Stability of a Frame-Based Maximal Weight Matching Algorithm with Transfer Speedup

Xike Li, *Student Member, IEEE* and Itamar Elhanany, *Senior Member, IEEE*

***Abstract*— It has been shown that maximal size matching algorithms for input-queued switches are stable under any admissible traffic conditions with a scheduling speedup of 2. However, as link speeds increase, the computational complexity of these algorithms limits their applicability in high port-density switches and routers. In this letter we describe a Frame-Based Maximal Weight Matching (FMWM) algorithm in which a new scheduling decision is issued once every several cell times. Between scheduling decisions, the configuration of the crossbar switch remains unchanged. We prove that the FMWM algorithm is stable with an internal buffer transfer speedup of 2, thereby significantly relaxing the timing constraints on the scheduling process. Simulation results illustrate the impact of the algorithm on the average cell delay for different traffic scenarios.**

***Index Terms*— Packet scheduling algorithms, switching architectures, Lyapunov stability analysis.**

I. Introduction

INPUT-QUEUED packet switching architectures are commonly employed by Internet routers, as they offer pragmatic scalability properties while requiring moderate memory bandwidth. Variable size packets are typically segmented into fixed size cells and are reassembled at the egress side of the switch/router. Arriving cells are buffered at the input ports prior to traversing a crossbar switch on route to their destination (egress) ports. A common technique for overcoming potential blocking and congestion at the input ports is called virtual output queueing (VOQ)[1]. In VOQ, a separate queue is maintained at the ingress port for each of the N output destinations. A scheduler is responsible for determining a matching configuration between inputs and outputs, whereby at most one input can be matched to one output at any given time, and vice versa.

A switch with a speedup of 1 is said to allow at most one cell from each input to traverse the crossbar during one time slot. Prior literature typically assumes that if a switch has a speedup of s, where $s \in \{1, ..., N\}$, it is said to issue s scheduling decisions during each time slot. Correspondingly, s transmissions of cells, from input queues to output ports may occur. We shall refer to such speedup as *scheduling* speedup. As line rates increase and the duration of packets decreases, scheduling speedup becomes a limiting factor with respect to scalability. To address this issue, we define the *transfer speedup,* η, as the ratio between the rate at which cells can be transferred from input buffers to output ports and the external cell arrival rate. Whenever $\eta > 1$, buffering is required at the output ports, since more than one cell may arrive during a single time slot. Such switching architectures are commonly referred to as combined input-and-output-queued (CIOQ)[2].

Many scheduling algorithms for input-queued switches have been proposed in recent years, with a common goal of offering scalability together with low delay characteristics. In the context of the latter, a fundamental requirement from any scheduling algorithm is stability. Stated coarsely, a switch is said to be stable if all its queues are bounded and, hence, do not backlog unlimitedly. Once a switch has been proven to be stable, its performance can be evaluated by means of simulations with reasonable confidence. It has been shown that for a broad class of traffic arrival patterns, all *maximal matching* algorithms yield a stable switch of any size with a *scheduling speedup* of 2 [2][3]. This stability property holds while delivering a throughput of up to 100%. A subset of maximal matching algorithms, which inherits all of the former's properties, is *maximal weight matching* (MWM) algorithms. In these algorithms greedy convergence to a maximal aggregate matching weight is obtained.

The increase in link rates directly causes a decrease in cell duration times to the point where cell-by-cell switching is no longer considered a pragmatic approach. To address this issue, we propose the frame-based maximal weight matching (FMWM) algorithm with *transfer speedup*, in which scheduling decisions are issued in accordance with the MWM algorithm, however they are kept unchanged for a duration of k consecutive time slots. By reconfiguring the crossbar switch once every several time slots, we significantly relax the timing constraints imposed on the scheduling algorithm [4].

II. Stability of the FMWM Algorithm with Transfer Speedup

Consider a CIOQ switch with N ports. Let $Q_{ij}(t)$ denote the VOQ size at input i holding cells destined to output j at time t. We define the corresponding random arrival process, $A_{ij}(t) \in \{0, 1\}$, with a mean rate of cell arrivals from input i to output j, $E[A_{ij}(t)] = \lambda_{ij} \leq 1$. We consider a simple FMWM algorithm which consists of an iterative process whereby in each iteration the maximal weight (queue size) is found and a match is recorded between its associated input-output pair. Each time a match is generated, its respective input and output pair is removed from contending during subsequent iterations. Assuming the weight matrix is not completely null, the number of iterations ranges from 1 to N.

