Stability of Frame-Based Maximal Weight Matching Algorithms with Reconfiguration Delays

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

Abstract— **It has been shown that maximal weight matching algorithms for input-queued switches are stable under any admissible traffic conditions with a speedup of 2. As link speeds increase, the computational complexity of these algorithms limits their applicability in high port-density switches and routers. In Frame-Based Maximal Weight Matching (FMWM), a new scheduling decision is generated once every several packet times, as opposed to on a packet-by-packet basis. Between new scheduling decisions, the configuration of the crossbar switch remains unchanged. We show that the FMWM algorithm is also stable with a speedup of 2 and obtain the speedup required for stability in systems with non-negligible reconfiguration delays. Simulation results illustrate the impact of the frame size on the delay experienced by arriving packets under different traffic scenarios.**

Keywords— **Packet scheduling algorithms, switching architectures, stability analysis.**

I. Introduction

Input-queued packet switching architectures are commonly utilized in Internet routers as they offer pragmatic scalability while requiring moderate memory bandwidth. In such architectures, arriving packets are buffered at the ingress ports before traversing a crossbar switch en route to their destination (egress) ports. Typical switching fabrics partition variable size packets into fixed sized cells, each corresponding to a single internal time slot, which are later reassembled into packets prior to departing the router.

A common technique for overcoming potential blocking and congestion at the input ports is called virtual output queueing (VOQ). In VOQ, a separate queue is maintained at the ingress port for each of the N output destinations. Since each queue contains packets designated for a distinct output, the head-of -line blocking phenomena is avoided. A scheduler, whether implemented in a centralized or distributed manner, is responsible for receiving transmission requests from the virtual output queues and determining a matching (or scheduling) configuration between the inputs and the outputs, whereby at most one input is matched to one output at any given time, and visa versa.

A switch with a speedup of 1 is said to allow at most one packet from each input to traverse the crossbar during one time slot. If a switch has a speedup of s, where $s \in \{1, ..., N\}$, it is said to issue s scheduling decisions, and correspondingly s transmissions of packets from input buffers to output ports, during a single time slot. When $s > 1$ buffers are required at the output ports as well. Such architectures are commonly referred to as combined input-and-output-queued (CIOQ)[1]. Naturally, the need for speedup greater than 1 introduces stringent hardware constrains, as it necessitates high memory bandwidth resources. Many scheduling algorithms have been proposed in recent years, with a common goal of which to offer scalability (with respect to port densities and link rates) as well as high-performance. In the context of performance, 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, never backlog indefinitely. 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 speedup of 2 [1][2]. This stability property holds while delivering a throughput of up to 100%. One subset of maximal matching algorithms is *maximal weight matching* (MWM) algorithms, in which greedy convergence to a maximal aggregate matching weight is obtained. The increase in link rates directly causes a decrease in packet duration times to a point where packet-by-packet switching is no longer considered a pragmatic approach. To address this point, we introduce the frame-based maximal weight matching (FMWM) algorithm, 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 relax the timing constraints imposed on the scheduling algorithm. An immediate key question pertains to what are the speedup requirements under which such an algorithm yields a stable switch. Furthermore, given that stability is guaranteed, a complementing question would be: what are the implications of increased switching intervals on the performance? This paper aims to address these important questions.

The rest of the paper is structured as follows. Section II is dedicated to the stability proof of the FMWM scheduling algorithm. In Section III a bursty traffic model, which can generate 100% traffic loads, is presented. Section IV discusses simulation results, while in Section V the conclusions are drawn.

II. Stability of the FMWM Algorithm

Consider a CIOQ switch with N ports, as illustrated in figure 1. Let $Q_{ij}(t)$ denote the VOQ size at input i holding packets 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 packet arrivals from input i to output

The authors are with the Electrical & Computer Engineering department at The University of Tennessee, Knoxville. Email: {xli6,itamar}@utk.edu. This work has been partially supported by the Department of Energy (DOE) under research grant DE-FG02-04ER25607.

j, $E[A_{ij}(t)] = \lambda_{ij} \leq 1$. We consider the simple FMWM which consists on an iterative process whereby in each iteration the maximal weight is found and a match is registered between its associated input-ouput pair. Each time a match is generated, the respective input and output are removed from contending during the following 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 matrix $S(t) = \{S_{ij}(t)\}$, where $Sij(t) = 1$ if input i is matched to output j at time t, otherwise $Sij(t) = 0$.

