# Delay Bounds for Approximate Maximum Weight Matching Algorithms for Input Queued Switches

Devavrat Shah, Milind Kopikare
*devavrat@cs.stanford.edu,kopi@stanford.edu*
Dept. of CS, Stanford University; Dept. of EE, Stanford University

*Abstract*— **Input Queued(IQ) switch architecture has been of recent interest due to its low memory bandwidth requirement. A scheduling algorithm is required to schedule the transfer of packets through cross-bar switch fabric at everytime slot. The performance, that is throughput and delay, of a switch depends on the scheduling algorithm. The Maximum weight matching(MWM) algorithm is known to deliver 100% throughput under any admissible traffic [2][3][4]. In [5], Leonardi et. al. obtained non-trivial bound on the delay for MWM algorithm under admissible Bernoulli i.i.d. traffic. There has been a lot of interesting work done over time to analyze throughput of scheduling algorithms. But apart from [5], there has not been any work done to obtain bounds on delay of scheduling algorithms. The MWM algorithm is perceived to be very good scheduling algorithm in general and simulations have suggested that it performs better than most of the known algorithms in terms of delay. But it is very complex to implement. Hence many simple to implement approximations to MWM are proposed.**

**In this paper, we study a class of approximation algorithms to MWM, which always obtain a schedule whose weight $W$ differs from the weight of MWM schedule $W^*$ by at most $f(W^*)$, where $f(.)$ is a sub-linear function. We call this difference in weight as "approximation distance" of algorithm from MWM. We denote this class of algorithms by 1-APRX. We prove that any 1-APRX algorithm is stable, that is, it delivers upto 100% of throughput under any admissible Bernoulli i.i.d. input traffic. Under any admissible Bernoulli i.i.d. traffic, we obtain bounds on the average queue length(equivalently delay) of the 1-APRX algorithms using a Lyapunov function technique, which was motivated in [5]. The delay bounds obtained for the 1-APRX algorithm is linearly related with the "approximation distance", which matches the intuition that better the weight of schedule, better the algorithm will perform. Interestingly, simulations show a linear relationship between the average queue length(equivalently delay) and the "approximation distance". Thus, the "approximation distance" of a scheduling algorithm can serve as a metric to differentiate between the performance of different stable algorithms, even though throughput may be same for these algorithms.**

**We also obtain a novel heuristic tighter bound on the average queue length (equivalently delay) under uniform Bernoulli i.i.d. traffic for MWM using a very simple argument.**

## I. Introduction and motivation

Output Queued switches (or known architectures other than the IQ architecture) are becoming increasingly difficult to implement due to its high memory bandwidth requirements and hence poor scaling at high line speeds. Input Queued switches on the other hand, have been of recent interest among researchers and industry people because of its capability of operating at high line speeds with lower memory bandwidth requirement. We briefly introduce the popular model of the input-queued switch with crossbar architecture.

Consider the $N \times N$ crossbar IQ switch shown in figure 1. At each of the inputs, there are $N$ separate Virtual Output Queues (VOQ) corresponding to each of the $N$ outputs. A VOQ for output $j$ at input $i$ is denoted by $\text{VOQ}_{ij}$. Let $Q_{ij}(t)$ denote the number of packets in $\text{VOQ}_{ij}$ at time $t$. At any time $t$, consider the bipartite graph induced by these $N$ inputs-outputs, where an input $i$ has an edge to output $j$ iff $Q_{ij}(t) > 0$. The cross-bar constraints of an IQ switch require that at every time slot

