PROBLEM 1 (32 pts)

(a) Why is constant bit-rate (CBR) traffic easier to handle than variable bit-rate (VBR) traffic with respect to traffic management when trying to achieve high utilization? In particular, given two traffic streams—CBR and VBR—with identical mean data rates, why is achieving higher utilization with VBR traffic more difficult?

It is said that *perfect* utilization is not achievable in M/M/1 queues. What is the technical underpinning of this? Is this still true if the input process is CBR and the service time deterministic?

(b) What is TCP's *Slow Start* and why is it a misnomer? In what sense is it supposed to be "slow"? TCP's linear increase/exponential decrease congestion control is termed *window*-based control since TCP uses the notion of "window"—as inherited from implementing reliable transmission through ARQ—to throttle the traffic rate sent to a network. How does TCP implement its *linear increase* phase and why is this implementation matter nontrivial and easy to mistake?

(c) Assume you have collected traffic traces at a router by logging into a file the number of packets (for simplicity assume they are of the same size, e.g., cells in ATM networks) flowing through a specific output link every 10ms. Given this trace file (or time series), assume you are asked to check using "rough visual inspection" whether the traffic going through your router is *self-similar* or not. How would you go about doing this? What are some of the perils to be aware of when rendering your judgement—in addition to your eyes being a bit tired after a CS 5xx assignment—assuming traffic measurement spans over a time interval, say, on the order of days or weeks.

(d) Assume you have opened up shop as a network service provider and your specialty is in trans-oceanic fiber optic cables (e.g., connecting Boston and Lisbon) which are point-to-point links which must be shared among your customer base. Your goal is to maximize profit which comes from admitting as many customers as possible while servicing their bandwidth needs. What are the main design considerations coming into play when choosing between a TDM-based link arbitration scheme vs. an asynchronous, i.e., packet-based scheme? Give two scenarios where one is better suited than the other. Does the fact that IP is going to be run as one of the higher layer protocols affect your decision making?

PROBLEM 2 (32 pts)

(a) How does the problem of controlling congestion differ from that of controlling room temperature using a thermostat? Which one is easier to control and in what sense? What is the practical impact of this difference to traffic management?

What influence does time lag or outdatedness of measured information (e.g., measured throughput as reported by the receiver to the sender via feedback) have on control? Is there anything that can be done on the control side to offset the negative impact of time lags?

(b) What is system optimal routing and how does it differ from user optimal routing? How can user optimal routing lead to instabilities (think of a "ping-pong" effect) that might be avoidable with system optimal routing? Given the general superiority (or is it so) of system optimal routing, what are some of the reasons that it is not used in practice, say, on the Internet? What is *multimedia* routing and does it complicate matters for user optimal routing?

(c) It is said that congestion control in high-bandwidth networks has become an even more difficult problem than before due to the increase in delay-bandwidth product. Why is this so? (*Hint: End-to-end paradigm.*) If you were allowed to rearchitect the network to address this specific problem, what would be a possible new approach? What, if any, are the drawbacks of your approach?

(d) Assume you are at a restaurant in Indianapolis and you overhear two people at the next table talking about "IP-over-IP." Having just gotten a C in CS 536 a couple of months back you feel confident and compelled to go over and tell them to stop talking nonsense. On your way over you recognize one of the people as being a class mate of yours who had gotten a $C+$ in the same course. Although your first impulse may still be well justified, you hesitate being thoroughly impressed by your class mate's accomplishment.

Assuming the two people are in mastery of all their senses, first, what would it technically mean to do IP-over-IP—describe what happens to a packet in the protocol stack—and second, what possible use or application might this have? Try to be creative in the second part.

PROBLEM 3 (36 pts)

(a) What is the technical definition of *congestion* when throughput is the performance measure of interest? Under this context, does TCP backoff when there is congestion? Can congestion exist in an M/M/1/n queue? What if we change the performance measure to *reliable throughput*?

If *delay* is made the performance measure, does congestion—in the technical sense—still exist? Discuss the implications of your answer to the control of delay using end-to-end feedback control and issues surrounding stability. Is there a single (scalar) performance measure that captures both throughput and delay simultaneously? Using this single measure, how would you go about performing QoS control? Again, assume a target QoS is set, and the goal is to achieve the target QoS in a stable manner using feedback control.

