Analysis of scheduling algorithms that provide 100% throughput in input-queued switches

Isaac Keslassy, Nick McKeown
Computer Systems Laboratory, Stanford University
Stanford, CA 94305-9030
{keslassy, nickm}@stanford.edu

***Abstract* -- Internet routers frequently use a crossbar switch to interconnect linecards. The crossbar switch is scheduled using an algorithm that picks a new crossbar configuration every cycle. Several scheduling algorithms have been shown to guarantee 100% throughput under a variety of traffic patterns. The first such algorithm was the maximum weight matching (MWM) algorithm in which the weight is the sum of the occupancies of the queues. We explore whether alternative weight functions, such as using the sum of the square of the occupancies, leads to stronger or weaker stability. The first result of this paper is that a broad class of weight functions give rise to strong stability, including the sum of the squares, the sum of the cubes and so on. A counter-intuitive result, indicating a limitation of the Lyapunov method, is that the sum of the square root of the occupancies is not included in this class, even though simulation suggests that the resulting average delay is lower than for the other functions. We also consider the simpler, $O(\log N)$, randomized scheduling algorithm (TASS) proposed by Tassiulas. We show similar results for different weight functions as for MWM. We finally show that TASS gives 100% throughput when the weights are noisy, or out-of-date.**

I. INTRODUCTION

It is common for high performance packet switches (e.g. Internet routers, ATM switches and Ethernet switches) to use a crossbar switching fabric and input queues to hold packets during times of congestion. Karol et al. [1] showed that input queued switches can suffer from reduced throughput due to head of line blocking. And so it is now common for input queued packet switches to maintain virtual output queues (VOQs) [2]. Such switches, when combined with a suitable scheduling algorithm, have been shown to achieve 100% throughput [3][4] for traffic that is uniformly or non-uniformly distributed over the outputs of the switch. The ability of a switch to achieve 100% throughput is desirable to a network operator, as it assures that all of the (expensive) link capacity can be utilized.

A scheduling algorithm known to provide 100% throughput is maximum weight matching (MWM) [4]. In cell time n, the scheduler calculates weight w by summing the occupancies of all the virtual output queues. It then finds the switch configuration that maximizes w. In [4] it is shown that this algorithm leads to strong stability: i.e. $E[X_{ij}(n)] < \infty$ for all *i,j*, where $X_{ij}(n)$ is the occupancy at time *n* of the VOQ at input *i* that holds cells destined for output *j*.

It is worth asking if calculating weights as the sum of the VOQ occupancies is the best, or only, way to achieve strong stability. For example, what if the weight was the sum of the

This work was supported by a Wakerly Stanford Graduate Fellowship and by the Powell Foundation.

square of the occupancies, i.e. $w = \sum_{i,j} X_{ij}^2(n)$; or perhaps $w = \sum_{i,j} \sqrt{X_{ij}}(n)$. Will this lead to strong stability, and if so, what weight function should we choose? We will explore this question in the first part of the paper and show that a broad class of weight functions lead to strong stability, while some others lead to weaker forms of stability, and yet others lead to known instability.

Unfortunately, the scheduling algorithms that achieve 100% throughput using a maximum weight matching are not implementable in hardware at high speed, and so are not used in practice. On the other hand, implementable and widely used hardware algorithms such as WFA [6], PIM [7] and *i*Slip [8], can not guarantee 100% throughput. And so Tassiulas recently proposed a new randomized scheduling algorithm (TASS, [9]). Using memory, TASS exploits the correlation between successive cell times to approximate MWM. Let the weight $w(M)$ of any match M be the sum of the lengths of the virtual output queues it services, and let $TASS(n)$ and $MWM(n)$ be the matches that TASS and MWM choose at time step *n*. Then TASS works as follows. At the end of each time step $n-1$, the scheduler keeps in memory $TASS(n-1)$. At the following time step n, it computes a new match $C(n)$, and compares $w(C(n))$ with $w(TASS(n-1))$, keeping the match that has the biggest weight. Thus,

$$\begin{cases} TASS(n) = C(n) \text{ if } w(C(n)) \geq w(TASS(n-1)) \\ TASS(n) = TASS(n-1) \text{ otherwise.} \end{cases} . \quad (1)$$