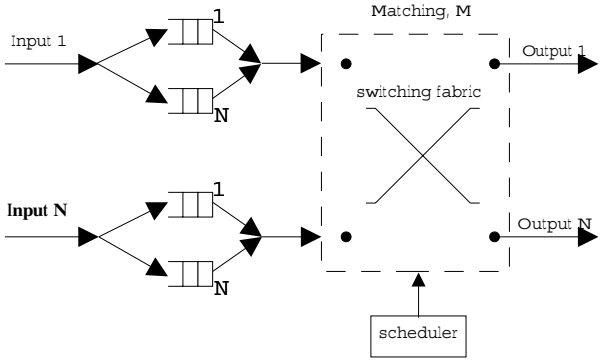


Fig. 1. Logical structure of an input-queued cell switch

$t$, each input can be connected to at most one output and each output can receive at most one packet from inputs. This means that a feasible "schedule" of input-output is a "matching" on this bipartite graph. A matching of input-output can be represented as a permutation matrix $\pi = (\pi_{ij})_{i,j \le N}$: $\pi_{ij} = 1$ iff input $i$ is matched to output $j$ in the matching. A scheduling or matching algorithm $\mathcal{S}$ obtains a permutation (matching) $\pi^{\mathcal{S}}(n)$ for every time slot $t$. In this paper, we consider all scheduling algorithms with *speed up* 1, that is, packets are transferred only once per time slot. Let $\lambda_{ij}$ denote the arrival rate at input $i$ for output $j$. An arrival traffic is called *admissible* if (a) $\sum_j \lambda_{ij} < 1, \forall i$, and (b) $\sum_i \lambda_{ij} < 1, \forall j$.

**Definition 1. Weight of a schedule:** Weight of a VOQ refers usually, but not restricted to, the length (number of packets in backlog) of the VOQ. Weight of the schedule is the sum of the weight of all the VOQs that have been scheduled (matched) to the outputs in that time slot. That is, if $\pi = [\pi_{ij}]$ is a schedule and $Q(t) = [Q_{ij}(t)]$ is the switch state at time $t$, then the weight of schedule $\pi$ at time $t$ is $\sum_{ij} \pi_{ij} Q_{ij}(t)$. We will also use the *inner product* notation to define weight: the inner product of $\pi$ and $Q(t)$ is $\langle \pi, Q(t) \rangle = \sum_{ij} \pi_{ij} Q_{ij}(t)$. We denote the weight of schedule $\pi$ by $W_\pi$.

The well known maximum weight matching (MWM) scheduling algorithm finds the matching (schedule) with maximum weight among all possible $N!$ matchings. MWM is know to deliver 100% throughput for any admissible traffic [2], [3], [4]. In [5], Leonardi et. al. obtained non-trivial bound for MWM under any admissible Bernoulli i.i.d. traffic. Simulation study by many researchers has suggested that MWM provides very good delay properties. Thus MWM has desirable properties. But it is very complex to implement. This has led to many simple approximations to MWM. Consider the following class of approximations to MWM:

**Definition 2. 1-APRX:** Let the weight of a schedule obtained by a scheduling algorithm $B$ be $W_B$. Let the weight of the maximum weight match for the same switch state be $W^*$. $B$ is defined to be a 1-APRX to MWM, if the following property is always true: $W_B \geq W^* - f(W^*)$, where $f(\cdot)$ is a sub-linear function, that is, $\lim_{x\to\infty} \frac{f(x)}{x} = 0$ for any switch state.

For a 1-APRX algorithm $B$, the bound on the difference between its weight and weight of MWM is defined as "approximation distance" of that algorithm to MWM. Note that in the above definition, we can call $f(W^*)$ as an "approximation distance" of the algorithm $B$. To avoid ambiguity on the notion of "approximation distance", we will denote the smallest such $f(W^*)$ as the approximation distance.

We will show that all 1-APRX algorithms deliver 100% throughput for any admissible traffic. Since throughput does not seem to be a good enough metric to differentiate between these algorithms, we will also study the delay offered by these algorithms. In this paper, we will use delay as a metric to evaluate the performance of all such approximate algorithms. We will analyze, both theoretically and through experiments, the delay bounds on the class of 1-APRX algorithms for MWM.