The configuration of the crossbar, which is the outcome of the FMWM algorithm, can be represented by the permutation

Manuscript received April 30, 2005. The associate editor coordinating the review of this letter and approving it for publication was Prof. Iakovos Venieris. This work has been partially supported by the Dept. of Energy (DOE) under research grant DE-FG02-04ER25607.

The authors are with the Electrical & Computer Engineering Dept. at The University of Tennessee, Knoxville, TN (e-mail: {xli6, itamar}@utk.edu).

Digital Object Identifier 10.1109/LCOMM.2005.10032.

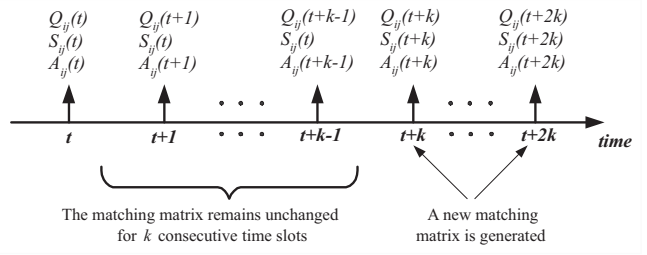

Fig. 1. Buffer dynamics under the FMWM scheduling algorithm.

matrix, $S(t) = \{S_{ij}(t)\}$, where $S_{ij}(t) = 1$ if input i is matched to output j at time t, otherwise $S_{ij}(t) = 0$. Based on the weights of the queues, a schedule is obtained which remains unchanged for k consecutive time slots. A new schedule will only occur at time $t+k$, reflected by $S_{ij}(t+k)$, as depicted in Fig. 1. It should be noted that although we restrict our attention to a weighting scheme which reflects only on the queue occupancies, a broader definition of queue weights may be applied.

Definition 1: An arrival process is said to be strictly admissible if

$$\sum_{i=1}^{N} \lambda_{ij} \leq 1 \; and \; \sum_{j=1}^{N} \lambda_{ij} \leq 1. \tag{1}$$

Definition 2: Let the *queue occupancy vector* be defined as

$$Q(t) = [Q_{11}(t), ..., Q_{1N}(t), ..., Q_{NN}(t)]^T. \tag{2}$$

Definition 3: The weight of the FMWM algorithm at time t is given by

$$\begin{aligned} W^{FMWM}(t) &= \sum_{i,j} Q_{ij}(t) S_{ij}^{FMWM}(t) \\ &= \left\langle Q(t), S^{FMWM}(t) \right\rangle \end{aligned} \tag{3}$$

where $S_{ij}^{FMWM}(t)$ denotes the matching configurations established by the algorithm at time t.

Theorem 1: A CIOQ switch running the FMWM scheduling algorithm with a transfer speedup of 2 is stable under admissible traffic for any frame size k.

Proof: We will derive the sufficient speedup value, η, as follows. Since at most k cells may arrive during k time slots, when applying the FMWM algorithm the following inequality holds

$$Q_{ij}(t+k) = [Q_{ij}(t) - \eta k S_{ij}(t)]^+ + \sum_{m=0}^{k-1} A_{ij}(t+m), \tag{4}$$

from which we can write

$$\begin{aligned} & Q_{ij}^2(t+k) - Q_{ij}^2(t) \\ \leq \; & 2Q_{ij}(t) \left[\left(\sum_{m=0}^{k-1} A_{ij}(t+m) \right) - \eta k S_{ij}(t) \right] + k^2, \end{aligned} \tag{5}$$

for $Q_{ij}(t) > \eta k$, and

$$Q_{ij}^2(t+k) - Q_{ij}^2(t) \leq 2\eta k^2 + k^2, \tag{6}$$

for $Q_{ij}(t) \leq \eta k$. The term $\eta k S_{ij}(t)$ expresses the ηk consecutive transmissions that may occur during a frame interval. Next, we construct a discrete-time quadratic Lyapunov function, $L(t)$, such that $L(t) = \langle Q_t, Q_t \rangle = \sum_{i,j} Q_{ij}^2(t)$ [5][6]. In order to prove the algorithm yields a stable queueing system, we would like to show that beyond a given threshold of maximal weight there is a negative drift in the state (queue occupancies) of the system. As an expression of a k time slot lag, we can write