In the case of Bernoulli i.i.d. traffic, if $Pr(C(n) = MWM(n)) > \varepsilon$ for all n with $\varepsilon > 0$, then Tassiulas proves that TASS achieves 100% throughput. Therefore, when the above conditions are satisfied, it is possible to obtain 100% throughput using a much simpler scheduling algorithm. Indeed, the only complexity added to the computation of $C(n)$ lies in the computation and comparison of the weights. This represents an $O(\log N)$ additional complexity, where *N* is the number of ports (neglecting for the moment the role of weights in the complexity). Since choosing $C(n)$ randomly among all possible matches satisfies the above conditions, one can therefore prove strong stability in $O(\log N)$ steps – this is to be compared with $O(N^{2.5}\log N)$ for a sequential computation of MWM (Gabow and Tarjan [10]), and $O(N^{2/3}(\log N)^4)$ for a parallel computation of MWM with a polynomial number of processors (Goldberg et al., [11]).

We are interested in several aspects of the TASS algorithm:

1. TASS attempts to track MWM. Is the difference in the weight of the match bounded in each step? We show that the bound is finite at each step.
2. What if, like we did above for MWM, we calculated the weight differently? For example, with some function of the VOQ occupancies. We show that a broad class of weight functions provide 100% throughput.
3. Finally, what if the weights are noisy, due perhaps to rounding errors when calculating weights, or if a small number of bits is used to represent the weight? Or perhaps the scheduler is pipelined and is using weight information that is out of date when the calculation is performed? We show that when the weights are noisy or out of date within a finite bound, then 100% throughput is still guaranteed.

II. Choosing The Weight In The MWM Algorithm

A. Notations

Several notations will be used in this article. N is the number of ports of the packet switch. X is the $N \times N$ matrix representing the lengths of the VOQs. X_{ij} is the number of cells going from input i to output j, with $1 \le i, j \le N$.

A match M is a permutation matrix of size $N \times N$. The weight of a match M with respect to X is $w(M) = w_X(M) = \langle X, M \rangle = \sum_{i,j} X_{ij} M_{ij}$.

For every time slot n, $A(n)$ and $D(n)$ are respectively the arrival and departure matrices. Thus, $X(n+1) = X(n) + A(n) - D(n)$ with $X(0) = 0$. The arrival rate matrix is noted λ. It is said to be admissible if $\sum_i \lambda_{ij} < 1$ and $\sum_j \lambda_{ij} < 1$ for all i,j.

B. The weight dilemma

In this section, we will focus on the schemes, called $MWM - f(X)$, in which the weight of a VOQ depends only on its queue length: $w(i,j) = f(X(i,j))$. For instance, the default MWM in [4] is $MWM - X$, Maximum Size Matching (MSM) is $MWM - \delta(X > 0)$, and MWM with the squares of the queue lengths is $MWM - X^2$.

Intuitively, in order to maximize the throughput, one should maximize the instantaneous throughput, and therefore use MSM. However, this intuition proves wrong, and it is known that MSM does not achieve 100% throughput while $MWM - X$ does [4]. This difference between both schemes comes from the fact that MSM does not take into account queue lengths. Therefore, one could wonder: since packet switch stability increases when the heaviest queues are advantaged, shouldn't the heaviest queues be given an even greater weight? What if we use $MWM - X^2$, or $MWM - X^3$ instead of $MWM - X$?

C. The Weight Choice

One way to choose the weight function is to obtain stability results for each weight function, and see which one gives the strongest stability results. The following theorem, valid within a large class of functions $f(.)$, shows that the Lyapunov method used in [4] suggests stronger stability with increasing powers of the occupancy.