This paper is mainly about analysis of the average delays of scheduling algorithms. We first obtain heuristic delay bounds for the MWM scheduling algorithm under i.i.d. uniform traffic using a very simple argument. This turns out to be tighter than the bounds obtained in [5]. Interestingly, these bounds can be extended for the non-i.i.d. traffic but not non-uniform traffic. Next we obtain bounds on delay (average queue length) for 1-APRX algorithms. As it turns out, these bounds are linearly related to the "approximation distance" of these algorithms. Extended simulation study confirms this observation. We would like to note that this matches the intuition: *better the approximation an algorithm is to MWM in terms of "weight", better it is in terms of delay.*

The rest of the paper is organized as follows: In section II, we discuss the stability and delay bounds of the 1-APRX algorithms. In particular, section II-B.1 discusses the (heuristic) novel tighter bounds on MWM under uniform traffic. In section II-B.2, we present the technique to analyze the bounds on the average queue length or delay for 1-APRX schemes. In section III, we consider two particular algorithms, which can be seen as implementable versions of MWM. We show that they are 1-APRX algorithms. In section III-A, we provide an extensive simulation study about the average delay of these algorithms, which confirms our theoretical results. Finally, we conclude in section IV.

## II. 1-APRX Algorithms: Stability and Delay Bounds

In this section, we consider the performance of 1-approximation algorithms to MWM in terms of throughput and delay bounds.

### A. Approximate Algorithms: Stability

The following theorem proves that 1-APRX algorithms deliver 100% throughput.

**Theorem 1.** *Let $W^\star(t)$ denote the weight of maximum weight matching schedule at time $t$, with respect to switch state $Q(t)$. Let $B$ be a 1-approximation algorithm to MWM. Let $W^B(t)$ denotes its weight at time $t$. Further, $B$ has property that,*

$$W^B(t) \geq W^*(t) - f(W^*(t)), \forall t,$$

*where $f(\cdot)$ is a sub-linear function. Then, the scheduling algorithm $B$ is stable under any admissible Bernoulli i.i.d. input traffic.*

*Proof.* We will use the approach similar to [2]. Let $V(Q(t)) = \sum_{i,j} Q_{ij}^2(t)$ be the usual quadratic Lyapunov function. To establish stability it suffices to prove that for some $\delta > 0$ and $K > 0$

$$E(V(Q(t+1)) - V(Q(t))|Q(t)) \leq -\delta W^\star(t), \quad \text{whenever } W^\star(t) \geq K,$$

Consider the following:

$$V(Q(t+1)) - V(Q(t)) = \sum_{i,j}[Q_{ij}^2(t+1) - Q_{ij}^2(t)] = \sum_{i,j}[Q_{ij}(t+1) - Q_{ij}(t)][Q_{ij}(t+1) + Q_{ij}(t)].$$

Let $S(t) = [S_{ij}(t)]$ be the schedule used by $B$ at time $t$ and let $A_{ij}(t)$ denote arrivals to $VOQ_{ij}$ at time $t$. We know that

$$\begin{aligned} Q_{ij}(t+1) &= [Q_{ij}(t) - S_{ij}(t)]^+ + A_{ij}(t+1) \\ &\leq \max\{[Q_{ij}(t) - S_{ij}(t)] + A_{ij}(t+1), 1\}. \end{aligned} \tag{1}$$

Hence, we obtain

$$\begin{aligned} &V(Q(t+1)) - V(Q(t)) \\ &\quad\leq \sum_{i,j}[(A_{ij}(t+1) - S_{ij}(t))(2Q_{ij}(t)+1) + 1] \\ &\quad\leq \sum_{i,j}[(A_{ij}(t+1) - S_{ij}(t))(2Q_{ij}(t))] + 2N^2 \end{aligned}$$

Taking conditional expectations with respect to $Q(t)$ yields

$$\begin{aligned} &E(V(Q(t+1)) - V(Q(t))|Q(t)) \\ &\quad\leq 2\sum_{ij} Q_{ij}(t)[E(A_{ij}(t) - S_{ij}(t)|Q(t))] + 2N^2 \\ &\quad= 2\sum_{ij} Q_{ij}(t)[\lambda_{ij} - S_{ij}(t)] + 2N^2 \end{aligned}$$

Since the arrival rate matrix, $\Lambda$, is admissible it is strictly doubly sub-stochastic. Therefore, from arguments made in Lemma 2 of [2], we may write $\sum_{ij} Q_{ij}(t)\lambda_{ij} = \langle Q(t), \Lambda\rangle \leq \sum_k \gamma_k \langle \Pi_k, Q(t)\rangle$, where the $\Pi_k$ are permutation matrices and $\gamma_k \geq 0$ and $\sum_k \gamma_k < 1$.