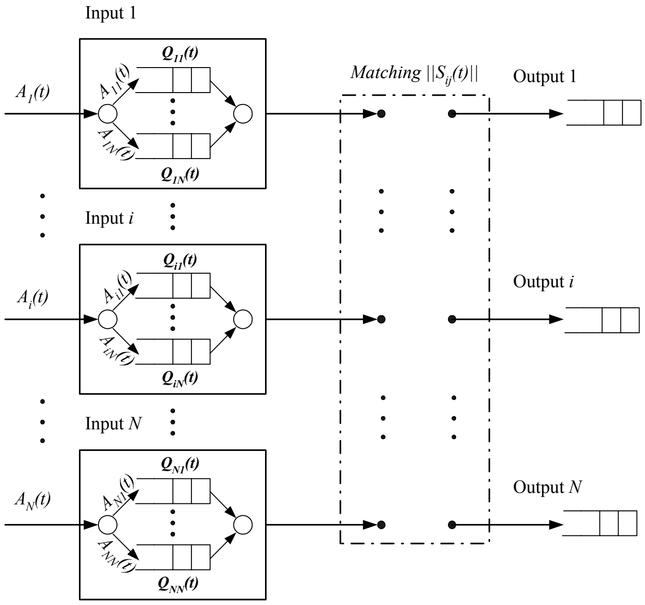

Fig. 1. An N-port combined-input-output-queued (CIOQ) switch architecture.

Without loss of generality, let us assume that at time t we have explicit knowledge of $Q_{ij}(t)$. We thus assign a weight value, equal to the queue length, to each VOQ based on which the scheduler establishes the maximal weight matching for k consecutive time slots. As a result, at time $t+k$, we have a new matching/scheduling matrix $S_{ij}(t+k)$. As depicted in figure 2, the matching matrix remains unchanged during the following k time slots. 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.

There have been many crossbar technologies introduced in the literature with applications to input-queued packet switching fabrics. These range from electronics-based to optical technologies, such as wavelength division multiplexing (WDM) passive stars [3]. In most practical systems, there is a non-negligable reconfiguration delay that is introduced when the crosspoint switch is configured. Such delay can range from several nanoseconds to a few tens of microseconds. In addition, high-speed serializer/deserializer (SerDes) devices typically require somewhere in the order of 100 bit-times to lock on to a new signal. This locking process becomes ever more intricate as link speeds continue to increase. Similar durations are observed in optical WDM tunable lasers, which require time to switch from one wavelength to another.

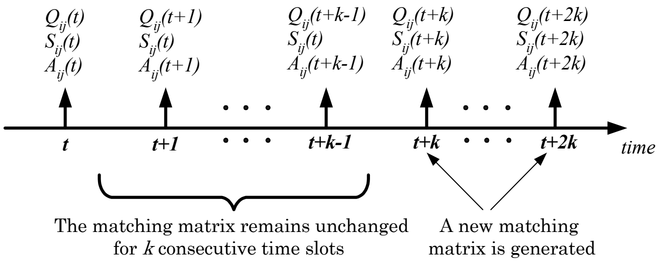

Fig. 2. Buffer dynamics under the FMWM scheduling algorithm.

In the interest of incorporating such reconfiguration delays, let us define κ_R as the mean duration (in bit times) required for the system to reconfigure itself. Letting p denotes the number of payload bits in each fixed-size packet, the total number of bits in a frame is $\kappa_R + k \cdot p$.An obvious way of overcoming the inefficiency introduced by the reconfiguration dead-times, is to speedup the transfer rate of packets from the VOQs to the output buffers. Correspondingly, a speedup factor of 2 suggests that up to two packets can be transferred from each VOQ to an output buffer during a single time slot. The output buffers transmit the queued packets at the link rate, in the order they arrive. It should be noted that while the notion of speedup commonly implies multiple scheduling cycles during each time slot, here we simply assume an increase in the internal packet transfer rate such that no increase of computational effort is required.

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 produced by the FMWM algorithm at time t is given by

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

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

Theorem 1: An input-queued switch with an average reconfiguration delay of κ_R, running the FMWM scheduling algorithm with a speedup of 2 $\left(1+\frac{\kappa_R}{kp}\right)$ is stable under admissible traffic for any finite frame size k.