$$L(t+k) - L(t) = \sum_{ij} \left(Q_{ij}^2(t+k) - Q_{ij}^2(t) \right). \tag{7}$$

By partitioning the above into the case of $Q_{ij}(t) < \eta k$ and $Q_{ij}(t) \geq \eta k$, we deduct the following

$$\begin{aligned} & E\left[L(t+k) - L(t) | Q(t)\right] \leq \\ & \sum_{ij} \left\{ 2Q_{ij}(t) \left[E\left(\sum_{m=0}^{k-1} A_{ij}(t+m) \right) - \eta k S_{ij}(t) \right] + k^2 \right\} \cdot \\ & \Pr\left(Q_{ij}(t) \geq \eta k\right) + \sum_{ij} k(2\eta k + k) \cdot \Pr\left(Q_{ij}(t) < \eta k\right) \\ & \leq \sum_{ij} 2Q_{ij}(t) \left[E\left(\sum_{m=0}^{k-1} A_{ij}(t+m) \right) - \eta k S_{ij}(t) \right] \\ & + \sum_{ij} 2(\eta+1)k^2 \\ & \leq 2(\eta+1)k^2 N^2 + \sum_{ij} 2Q_{ij}(t) \left[k\lambda_{ij} - \eta k S_{ij}(t) \right] \\ & \leq 2k \left[\langle \Lambda, Q_t \rangle - \eta \langle S, Q_t \rangle \right] + 2k^2 N^2 (1+\eta) \end{aligned} \tag{8}$$

where $\Lambda = \|\lambda_{ij}\|$ denotes the admissible arrival rate matrix, which is doubly substochastic. We, therefore, observe that for all $S_{ij}(t) \neq 0$,

$$2S_{ij}(t) = 2 > \sum_{l=1}^{N} (\lambda_{il} + \lambda_{lj}) \tag{9}$$

which stems from the fact that FMWM guarantees that $S_{ij}(t) \neq 0$ always points to the largest value on row i and column j, respectively. Since FMWM removes row i and column j after each iteration, (9) holds for all iterations. Given that Q_t is referred to identically on both sides of the inequality in (8), we conclude that $\langle \Lambda, Q_t \rangle < 2 \langle S, Q_t \rangle$. Hence, we may say that for $\eta \geq 2$ the following inequality holds: $\langle \Lambda, Q_t \rangle < \eta \langle S, Q_t \rangle = \eta W^{FMWM}(t)$. This suggests that there exists a value $\bar{\alpha} < 1$ for which $\langle \Lambda, Q_t \rangle < \bar{\alpha} \eta W^{FMWM}(t)$. Applying the latter to (8) yields

$$\begin{aligned} & E\left[L(t+k) - L(t) | Q(t)\right] \\ & \leq 2k\eta(\bar{\alpha} - 1) W^{FMWM}(t) + 2k^2 N^2 (1+\eta). \end{aligned} \tag{10}$$

Thus, for all $W^{FMWM}(t) > \frac{kN^2(1+\eta)}{\eta(1-\bar{\alpha})}$, we find that $E\left[L(t+k) - L(t) | Q(t)\right] < 0$, which concludes the stability proof.

III. Simulation Results

In order to evaluate the performance of the FMWM algorithm, two sets of simulations were carried out. In both cases, a 6-port switch was considered with a transfer speedup of 2.

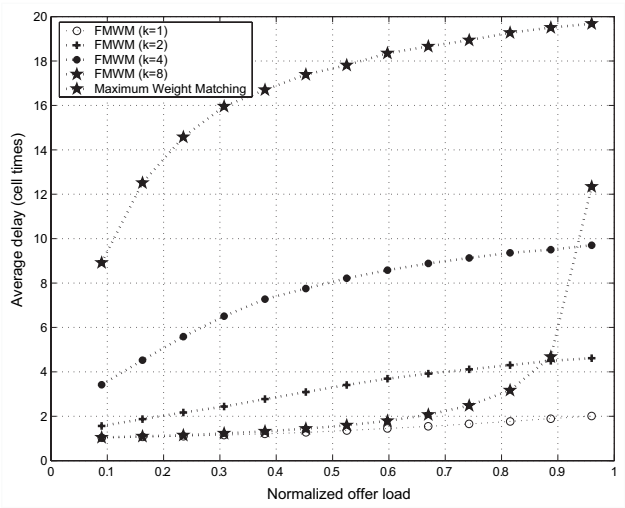

Fig. 2. Average cell delay when arrivals are Bernoulli i.i.d. and uniformly distributed, for different frame sizes (k), shown in comparison to maximum weight matching.