Let $W_{\Pi_k} = \langle \Pi_k, Q(t)\rangle$ and let $\delta = 1 - \sum_k \gamma_k$. Putting the

above observations together, we get

$$\begin{aligned}
&E(V(Q(t+1)) - V(Q(t))|Q(t)) \\
&\quad \leq 2(\sum_k \gamma_k W_{\Pi_k}(t) - W^B(t)) + 2N^2 \\
&= 2(\sum_k \gamma_k W_{\Pi_k}(t) - W^\star(t) + W^\star(t) - W^B(t)) + 2N^2 \\
&\leq 2(\sum_k \gamma_k - 1)W^\star(t) + 2f(W^\star(t)) + 2N^2 \\
&\qquad = -2\delta W^\star(t) + 2f(W^\star(t)) + 2N^2
\end{aligned}$$

Since $f(\cdot)$ is sub-linear function, for large enough constant $K > 0$, we obtain,

$$E(V(Q(t+1))-V(Q(t))|Q(t)) \leq -\delta W^\star(t), \quad W^\star(t) \geq K$$

This proves the stability of algorithm $B$. □

This theorem shows that, all such 1-approximation algorithms have the same throughput region as MWM. Thus only throughput consideration does not let us differentiate between the performance of such algorithms. This motivates the study of delay bounds of these algorithms, which we study next.

### *B. Delay Bounds*

In this section we present different theoretical bounds on average delays of 1-APRX scheduling algorithms.

#### B.1 Delays for MWM

In [5], Leonardi et. al. provided bounds on performance of MWM. Their method provides tight bounds for uniform traffic, that is the arrival rate for each VOQ is the same. In this section we present a very simple but powerful way to obtain heuristic bounds on the performance of MWM under uniform traffic. These bounds can be extended to non-i.i.d. arrival traffic, but it is particular to uniform traffic (rather particular to throughput region where $\lambda_{ij} < 1/N \forall i, j$). As it turns out, these bounds are little tighter than the bounds provided in [5].

We first consider the following simple randomized scheduling algorithm, which we denote as RANDOM:
*(a)* Every cell-time, pick a matching $R$ uniformly at random out of all $N!$ possible matchings.
*(b)* Use matching $R$ as schedule for this particular time.

Intuitively it seems that the algorithm RANDOM is worse than MWM (at the same time, we do not know if that is true !). Hence, the average delay(or average queue length) under RANDOM should serve as an upper bound for MWM. This motivates the analysis of the average queue length under this RANDOM policy.

Under uniform traffic, the arrival rates are such that, $\lambda_{ij} = \lambda < \frac{1}{N}, \forall i, j$. Under scheduling policy RANDOM, the probability that a particular VOQ $Q_{ij}$ receives a service, is $\frac{(N-1)!}{N!} = \frac{1}{N}$. Thus, all queues $Q_{ij}(1 \leq i, j \leq N)$ become discrete time queues with Geometric arrival process of rate $\lambda$ and Geometric service process of rate $1/N$, which is a discrete approximation to usual M/M/1 queue. Thus to obtain average queue length (or delay bounds) we need to analyze a simple discrete time queue with such friendly arrival-departure distribution. We state the following well known lemma from very basic queueing theory.

**Lemma 1.** *For a FCFS discrete time version of M/M/1 queue, which has Geometric arrival process of rate $\lambda$(probability of arrival) and Geometric service process with rate $\mu > \lambda$, the average queue length is*

$$\frac{\lambda(1-\mu)}{(\mu-\lambda)}$$

*Proof.* We will sketch the proof of this lemma. We can model this system as a discrete time Markov chain, with the state of this system being the number of packets in the queue. Let $X(t)$ denote the state(number of packets) of the system at time $t$. At time $t+1$, an arrival occurs with probability $\lambda$ and departure occurs with probability $\mu$. Thus, $X(t+1) = X(t) + 1$ with probability $\lambda(1-\mu) \stackrel{\triangle}{=} a$, $X(t+1) = [X(t)-1]^+$ with probability $\mu(1-\lambda) \stackrel{\triangle}{=} b$ and $X(t+1) = X(t)$ otherwise. This can be easily analyzed and the steady state distribution gives the desired average queue length as $\frac{\lambda(1-\mu)}{(\mu-\lambda)}$. An interested reader can look into [13] for reference. □

