Problem Set 4

ECE567 Fall 2009

Each problem is worth 10 points. Due: Oct. 20

Do any four of the six problems.

1. Consider an $N \times N$ VoQ switch where the arrivals into queue (i, j) are Bernoulli with mean λ_{ij}. Assume that

$$\sum_i \lambda_{ik} < 1 \quad and \quad \sum_j \lambda_{lj} < 1, \quad \forall k, l.$$

Compute an upper bound on the steady-state mean of the sum of the queue lengths in the network assuming that the max-weight scheduling algorithm discussed in class is used. Try to get as tight a bound on the upper bound as possible. *Hint: Proceed as in the Lyapunov stability proof, but assume that the system is already in steady state.*

2. In class, we showed that the max-weight algorithm which maximizes the sum of the product of the service rates and the queue lengths, subject to scheduling constraints, achieves 100% throughput in a high-speed switch. Instead, consider the following algorithm where the schedule $\{I_{ij}\}$ at time slot k is chosen to maximize

$$\sum_{ij} I_{ij} q_{ij}^2(k)$$

subject to scheduling constraints. In words, we are now choosing the link weights to be the square of the queue lengths. Show that this algorithm achieves 100% throughput. *Hint: Consider the Lyapunov function*

$$\sum_{i,j} q_{ij}^3.$$

3. While the maxweight matching algorithm for bipartite is a polynomial-time algorithm, it is still too complex to implement in a high-speed switch. An alternative is the maximal matching algorithm which works as follows: consider only those links whose queue lengths are greater than zero. Find a maximal matching among these links, i.e., a matching such that no more links can be added to the schedule. Show that the maximal algorithm stabilizes the queues under the following condition:

$$\sum_j \lambda_{ij} < 1/2 \quad \forall i \quad \text{ and } \quad \sum_i \lambda_{ij} < 1/2 \quad \forall j.$$

Hint:

(i) If I_{ij} is an indicator function indicating whether link (i, j) is included in the matching, then the maximal matching algorithm has the following property: if $q_{ij} > 0$, then $\sum_j I_{ij} + \sum_i I_{ij} \geq 1$.

(ii) Define $E_{ij} = \{(l, k) : l = i \text{ or } k = j\}$, i.e., it is the set of links that are neighbors of (i, j) including (i, j). Use the Lyapunov function

$$V(q) = \sum_{i,j} q_{ij} \left(\sum_{(j,k) \in E_{ij}} q_{jk} \right).$$

(iii) When using the above Lyapunov function, it may be useful to first pretend that the queue dynamics are given by deterministic differential equations and get a feel for the drift by differentiation. Then, use Foster's criterion to complete the proof rigorously.

4. Consider a cellular wireless network consisting of a base station and two receivers, call them A and B. The channel from the base station to receiver A can be in one of three states: 0, 1, and 3 with equal probability, where the index of the state is the number of packets that can be transmitted in that channel state. Assume that the channel to receiver B is identical to and independent of the channel state to receiver A. Further, assume that the base station can transmit to only one receiver in each time slot. Draw the capacity region of this wireless network.

5. For the cellular network model in the previous problem, suppose that packet arrivals occur according to two independent Bernoulli processes, with the mean arrival rate of packets intended for receiver i equal to λ_i, where (λ_1, λ_2) lies within the capacity region. Compute an upper bound on the sum of mean queue lengths assuming that the maxweight algorithm is used.

6. In ad hoc wireless network, recall that the throughput-optimal scheduling algorithm requires the network to solve a maxweight independent set problem with queue lengths as weights. In this exercise, we will show that an approximation to the maxweight independent set scheduling algorithm is also throughput optimal. Towards this end, suppose that the scheduling algorithm is a randomized algorithm which picks a schedule in each time slot according to some probabilistic rule. Further, suppose that the probabilistic rule satisfies the following property: given any $\delta > 0$ and $\epsilon > 0$, there exists a finite set $B_{\delta,\epsilon}$ such that, with probability greater than or equal to $(1 - \delta)$, the randomized algorithm produces a schedule whose weight is greater than or equal to $(1 - \epsilon)$ times the weight of the maxweight schedule whenever the vector of queue lengths lie in B^c. Show that such a randomized algorithm is also throughput-optimal, i.e., it stabilizes all arrival rates within the capacity region.