Theorem 1 *Assume that $f(.)$ is nonnegative and continuously differentiable on $\Re^+$, and that $\lim_{x \to \infty} \left\{ \frac{\max_{[x-1, x+1]} |f'(u)|}{f(x)} \right\} = 0$. Then, for any Bernoulli iid admissible traffic, and for all i,j, $MWM - f(X)$ satisfies:*

$$E[f(X_{ij}(n))] < \infty \tag{2}$$

Proof: The proof is similar to the proof that MWM-X has 100% throughput. Let ε be such that $\sum_i \lambda_{ij} \le 1 - \varepsilon$ and $\sum_j \lambda_{ij} \le 1 - \varepsilon$ for any *i,j*. Define: $\Delta(n) \equiv A(n) - D(n)$,

$MWM \equiv MWM - f(Q)$, $F(x) \equiv \int_o^x f(u)du$, $F(X(n)) \equiv \sum_{i,j} F(X_{ij}(n))$, and

$DRIFT \equiv E[F(X(n+1)) - F(X(n))|X(n)]$. Then:

$$\begin{aligned} DRIFT &= E[F(X(n) + \Delta(n)) - F(X(n))|X(n)] \\ &= \sum_{i,j} E[F(X_{ij}(n) + \Delta_{ij}(n)) - F(X_{ij}(n))\big|X(n)] \\ &= \sum_{i,j} E\left[\int_0^{\Delta_{ij}(n)} f(X_{ij}(n) + u)du \,\middle|\, X(n)\right]. \end{aligned}$$

Using the integral version of Taylor formula:

$$\begin{aligned} DRIFT &= \sum_{i,j} E\left[\Delta_{ij}(n) \cdot f(X_{ij}(n)) + \int_0^{\Delta_{ij}(n)} (\Delta_{ij}(n) - u) \cdot f'(X_{ij}(n) + u)du \,\middle|\, X(n)\right] \\ &\le E[\langle f(X(n)), \Delta(n)\rangle |X(n)] + \sum_{i,j} E\left[\frac{\Delta_{ij}^2(n)}{2}\left\{\max_{[X_{ij}(n)-1,\, X_{ij}(n)+1]} |f'(u)|\right\}\middle| X(n)\right] \end{aligned}$$

In particular,

$$\begin{aligned} E[\langle f(X(n)), \Delta(n)\rangle |X(n)] &= E[\langle f(X(n)), A(n) - D(n)\rangle |X(n)] \\ &= E[\langle f(X(n)), A(n) - MWM(n)\rangle |X(n)] \\ &= \langle f(X(n)), \lambda - (1-\varepsilon) \cdot MWM(n)\rangle - \varepsilon\langle f(X(n)), MWM(n)\rangle \\ &\le 0 - \varepsilon \cdot \max_{i,j} |f(X_{ij}(n))| \end{aligned}$$

(using Birkhoff's theorem [12]). Hence, since $-1 \le \Delta_{ij}(n) \le 1$,

$$DRIFT \le -\varepsilon \cdot \max_{i,j} |f(X_{ij}(n))| + \frac{N^2}{2} \cdot \max_{i,j}\left\{\max_{[X_{ij}(n)-1,\, X_{ij}(n)+1]} |f'(u)|\right\}$$

Let $\varepsilon' > 0$. Then, since f' is continuous, there exists K such that for all $x \ge 0$, $\max_{[x-1,\, x+1]} |f'(u)| < \frac{\varepsilon\varepsilon'}{N^2} \cdot |f(x)| + K$. Thus,

$$DRIFT \le -\varepsilon \cdot \left(1 - \frac{\varepsilon'}{2}\right) \cdot \max_{i,j} |f(X_{ij}(n))| + \frac{K \cdot N^2}{2}.$$

Since ε' is arbitrary, it is possible to assign $\varepsilon' = 1$. Using the fact that $\frac{K \cdot N^2}{2}$ is a constant and applying the Foster-Lyapunov theorem ([4][9]) leads to: $E\left\{\max_{i,j} |f(X_{ij}(n))|\right\} < \infty$, hence $E[f(X_{ij}(n))] < \infty$ for all *i,j*.

❐

Corollary 1 *With any Bernoulli iid admissible traffic, and for all i,j:*

1. For $MWM - X$, we have $E[X_{ij}(n)] < \infty$.

2. For $MWM - X^2$, we have $E[X_{ij}^2(n)] < \infty$, and therefore $E[X_{ij}(n)] < \infty$.

3. For $MWM - \sqrt{X}$, *we have* $E[\sqrt{X_{ij}(n)}] < \infty$.

Proof: (1) is obvious, and was already proved in [1].

