks?

PROBLEM 4

(a) Extend the router of Problem 3 in Assignment VIII, `my_router`, such that, one, it engages in buffering of incoming packets into a queue of size B (measured in fixed-size packet units), and two, it services the packets (i.e., dequeues them from the buffer for forwarding) according to a process where the process is determined by specifying the time between the servicing of successive packets, call it Y. Let the inter-packet service time Y be a random variable with distribution $\mathcal{D}_Y$. Notice that both enqueueing of newly arriving packets and dequeueing of departing packets are asynchronous events triggered by `SIGPOLL` and `SIGALRM`, respectively. You need to make sure that the asynchronous events do not get in each other's way affecting their correction functioning.

Test your router by creating sender and receiver applications (similar to Assignment VIII) where the sender's packets are routed through the router to the receiver running on separate hosts. The sender, again, sends suitably sequenced packets with a private header but sender/receiver do not engage in ARQ. That is, communcation is left unreliable (on top of UDP). The sender is characterized by an inter-packet send time S (as before) transmitting fixed-size packets of 512 bytes. Let S be a random variable with distribution $\mathcal{D}_S$.

The sender monitors its send events, logging the sequence number of the packet and its send time stamp (in msec granularity) which is *later*—to avoid the logging induced disk I/O from affecting the timings—written into a file. Similarly for the receiver on receive events. The router engages in logging whenever a new packet arrives (i.e., inside the `SIGPOLL` handler), recording the length of the updated queue and time stamp. Whenever a packet is dropped due to buffer overflow, the sequence number of the dropped packet and its time stamp are logged as well.

Test your system by transmitting 500 packets setting the inter-packet send time fixed at $S = 30$ ms (i.e., deterministic) and inter-packet service time Y fixed at $S = 30$ ms. Set the router buffer capacity B at 30 packets. Plot the logs (in total 4) recorded at the sender, router, and receiver. Create zoomed plots of the above over a time scale showing what transpires for the first 50 packets only. The latter also comes in handy to verify that your system is working correctly.

Repeat the runs with $S = 25, 35$ ms. Show again the zoomed plots. Give an interpretation of your results.

(b) Repeat your experiments, however, now with Y and S stochastic. Take D_Y and D_S as exponential distributions with parameter λ_Y, λ_S, respectively. Noting that an exponential distribution with parameter λ has mean $1/\lambda$, set $1/\lambda_Y = 30$ ms and $1/\lambda_S = 30$ ms. Use the exponential random variate generator posted on the course homepage, `exponential_dist()`. When initializing the sender and router, use `srandom()` with seed 0 to keep the random variates generated across different runs constant.

Repeat the runs with $1/\lambda_S = 20, 25, 35, 40$ ms. Compute the mean queue length, packet loss rate, and their standard deviations from the router log and plot them (separately) as functions of S. Give an interpretation of your results. Which sender/server rate combination would you conclude is "best"? Are there instances in your runs where the receiver is missing packets that are not accounted for by packet drops at the router? If so, then how can this be explained?