Stanford University
Department of Electrical Engineering

EE384Y, Spring 2006 | Date Issued: Monday, April 17
Problem Set No 1 | Due: Friday, April 28 by 4pm

Please put your homework in the file bins on third floor of Gates building. No late homework is accepted unless formerly approved by a Professor.

---

1. Stability with maximum weight matching

In class we saw that a switch with VOQs and Bernoulli arrivals is stable if maximum weight matching is used, and if the weights are equal to occupancies of the VOQs. The algorithm is often called LQF (Longest Queue First) [even though it may not actually serve the longest queue!].
A disadvantage of LQF is that a non-empty VOQ may remain un-serviced indefinitely.

(a) Illustrate and describe a sequence of arrivals in a 2×2 switch in which a VOQ can remain unserved indefinitely.

A possible solution to this problem is to make the weight on each edge equal to the waiting time, $W_{ij}(n)$, of the cell at HOL of VOQ (i, j). At time n, $W_{ij}(n)$ is defined to be the time since the HOL cell arrived at the switch. The algorithm is now called OCF (Oldest Cell First).
It is worth asking if OCF is stable. It turns out that it is (under the usual conditions), but it is too complex to prove here (feel free to prove it if you'd like!).
Let's look at some properties of the waiting times.

- If VOQ (i, j) is unserved at time n, then $W_{ij}(n + 1) = W_{ij}(n) + 1$.
- If VOQ (i, j) is served at time n, and there are no more cells in the queue, we define $W_{ij}(n + 1) = 0$.

(b) Consider the HOL cell in VOQ (i, j) at time n. The inter-arrival time to the system between it, and the next cell to arrive to the same queue, is denoted $T_{ij}(n)$. If VOQ (i, j) is served at time n, and there is a new cell HOL cell behind it, write down an expression for $W_{ij}(n + 1)$.

(c) Write down an expression for vector:

$$\underline{W}(n + 1) = [W_{11}(n + 1), \ldots, W_{NN}(n + 1)]^T$$

in terms of $\underline{W}(n)$, $\underline{T}(n) = [T_{11}(n), \ldots, T_{NN}(n)]^T$ and service vector $\underline{S}(n) = [S_{11}(n), \ldots, S_{NN}(n)]^T$ which defines which queues are served at time n.

(d) If $E[\|\underline{W}(n)\|] < \infty$, can we conclude that $E[\|\underline{L}(n)\|] < \infty$ also (where $\underline{L}(n)$ is the vector of queue occupancies $L_{ij}(n)$)? Explain.

2. Linear Combination of Maximum Weight Matchings

In this problem we shall consider two maximum weight matching algorithms. Let $L_{ij}(n)$ denote the queue size of the VOQ_{ij} at time n. Similarly, let $W_{ij}(n)$ be the waiting time of the oldest cell in VOQ_{ij} at time n. Let, $w_{ij}(n)$ be the weight of VOQ_{ij} at time n.
In class we considered the following two *stable* maximum weight matching algorithms.

(a) Longest Queue First (LQF), which uses a maximum weight matching, where the weight $w_{ij}(n) = L_{ij}(n)$.

(b) Oldest Cell First (OCF), which uses a maximum weight matching, where the weight $w_{ij}(n) = W_{ij}(n)$.
We know that LQF is not fair in the sense that it can starve certain queues for specific arrival patterns. Though, OCF does not suffer from starvation, it can possibly serve shorter queues and hence might suffer from larger delays. You and Estella, are both network design engineers at a hot bay area startup. You decide to come up with the following clever algorithm which you believe combines the best features of both algorithms.

(c) Smart Cells First (SCF), which uses a maximum weight matching, where the weight $w_{ij}(n) = 0.5L_{ij}(n) + 0.5W_{ij}(n)$.
To convince your customers, prove the stability of the SCF algorithm for all admissible Bernoulli IID traffic.

3. Pipelining Stability

This question is about the stability of a switch that uses the maximum weight matching algorithm, with weights equal to the queue occupancy; i.e. $w_{ij}(n) = Q_{ij}(n)$. We already know that such a switch is stable for all admissible and independent arrivals.
In practice, it is common to build a switch scheduler that is pipelined. Pipelining is easier if the switch scheduler can make its decisions based on slightly out of date information. In this question, we will explore whether such a switch is stable.
We will define the out-of-date occupancy at time n,

$$\hat{Q}_{ij}(n) = Q_{ij}(n-k),$$

where k is the number of time slots that the occupancy information is out of date.

(a) Prove that

$$Q_{ij}(n) - k \leq \hat{Q}_{ij}(n) \leq Q_{ij}(n) + k. \tag{1}$$

At time n, the switch scheduler picks the maximum weight matching based on $\hat{Q}(n)$; i.e. it finds

$$\hat{S}(n) = \arg\max_{S(n)} \left(\hat{Q}^T(n) S(n) \right).$$

(b) Argue that

$$\hat{Q}^T(n)\hat{S}(n) \geq \hat{Q}^T(n) S^*(n), \tag{2}$$

where

$$S^*(n) = \arg\max_{S(n)} \left(Q^T(n) S(n) \right).$$

(c) We can deduce from the lectures that for $L(n) = Q^T(n)Q(n)$,

$$E[L(n+1) - L(n) | L(n)] \leq 2E[Q^T(n)(\lambda - \hat{S}^{(}n))] + 2N + N^2 \tag{3}$$

Show that

$$E[L(n+1) - L(n) | L(n)] \leq f(k, N), \tag{4}$$

where f is a function of k and N. Provide f.

(d) If we increase k, can we expect delay to increase, decrease or stay the same? Explain your answer.