(2) follows from Cauchy-Schwarz theorem.

(3) is obvious, using an infinitely differentiable function equal to $f(.)$ on $\{0\}$ and $[1, \infty)$ (for instance with polynomial interpolation).

❒

Corollary 2 *Assume that* a *and* b *are two positive constants. With any Bernoulli i.i.d. admissible traffic, and any scheme* $MWM - f(X)$ *satisfying the conditions of the above theorem, if* $f(x) \geq ax$ *for all* $x \geq b$, *then the packet switch is strongly stable, i.e. for all i,j,* $E[X_{ij}(n)] < \infty$.

Proof: For all $x \geq 0$, $f(x) \geq a(x-b)$, thus $E[a(X_{ij}(n) - b)] < \infty$ which implies $E[X_{ij}(n)] < \infty$.

❒

Corollary 3 *With any Bernoulli i.i.d. admissible traffic,* $MWM - X$ *and* $MWM - X^2$ *are strongly stable.*

D. Simulations

From the theorems above, we could easily believe that giving a greater importance to long queues brings stronger stability, and perhaps lowers delays. However, surprisingly, our simulations indicate that this assumption is wrong. As shown in Figure 1, $MWM - \sqrt{X}$ had a lower average delay than $MWM - X$, which had a lower average delay than $MWM - X^2$.,

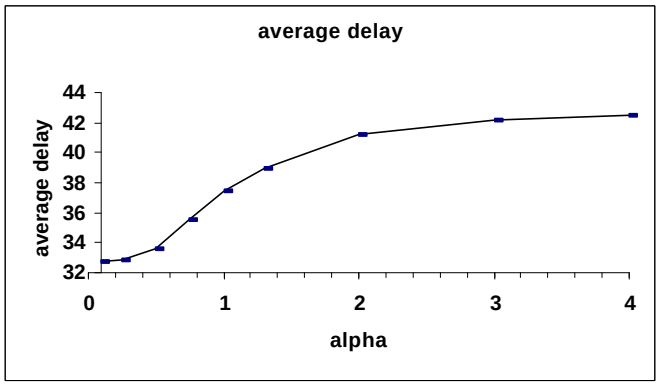

Figure 1: Average delay for $MWM - X^{\alpha}$.

This simulation is for the 3-port switch configuration described in [4], for which MSM is unstable[1]. Nevertheless, these results are representative of the results obtained as well with most of the other traffic patterns. Thus, this could mean that strong stability and low average delays are not necessarily linked ; this could also mean that the Lyapunov techniques used above do not necessarily reflect everything as far as stability is concerned.

III. Does TASS Track MWM?

The objective of the TASS scheduling algorithm is to approximate MWM, by successively

[1.] The traffic was Bernoulli i.i.d., and the simulation run for 500,000 time slots.

guessing new matches and keeping the heaviest ones. Does it succeed in tracking MWM? If it is possible to show that TASS does succeed in approximating MWM, then analyzing MWM will be helpful for analyzing TASS. The following theorem shows that TASS does indeed track MWM within a finite bound.

Theorem 2 *Let* $\delta(n) \equiv E[w(MWM(n)) - w(TASS(n))]$. *Then:*

$$\delta(n) < \infty \tag{3}$$

Proof: Let $d(n) \equiv w(MWM(n)) - w(TASS(n))$ and let $Y(n)$ be the pair $(X(n), d(n))$. We use the following properties:

1. $Pr(C(n+1) = MWM(n+1)) > \varepsilon$,
2. For any match M, $\langle X(n), M \rangle \le \langle X(n), MWM(n) \rangle$,
3. $\langle X(n+1), TASS(n+1) \rangle \ge \langle X(n+1), TASS(n) \rangle$.