From the above discussion, and Lemma 1 we obtain the following theorem:

**Theorem 2.** *Under i.i.d. uniform traffic with arrival rate $\lambda_{ij} = \lambda, \forall i, j$, under the* RANDOM *scheduling algorithm, in the equilibrium the average queue length of a VOQ $Q_{ij}, \forall i, j$ is*

$$L(\lambda) = \frac{\lambda(1-1/N)}{(1/N-\lambda)}$$

*where $\lambda < 1/N$.*

The normalized loading factor of switch arrival process is $\rho = \lambda N$. Hence from the above theorem, the average queue length is

$$L(\rho) = \frac{\rho}{(1-\rho)}\frac{N-1}{N}$$

Intuitively, this is a possible upper bound on the average queue length under MWM. We do not claim this as an upper bound since there is no theoretical result that says that the delays of RANDOM are worse than that of MWM. Let us compare this bound with the bound obtained by Leonardi et. al. in [5] as,

$$\tilde{L}(\rho) = \frac{\rho}{(1-\rho)}\frac{N-\rho}{N}$$

Since, $\rho < 1$ it immediately implies that,

$$L(\rho) < \tilde{L}(\rho)$$

Note that, in analysis of the policy RANDOM, we used results of the average queue length for a discrete time FCFS queue with Geometric arrivals-services. There are many results known in queueing theory about the average delay of such queue under general arrival traffic assumption, depending on the characteristics of traffic(see [13]). All such bounds apply to the case of RANDOM, which provide a heuristic bounds on the average queue length under MWM policy for uniform traffic.

We would like to note that, the average queue length and average delay are related by Little's law: $L = \lambda W$, where $L$ is

the average length, $W$ average delay and $\lambda$ arrival rate. Thus bounds on average queue lengths and average delays can be derived from each other. We would like to note that, because of this reason, we will use bound on delay and bound on queue length interchangeably. As a note, stable switches do observe Little's Law, even though they may not be work-conserving.

### B.2 Delay bounds for Approximate Algorithms

In this section, to obtain bounds for all 1-APRX algorithms of MWM, we develop a new technique motivated from [5]. We would like to note that, these techniques(this paper and [5]) have similar flavor to previously well-known results, e.g. [15]. We obtain the bounds on the average queue length under any admissible Bernoulli i.i.d. arrival traffic.

For all 1-APRX algorithms (including MWM), from the proof of theorem 1, the following always holds:

$$E[V(Q(t+1)) - V(Q(t))|Q(t)] \leq -2\delta W^*(t) + f(W^*(t)) + 2N^2$$

where recall that $W^\star(t)$ is the weight of the MWM schedule. Though the methods used in the rest of the paper apply to all sub-linear functions $f(\cdot)$, we will restrict the analysis in the subsequent sections to the special case where $f(\cdot) = C$, a constant. However later we will generalise the result for any sub-linear function $f(\cdot)$. We would like to note that, in the rest of the section, $\|\cdot\|$ denotes the 1-norm of vector unless specified otherwise, that is, for $Q(t)$, $\|Q(t)\| = \sum_{i,j} Q_{ij}(t)$.

For the rest of the section, we will consider a 1-APRX algorithm of MWM, which we denote as $B$. Let $W^B(t)$ denote the weight of the schedule obtained by algorithm $B$ at time $t$. By the approximation property for $B$,

$$W^B(t) \geq W^\star(t) - C_b, \quad \forall t$$

where, $C_b$ is a constant, representing algorithm $B$'s approximation distance from MWM.