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

$$Q_{ij}(t+k-1) - Q_{ij}(t) \leq k, \tag{4}$$

from which we can write

$$Q_{ij}(t+k-1) - Q_{ij}(t) \leq \sum_{m=0}^{k-1} A_{ij}(t+m) - \eta k S_{ij}(t), \tag{5}$$

for $Q_{ij}(t) \geq \eta k$. 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 $L(t) = \langle Q_t, Q_t \rangle = \sum_{i,j} Q_{ij}^2(t)$ [4][5].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

$$\begin{aligned} & L(t+k-1) - L(t) = \\ & \sum_{ij} \left(Q_{ij}(t+k-1) - Q_{ij}(t)\right)\left(Q_{ij}(t+k-1) + Q_{ij}(t)\right). \end{aligned} \tag{6}$$

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

$$\begin{aligned} & E\left[L(t+k-1) - L(t) | Q(t)\right] \leq \\ & \sum_{ij}\left(\sum_{m=0}^{k-1} A_{ij}(t+m) - \eta k S_{ij}(t)\right) E\left[2Q_{ij}(t) + k\right] \cdot \\ & \Pr\left(Q_{ij}(t) \geq \eta k\right) + \sum_{ij} k(2\eta k + k) \cdot \Pr\left(Q_{ij}(t) < \eta k\right) \\ & \leq \sum_{ij}\left(\sum_{m=0}^{k-1} A_{ij}(t+m) - \eta k S_{ij}(t)\right) E\left[2Q_{ij}(t) + k\right] \\ & + \sum_{ij} k(2k\eta + k) \\ & \leq \sum_{ij} 2E\left[Q_{ij}(t)\right]\left(k\lambda_{ij} - \eta k S_{ij}(t)\right) \\ & + \sum_{ij} k^2 + k(2\eta k + k) \\ & \leq 2k\left[\langle \Lambda, Q_t \rangle - \eta \langle S, Q_t \rangle\right] + 2k^2N^2(1+\eta) \end{aligned} \tag{7}$$

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

$$2S_{ij}(t) = 2 > \sum_{l=1}^{N} (\lambda_{il} + \lambda_{lj}) \tag{8}$$

which stems from the fact that FMWM guarantees that $S_{ij}(t) \neq 0$ always points to the largest value on raw i and column j,respectively. Since FMWM removes raw i and column j after each iteration, (8) holds for all iterations. Hence, since Q_t is referred to identically on both sides of the inequality in (7), we conclude that $\langle \Lambda, Q_t \rangle < 2 \langle S, Q_t \rangle$. However, the latter does not take into account the additional speedup needed to overcome the reconfiguration delays. If κ_R denotes the portion of the frame that is "wasted" on reconfiguration, it implies that by speeding up the transmission of actual payload bits by $1+\frac{\kappa_R}{kp}$, the dead-times can be compensated for. Since the algorithm speedup requirement of 2 is independent of the additional speedup needed to compensate for reconfiguration dead-times, we conclude that for $\eta \geq 2\left(1+\frac{\kappa_R}{kp}\right)$ we have $\langle \Lambda, Q_t \rangle < \eta \langle S, Q_t \rangle = \eta W^{FMWM}(t)$. 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 (7) yields

$$\begin{aligned} & E\left[L(t+k-1) - L(t) | Q(t)\right] \\ & \leq 2k(\bar{\alpha}-1)W^{FMWM}(t) + 2k^2N^2(1+\eta). \end{aligned} \tag{9}$$

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

III. Multi-Queue ON/OFF Arrival Process with Maximal Throughput

In order to better evaluate the behavior of the FMWM algorithm, we have chosen to apply both bursty and non-bursty arrival patterns. In this section, we describe the bursty traffic generation model employed.

Consider a discrete-time, two-state Markov chain generating arrivals modeled by an ON/OFF source which alternates between the ON and OFF states. An arrival is generated for each time slot that the Markov chain spends in the ON state. Let the parameters α and β denote the probabilities that the Markov chain remains in states ON and OFF, respectively. Using α and β the load and mean burst sizes may be directly obtained. However, since at least a single OFF state separates two consecutive bursts, the maximal arrival rate is bounded by $B/(1+B)$ where B denotes the mean burst size. This bound is a significant limiting factor in the evaluation of high-speed packet switching fabrics. Moreover, in such evaluations there is a clear need for a heterogeneous model in which both burst sizes and average rates can be flexibly determined[6][7].