Then:

$$\begin{aligned}
E[d(n+1)|X(n)] &= E[\langle X(n+1), MWM(n+1) - TASS(n+1) \rangle | Y(n)] \\
&\le (1-\varepsilon) E[\langle X(n+1), MWM(n+1) - TASS(n+1) \rangle | Y(n), C(n+1) \ne MWM(n+1)] \\
&\le (1-\varepsilon) E[\langle X(n+1), MWM(n+1) - TASS(n) \rangle | Y(n), C(n+1) \ne MWM(n+1)] \\
&\le (1-\varepsilon) E[\langle X(n), MWM(n+1) - TASS(n) \rangle | Y(n), C(n+1) \ne MWM(n+1)] \\
&\quad + (1-\varepsilon) E[\langle A(n) - D(n), MWM(n+1) - TASS(n) \rangle | Y(n), C(n+1) \ne MWM(n+1)]. \\
&\le (1-\varepsilon) E[\langle X(n), MWM(n+1) - TASS(n) \rangle | Y(n), C(n+1) \ne MWM(n+1)] \\
&\quad + (1-\varepsilon) \cdot 2N \\
&\le (1-\varepsilon) \cdot (d(n) + 2N). \text{ Hence:}
\end{aligned}$$

$$\begin{aligned}
\delta(n+1) &= E[d(n+1)] \\
&= E\{E[d(n+1)|Y(n)]\} \\
&\le E\{(1-\varepsilon) \cdot (d(n) + 2N)\} \\
&\le (1-\varepsilon) \cdot (\delta(n) + 2N)
\end{aligned}$$

Thus, $\delta(0) = 0$ and $\delta(n+1) \le (1-\varepsilon) \cdot (\delta(n) + 2N)$ for $n \ge 0$. $\delta(n)$ is bounded by a converging geometric series. Let $u(n) = \frac{2N \cdot (1-\varepsilon)}{\varepsilon} - \delta(n)$. Then $u(0) = \frac{2N \cdot (1-\varepsilon)}{\varepsilon}$ and $u(n+1) \ge (1-\varepsilon) \cdot u(n)$ for $n \ge 0$. Hence $0 \le (1-\varepsilon)^n \le u(n)$ and $\delta(n) \le \frac{2N \cdot (1-\varepsilon)}{\varepsilon}$.

❒

Hence the expected difference between the weights of TASS and MWM is bounded. This means that TASS does indeed track closely MWM.

IV. Choosing The Weight In The TASS Algorithm

We can now ask the same question about TASS that we asked about MWM: is it best to use the sum of the VOQ occupancies, or would some other function lead to stronger stability or lower delay?

A. Theory

As with $MWM - f(X)$, let's define $TASS - f(X)$ as TASS with weight function $f(.)$. With several strong properties established for $MWM - f(X)$, we verify that they extend to $TASS - f(X)$, which is an approximation of $MWM - f(X)$. It is quite surprising that for the same broad conditions, it is indeed possible to get the same strong stability properties for

$TASS-f(X)$.

Theorem 3 *Assume that* $f(.)$ *is nonnegative and continuously differentiable on* $\Re^+$, *and that* $\lim_{x\to\infty}\left\{\frac{\max_{[x-1,x+1]}|f'(u)|}{f(x)}\right\} = 0$. *Then, for any Bernoulli iid admissible traffic, and for all i,j,* $TASS-f(X)$ *satisfies:*

$$E[f(X_{ij}(n))] < \infty \tag{4}$$

Proof: The proof is similar to that for $MWM-f(X)$, and we will use the same notation below, including $MWM \equiv MWM-f(X)$ and $TASS \equiv TASS-f(X)$.
Define $d(n) \equiv w(MWM(n)) - w(TASS(n)) = \langle f(X(n)), MWM(n) - TASS(n)\rangle$.
Define $Y(n)$ to be the pair $(X(n), d(n))$, $V(Y(n)) \equiv \frac{d(n)}{\varepsilon} + F(X(n))$, and $DRIFT = E[V(Y(n+1)) - V(Y(n))|X(n)]$.
Since V has two distinct members, we'll bound each one of them.
The first member of V is $\frac{d(n)}{\varepsilon}$. As in Theorem 1,

$$\begin{aligned}
&E[d(n+1)|Y(n)] \\
&\le (1-\varepsilon)E[\langle f(X(n+1)), MWM(n+1) - TASS(n+1)\rangle | Y(n), C(n+1) \ne MWM(n+1)] \\
&\le (1-\varepsilon)E[\langle f(X(n+1)), MWM(n+1) - TASS(n)\rangle | Y(n), C(n+1) \ne MWM(n+1)] \\
&\le (1-\varepsilon)d(n) + (1-\varepsilon)\sum_{i,j} E\left[\left\langle \int_0^{\Delta_{ij}(n)} f'(X_{ij}(n)+u)du, MWM(n+1) - TASS(n)\right\rangle \middle| Y(n)\right]
\end{aligned}$$