(b) Assume you are given variable bit-rate (VBR) traffic that is also *real-time*, e.g., compressed video as used in teleconferencing. Assume the real-time VBR traffic is characterized by two numbers, its peak rate λ^* (bps) which is an upper bound on the data rate and its mean rate $\bar{\lambda}$. Assume this traffic stream has to pass through a router which has bandwidth B (bps) but next to zero buffer capacity.

If *stringent QoS* means no packets should be dropped at this router, what bandwidth needs to be reserved? What is the utilization of the router? If occasional packet drops are permissible and packet loss rate is represented by c, then what is the trade-off relationship between QoS (i.e., c) and *burstiness* as represented by the peak-to-mean ratio $\lambda^*/\bar{\lambda}$? How—if at all—do your conclusions change if traffic is *non-real-time*?

(c) *Internet radio* refers to broadcasting radio station programs over best-effort IP Internet. A common trick is to use *buffering* combined with *prefetching* to render continuous, uninterrupted QoS to the end user or listener. That is, upon "tuning and clicking on a station," X seconds worth of audio is downloaded to the end user's machine before actually commencing play at the end user. Thereafter an attempt is made to keep X seconds worth of "future" audio in the user's buffer so that network disturbances can be tolerated without impacting the user's perceived audio quality. Give a careful formulation of the traffic control problem and how you would go about solving the problem of providing uninterrupted QoS. What is the main trade-off relationship that impacts performance? Can this scheme be used to broadcast live talk radio where users are allowed to call in and comment in real-time?

PROBLEM 4 (35 pts = 20 + 15)

(a) Assume you are given a congestion-susceptible network with unimodal load-throughput function $\mu = \mu(Q)$ where $\mu(Q)$ achieves maximum at the point (Q^*, μ^*). Let the system be described by the following two differential equations, representing the conservation of flow and a congestion control law, respectively:

$$dQ/dt = \lambda - \mu, \tag{0.1}$$

$$d\lambda/dt = \epsilon(\hat{Q} - Q). \tag{0.2}$$

Assume your target operating point is $(\hat{Q}, \hat{\mu})$ where $\hat{\mu} = \mu(\hat{Q})$, i.e., $\hat{\mu}$ corresponds to the throughput achieved when the load is $\hat{Q}$, and $\hat{Q} < Q^*$. Restricting yourself to the first quadrant $\{ (Q, \lambda) : Q \geq 0, \lambda \geq 0 \}$ of the (Q, λ) plane—note that μ is *determined* by Q so the two independent variables of interest are load Q and data rate λ—draw the phase portrait or vector field around the point $(Q, \lambda) = (\hat{Q}, \hat{\mu})$. Explain how to achieve this drawing. Is $(\hat{Q}, \hat{\mu})$ a fixed point? If so, is it stable? Explain.

In an actual end-to-end feedback congestion control system where the receiver is allowed to send feedback information to the sender for possible use in congestion control, given that Q and $\hat{Q}$ are difficult to observe quantities, but the *target* or *desired* throughput $\hat{\mu}$ may be specified by the receiver, how would you change (0.2) such that the target operating point is achieved in a feasible manner?

(b) The *Prisoner's Dilemma* problem states that if two prisoners cooperate with each other (i.e., do not betray each other) upon interrogation then both go free (the best of possible worlds, at least with respect to the prisoners' viewpoint), if one betrays and the other doesn't then the one who betrays gets a 1-year jail term whereas the one who doesn't gets a 10-year term, and if both betray each other then each gets a 5-year term. In a noncooperative world where each user (including prisoners) acts selfishly—chooses actions to optimize one's personal gain only—the prisoners would each choose to betray each other and serve 5-year terms.

Relate TCP's congestion control actions—upon noticing the possible on-set of congestion—to the Prisoner's Dilemma problem by drawing analogies. Assume there are two connections sharing common network resources. What would a noncooperative version of TCP look like in terms of its functioning? Under what conditions would your greedy TCP function optimally? Phrase your answer in terms of the bandwidth shared by two connections. Is there an "optimal" noncooperative TCP, and if so, what would it look like? How does this differ from the noncooperative Prisoner's Dilemma set-up?