To overcome the maximal traffic load constraint, we introduce the ON/OFF/Ω Markov-modulated arrival process, whereby in a transition from the ON state the process visits an Ω state while generating an additional packet to the same destination, It is only from the Ω state that the process can transition back to the OFF state. Consider a general case where an arriving packet can be destined to each of the N possible destination. As shown in figure 3, an arrival destined to the output i is generated for each time slot that the Markov chain spends in the ON$_i$ state, i=1,2,…,N, while no arrivals are noted when in the OFF state. Instead of directly transitioning from ON$_i$ states to OFF, a transition from ON$_i$ to Ω_i is prescribed. An arrival destined to output i is still generated

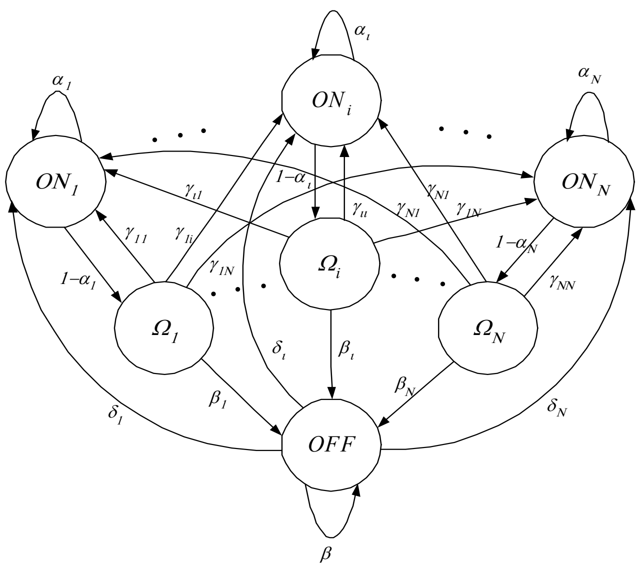

Fig. 3. The proposed maximal-throughput bursty arrival model.

while the Markov chain is in the Ω_i state. To that end, there are a total of *2N+1* states in the proposed model.

It can be shown that all transition probability can be obtained by solving a set of linear equations which are determined by the mean burst and arrival rates for each destination and visa versa. The mean arrival rate per output can be expressed as $\lambda_i = \pi_{ON_i} + \pi_{\Omega_i}$, while the mean burst size for output i is given by $\mathrm{MBS}_i = 1 + \frac{1}{1-\alpha_i}$. For the case of uniform distribution and identical mean burst sizes $(\beta_i \equiv \beta, \lambda_i = \lambda/N)$, we have $\gamma_{ij} = \frac{1-\beta}{N-1}$, $\forall i \neq j$. The key attribute of this model is that given any set of mean burst sizes and any traffic load distribution, the Markov chain in figure 3 can be constructed so as to yield the desired traffic generation engine. More importantly, the latter can achieve 100% traffic load.

IV. Simulation Results

In order to evaluate the performance of the FMWM algorithm under different traffic conditions and interval durations, three simulation sets were carried out. In all simulations a 6-port switch was considered with a speedup of $2\left(1+\frac{\kappa_R}{kp}\right)$. In the first simulation set, the arrival process was Bernoulli i.i.d. with uniformly distributed destination distribution. Figure 4 depicts the mean delay when employing FMWM with different switching frame sizes (k). As can be intuitively appreciated, the longer the frame the larger the mean delay, which stems from the fact that during many switching intervals less than k consecutive packets are being transmitted. Moreover, it is noted that larger frame sizes exhibit faster delay growth (steeper slope). This is because once a matching matrix is generated, the unmatched VOQs will not transmit any cells during the follow k time slots, yet still buffer newly arriving cells (which contribute to the average waiting time).

The second set of simulations examined the impact of bursty traffic on the performance of the FMWM algorithm.

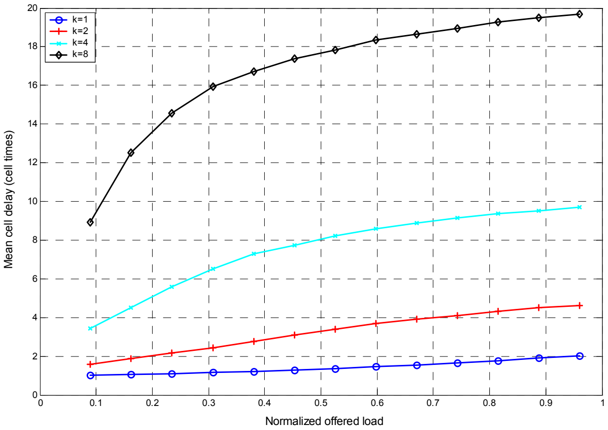

Fig. 4. Mean cell delay for Bernoulli i.i.d. uniformly distributed arrivals and different frame sizes (k).

Uniformly distributed bursty traffic with identical mean burst size (MBS) was employed. Figure 5 depicts the resulting mean delay for different frame sizes ($k = 1, 2, 4, 8$) but same mean burst size ($MBS = 8$). An interesting observation here is that the difference in delay between the first three frame sizes ($k = 1, 2$ and 4) is small relative to the delay increase shown for $k = 8$. This can be explained by the fact that bursts which are smaller or equal to the frame size result in fully utilized transmission intervals, while packets in bursts that are larger than the frame size experience higher average delay.