From Theorem 2 we know,
$E[d(n+1)|Y(n)] \le (1-\varepsilon)d(n) + (1-\varepsilon)(\varepsilon\cdot\varepsilon'\cdot\max_{i,j}|f(X_{ij}(n))| + K\cdot N^2)$
The second member of V is $F(X(n))$. From Theorem 2 we know,

$$\begin{aligned}
&E[\langle f(X(n)), \Delta(n)\rangle | X(n)] \\
&= E[\langle f(X(n)), A(n) - MWM(n) + MWM(n) - TASS(n)\rangle | Y(n)] \\
&\le \langle f(X(n)), \lambda - (1-\varepsilon)\cdot MWM(n)\rangle - \varepsilon\langle f(X(n)), MWM(n)\rangle + d(n) \\
&\le -\varepsilon\cdot\max_{i,j}|f(X_{ij}(n))| + d(n)
\end{aligned}$$

Therefore, using the integral inequality from Theorem 2,

$$\begin{aligned}
&E[F(X(n+1)) - F(X(n))|X(n)] \\
&\le -\varepsilon\cdot\max_{i,j}|f(X_{ij}(n))| + d(n) + \frac{N^2}{2}\cdot\max_{i,j}\left\{\max_{[X_{ij}(n)-1, X_{ij}(n)+1]}|f'(u)|\right\} \\
&\le -\varepsilon\left(1-\frac{\varepsilon'}{2}\right)\cdot\max_{i,j}|f(X_{ij}(n))| + d(n) + \frac{KN^2}{2}.
\end{aligned}$$

Summing the inequalities from both members of V:

$$DRIFT \le -\varepsilon\left(1-\frac{\varepsilon'}{2}\right)\cdot\max_{i,j}|f(X_{ij}(n))| + \frac{KN^2}{2} + \frac{(1-\varepsilon)}{\varepsilon}\cdot\left\{\varepsilon\varepsilon'\cdot\max_{i,j}|f(X_{ij}(n))| + 2KN^2\right\}$$

$$DRIFT \le \left\{ \max_{i,j} \left| f(X_{ij}(n)) \right| \right\} \cdot \left\{ -\varepsilon + \varepsilon' \cdot \left(1 - \frac{\varepsilon}{2} \right) \right\} + KN^2 \cdot \{ 1 + 2\varepsilon' \cdot (1 - \varepsilon) \}$$

Hence, choosing ε' small enough finally brings $E\left\{ \max_{i,j} \left| f(X_{ij}(n)) \right| \right\} < \infty$.

❒

The following corollaries are straightforward.

Corollary 4 *With any Bernoulli i.i.d. admissible traffic, and for all i,j:*
1. For $TASS - X$, *we have* $E[X_{ij}(n)] < \infty$.
2. For $TASS - X^2$, *we have* $E[X_{ij}^2(n)] < \infty$, *which implies that* $E[X_{ij}(n)] < \infty$.
3. For $TASS - \sqrt{X}$, *then* $E[\sqrt{X_{ij}(n)}] < \infty$.

In particular, this corollary proves that using $TASS - X^2$, it is possible to achieve both a fixed mean and a fixed variance of any VOQ length. Indeed, more generally, it is possible to get a bounded n^{th} moment of the queue sizes with $TASS - X^n$ for any $n \ge 1$.