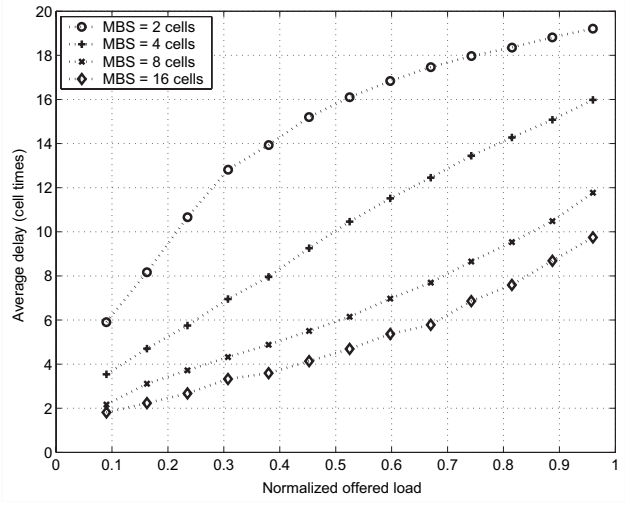

Fig. 3. Average cell delay as a function of the mean burst size for a fixed frame size of eight cells.

In the first simulation set, the arrival process was Bernoulli i.i.d. with uniform destination distribution. Fig. 2 depicts the average delay when employing FMWM for different frame sizes, compared to that of the Maximum Weight Matching algorithm. The latter is executed with a speedup of 1, since no additional speedup is required for its stability. As can be intuitively appreciated, the longer the frame the larger the average delay, which is due to the fact that during many switching intervals some queues may store less than k cells thereby under-utilizing the transmission intervals (frames). Moreover, it is noted that larger frame sizes exhibit faster delay growth (steeper slope), which can be explained by the fact that when a matching matrix is generated, the unmatched VOQs will not transmit any cells during k time slots, yet they continue to buffer newly arriving cells which increase the average delay.

The second set of simulations was targeted at examining the impact of bursty traffic on the delay characteristics. A two-state Markov-modulated (ON/OFF) process was assumed whereby bursts are uniformly distributed across the outputs. Fig. 3 shows the average delay as a function of the mean burst size for a fixed frame duration of 8 cells. An inverse relationship between the mean bursy size and the average delay is observed. The reason is that since the service discipline is inherently correlated, bursty traffic better utilizes the transmission intervals.

IV. Conclusions

This letter presents a frame-based maximal weight matching algorithm, with buffer transfer speedup, as a scalable scheduling scheme for large port-density input-queued switches. Through the use of Lyapunov functions, it was shown that a transfer speedup of 2 is sufficient to guarantee stability under any admissible traffic conditions. Since the service discipline governing FMWM is inherently correlated, it has been shown that cells in bursty traffic patterns often experience lower average delay than that experienced by cell arrivals which are Bernoulli i.i.d. This is an important observation in view of the fact that real-life data traffic tends to be correlated on different levels. The frame-based switching analysis presented can be broadened to address a range of other input-queued scheduling algorithms.

References

[1] Y. Tamir and G. L. Frazier, "High-performance multi-queue buffers for VLSI communications switches," in *Proc. ISCA '88: 15th Annual International Symposium on Computer Architecture*, pp. 343–354, 1988.

[2] J. G. Dai and B. Prabhakar, "The throghput of data switches with and without speedup," *IEEE INFOCOM*, pp. 556-564, Mar. 2000.

[3] I. Keslassy, R. Z. Shen, and N. McKeown, "Maximum size matching is unstable for any packet switch," *IEEE Commun. Lett.*, vol. 7, pp. 496-498, Oct. 2003.

[4] I. Elhanany and D. Sadot, "DISA: a robust scheduling algorithm for scalable crosspoint-based switch fabrics," *IEEE J. Select. Areas Commun.*, vol. 21, pp. 535-545, May 2003.

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

[6] P. R. Kumar and S. P. Meyn, "Stability of queueing networks and scheduling policies, *IEEE Trans. Automat. Contr.*, vol. 40, pp. 251-260, Feb. 1995.