The last set of simulations was targeted at examining the impact of the mean burst size on the delay performance. Once again, a 6 port switch was considered whereby bursts are uniformly distributed across the outputs. Figure 6 shows the mean delay as a function of the mean burst sizes for a fixed frame size of 8 cells. A clear difference in relative performance is observed between the lower load and higher load regions. It appears that the larger the mean burst size the slower the relative increase in delay due to higher loads. This could be explained by the fact that the large mean burst sizes already introduce considerable delays at lower loads due to "overflowing" the frame boundaries. As such, the increase in the average delay is not as drastic as that of traffic which carries smaller mean burst sizes.

V. Conclusions

This paper studies the frame-based maximal weight matching algorithm as a scalable scheduling scheme for large port-density input-queued switches. Through the use of Lyapunov functions, a sufficient speedup needed to guarantee stability has been obtained. Since the service discipline governing FMWM is inherently correlated, it has been shown that packets belonging to bursty traffic often experience lower average delay than that experienced by packet 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. Moreover,

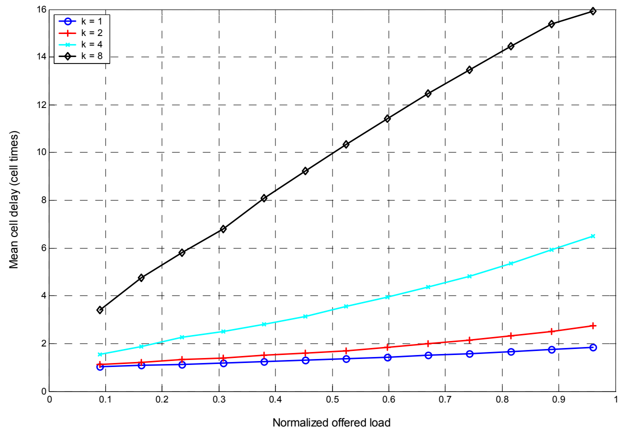

Fig. 5. Mean cell delay as a function of the offered load for uniformly distributed arrivals with mean burst size of 8 cells and different frame sizes (k).

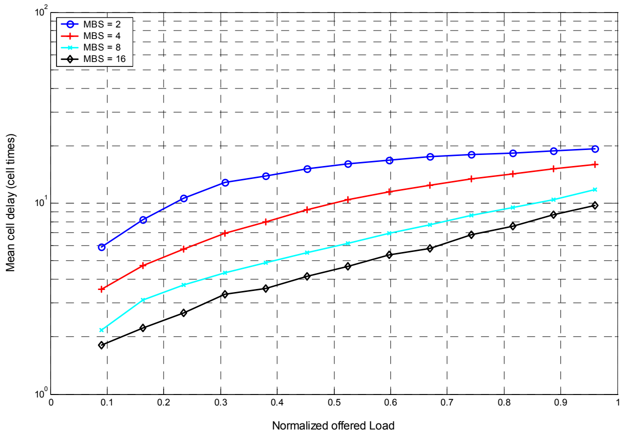

Fig. 6. Mean cell delay as a function of the mean burst size (MBS) for a frame of 8 cells.

the frame switching analysis framework presented here can be broadened to address a range of input-queued scheduling algorithms.

References

[1] J. Dai and B. Prabhakar, “The throghput of data switches with and without speedup,” *IEEE INFOCOM 2000*, pp. 556–564, March 2000.

[2] I. Keslassy, R. Zhang-shen, and N. McKeown, “Maximum size matching is unstable for any packet switch,” *IEEE Communication Letters*, vol. 7, pp. 496–498, October 2003.

[3] D. Sadot and I. Elhanany, “Optical switching speed requirements for terabit/sec packet over wdm networks,” *IEEE Photonic Technology Letters*, vol. 12, pp. 440–442, April 2000.

[4] P. Kumar and S. Meyn, “Stability of queueing networks and scheduling policies,” *IEEE Transactions on Automatic Control*, vol. 40, pp. 251–260, February 1995.

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

[6] A. L. Corte, A. Lombardo, and G. Schembra, “Modeling superposition of on-off correlated traffic sources in multimedia applications,” *IEEE INFOCOM 1995*, pp. 993–1000, 1995.

[7] I. Elhanany, D. Chiou, V. Tabatabaee, R. Noro, and A. Poursepanj, “The network processing forum switch fabric benchmark specifications: An overview„” *IEEE Network Magazine*, March 2005.