Corollary 5 *Assume that* a *and* b *are two positive constants. With any Bernoulli i.i.d. admissible traffic, and any scheme* $TASS - f(X)$ *satisfying the conditions of the above theorem, if* $f(x) \ge ax$ *for all* $x \ge b$, *then the packet switch is strongly stable.*

Corollary 6 *With any Bernoulli i.i.d. admissible traffic,* $TASS - X$ *and* $TASS - X^2$ *are strongly stable.*

B. Simulations

Our simulations of TASS used the same configuration as those of MWM, and are shown in Figure 2. The $C(n)$ that was chosen was generated by *i*Slip, in order to be closer to current real algorithms. Interestingly, the results are the same as those for MWM, and inverse to what could be expected from the above theorems: $TASS - iSlip - \sqrt{X}$ has the best delays, followed by $TASS - iSlip - X$, $TASS - iSlip - X^2$, and finally $TASS - iSlip - \delta(X > 0)$ (not represented here because unstable).

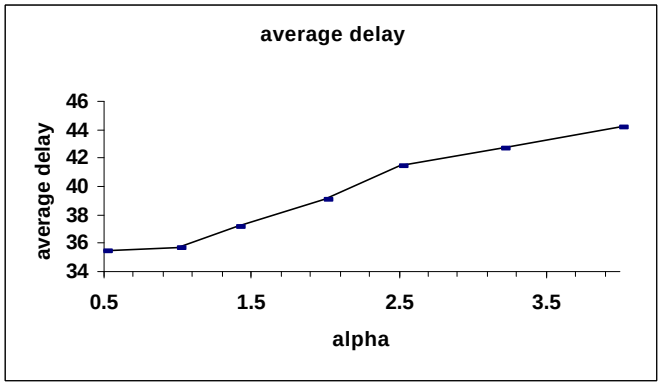

Figure 2: Average delay for $TASS - X^{\alpha}$.

V. NOISE

A. Rounding Errors

The previous section proved several stability results for $TASS - f(X)$, with a large class of functions f. However, if f is, for example, the square root, it is obvious that the packet switch will have to do some rounding before computing the maximum weight match. Additionally, the bandwidth of the channels between the linecards and the arbiter is limited, and therefore rounding weights may be useful to reduce the number of bits communicated to and stored by the scheduler.

For brevity, we'll only consider here the algorithm with $f(X) = X$. Assume that $E(n)$ is the quantization noise matrix at time slot n, $Z(n) = X(n) + E(n)$ is the quantized weight matrix, and TZ and MZ are the TASS and MWM algorithms with the perceived weight matrix equal to Z. Thus, there exists $\varepsilon' > 0$ such that $Pr(C(n) = MZ(n)) > \varepsilon'$. Then the following theorem shows that for a bounded error $E(n)$, TASS keeps its strong stability properties.

Theorem 4 *Assume that there exists a constant B such that $E(n) \le B$ for all n. Then, for any Bernoulli i.i.d. admissible traffic, and for all i,j, TZ satisfies:*

$$E[X_{ij}(n)] < \infty \tag{5}$$

Proof: Assume that $e(n) \equiv \langle Z(n), MZ(n) - TZ(n) \rangle$, $V(n) \equiv \frac{e(n)}{\varepsilon'} + \frac{X^2(n)}{2}$, and $DRIFT \equiv E[V(n+1) - V(n) | X(n)]$. Analyzing the two parts of V:

$$\begin{aligned}
&E[e(n+1)|X(n)] \\
&\le (1-\varepsilon') \cdot \{e(n) + E[\langle Z(n+1) - Z(n), MZ(n+1) - TZ(n) \rangle | Z(n), C(n+1) \ne MZ(n+1)]\} \\
&\le (1-\varepsilon') \cdot \{e(n) + E[\langle \Delta(n) + E(n+1) - E(n), MZ(n+1) - TZ(n) \rangle | \\
&Z(n), C(n+1) \ne MZ(n+1)]\} \\
&\le (1-\varepsilon') \cdot \{e(n) + 2N(B+1)\}
\end{aligned}$$

Similarly,

$$E\left[\frac{X^2(n+1) - X^2(n)}{2} \middle| X(n)\right] \le -\varepsilon \cdot \max_{i,j} |X_{ij}(n)| + d(n) + K_1 \text{ (with } K_1 \text{ a finite constant).}$$

Thus $DRIFT \le -\varepsilon \cdot \max_{i,j} |X_{ij}(n)| + d(n) - e(n) + K_2$ (with K_2 a finite constant).

$$\begin{aligned}
d(n) - e(n) &= \langle Z(n) - E(n), MWM(n) - TZ(n) \rangle - \langle Z(n), MZ(n) - TZ(n) \rangle \\
&= \langle Z(n), MWM(n) - MZ(n) \rangle - \langle E(n), MWM(n) - TZ(n) \rangle \\
&\le 0 + N
\end{aligned}$$

Finally $DRIFT \le -\varepsilon \cdot \max_{i,j} |X_{ij}(n)| + N + K_2$, hence the result.

❐

B. Delay Is Noise

The theorem on rounding noise has an application. Suppose that one wants to implement a pipelined version of TASS. Since there is a delay between the moment when weights are measured and the moment when they are used, the weights that are used in the computation are out-of-date. However, this delay is fixed, and therefore the number of arrivals and departures during this time is bounded – and so the noise is also bounded. Therefore, we have the following corollary.

Corollary 7 *The pipelined version of TASS is strongly stable for any Bernoulli i.i.d. admissible traffic.*

Proof: Suppose that the pipeline delay is equal to k.

$$X(n) = X(n-k) + \sum_{l=0}^{k-1} \{A(n-k+l) - D(n-k+l)\}$$
$$= Z(n) + \sum_{l=0}^{k-1} \{A(n-k+l) - D(n-k+l)\}$$

Hence $|E(n)| \le k$ and we can apply Theorem 4. (Note that it was also proved that a pipelined version of LPF is strongly stable in [5])

❒

Hence, we can consider Theorem 4 as generally representing a result on uncertainties of weights, these uncertainties coming from various sources such as rounding errors, pipelining, holding decisions, quantizing values, etc.

VI. Conclusions

In this paper we consider the performance of several crossbar scheduling algorithms. We focus on throughput, and explore the conditions under which the algorithms are strongly stable and give rise to 100% throughput. As a result, we find that various weight functions, noisy weights and even out-of-date weights give rise to 100% throughput.

VII. References

[1] M. Karol, M. Hluchyj, S. Morgan: "Input versus Output Queueing on a Space-Division Packet Switch", IEEE Trans. on Communications, vol. COM-35, no. 12, December 1987, pp. 1347-1356.

[2] Y. Tamir, G. Frazier: "High-Performance Multi-Queue Buffers for VLSI Communication Switches", Proc. of the 15th Int. Symp.on Computer Architecture, ACM SIGARCH vol. 16, no. 2, May 1988, pp. 343-354.

[3] J. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," in Proceedings of IEEE INFOCOM '00, Tel Aviv, Israel, March 2000, pp. 556 -- 564.

[4] N. McKeown, V. Anantharam and J. Walrand, "Achieving 100% throughput in an input-queued switch," IEEE INFOCOM 96, pp. 296-302, 1996.

[5] N. McKeown and A. Mekkittikul, "A practical scheduling algorithm to achieve 100% throughput in input-queued switches", IEEE INFOCOM 98, pp. 792-799, 1998.

[6] Y. Tamir and H. Chi. Symmetric crossbar arbiters for VLSI communication switches. IEEE Transactions on Parallel and Distributed Systems, 4(1):13--27, January 1993.

[7] T. E. Anderson, S. S. Owicki, J. B. Saxe, and C. P. Thacker,"High-speed switch scheduling for local-area networks," ACM Transactions on Computer Systems, vol. 11, no. 4, pp. 319 – 352, 1993.

[8] N. McKeown, "iSLIP: A Scheduling Algorithm for Input-Queued Switches", IEEE Transactions on Networking, Vol 7, No.2, April 1999

[9] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches", IEEE INFOCOM 98, vol. 2, pp. 533--539, 1998

[10] H.N. Gabow and R.E. Tarjan, "Faster Scaling Algorithms For Network Problems", SIAM Journal on Computing, 18:1013-1036, 1989

[11] A. V. Goldberg, S. A. Plotkin, and P. M. Vaidya. "Sublinear-Time Parallel Algorithms for Matching and Related Problems". Journal of Algorithms, 14:180--213, 1993

[12] Birkhoff, G.; "Tres obervaciones sobre el algebra lineal," Univ. Nac. Tucumán Rev. Ser. A5 (1946), pp.147-150.