For algorithm $B$, from the proof of theorem 1, we obtain,

$$E[V(Q(t+1)) - V(Q(t))|Q(t)] \leq -2\delta W^* + C_b + 2N^2 \quad (2)$$

Note that, now onwards, unless stated explicitly, $W^\star$ denotes the weight of the MWM schedule at the time in consideration. For example, $W^\star$ denotes $W^\star(t)$ in (2).

We would revisit the proof of Theorem 1 to obtain tighter bounds on the sum of the average queue length. In (1), we bounded the $Q_{ij}$ by the $\max\{[Q_{ij}(t) - S_{ij}(t)] + A_{ij}(t+1), 1\}$. This results in a constant $2N^2$ in (2) above. Instead, if we ignore the reflection condition of the queue and simply take,

$$Q_{ij}(t+1) = Q_{ij}(t) - S_{ij}(t) + A_{ij}(t+1)$$

then we obtain,

$$\begin{aligned} V(Q(t+1)) - V(Q(t)) &= \sum_{i,j} Q_{ij}(t+1)^2 - Q_{ij}(t)^2 \\ &= \sum_{i,j} (Q_{ij}(t) - S_{ij}(t) + A_{ij}(t+1))^2 \\ &\quad - Q_{ij}(t)^2 \\ &= \sum_{i,j} (A_{ij}(t+1) - S_{ij}(t))^2 + \\ &\quad 2Q_{ij}(t)(A_{ij}(t) - S_{ij}(t+1)) \end{aligned} \quad (3)$$

Before proceeding further, one might object regarding the use of such "unreflected" version of queueing evolution. We would like to note that the "reflection" condition becomes active only when $Q_{ij}(t) = 0$. This trivially gives a bound on $Q_{ij}(t+1)$ of 1. Which in turn, as we will see later can give bound, $E[\|Q(t+1)\|] \leq N^2$, if the "reflection" condition is active on all the $Q_{ij}(t)$. The "unreflected" version, as we will see later, will give bound, say $\tilde{b}$ for $E[\|Q(t+1)\|]$. Thus, the bound on $E[\|Q(t+1)\|]$ is $\max\{\tilde{b}, N^2\}$. Later we will see that, in general, $\tilde{b} \geq N^2$, and hence it is the important bound. In the rest of the paper, henceforth, we will not consider the "reflection" condition.

Using argument similar to used in proof of Theorem 1, and (3), for algorithm $B$, we obtain,

$$\begin{aligned} &E[V(Q(t+1)) - V(Q(t))|Q(t)] \leq -2\delta W^\star(t) \\ &\quad + C_b + \sum_{i,j} E[(A_{ij}(t+1) - S_{ij}(t))^2 | Q(t)] \\ &= -2\delta W^\star(t) + C_b + \sum_{i,j} E[(A_{ij}(t+1) - S_{ij}(t))^2] \\ &= -2\delta W^\star(t) + C_b + \sum_{i,j} E[A_{ij}(t+1)^2] + \\ &\quad \sum_{i,j} E[S_{ij}(t)^2] - 2E[A_{ij}(t+1)S_{ij}(t)] \end{aligned} \quad (4)$$

Before proceeding further, we would like to evaluate the exact values of many of the terms in (4).

First, consider the term $\delta$. As used in proof of Theorem 1, since arrival matrix $\Lambda$ is strictly doubly sub-stochastic, we can write,

$$\Lambda \leq \sum_k \gamma_k \Pi_k$$

where, $\gamma_k \geq 0, \sum_k \gamma_k < 1$. We would like to remind to the reader that, in above equation, the $\leq$ sign means the term-by-term domination of one matrix by the other. Further it is possible to get a collection of $\gamma_k$ such that $\sum_k \gamma_k = \max_i \sum_j \lambda_{ij}$. (refer to [14] for the procedure to obtain such). To explain this, consider the following example:

**Example 1.** *Consider the case of uniform traffic with $\lambda_{ij} = \lambda = \rho/N, \forall i, j$. Consider the $N$ disjoint permutations: $\Pi^k = [\Pi^k_{ij}]$, such that, $\Pi^k_{ij} = 1$ iff $j = (i+k) \mod N$, for $k = 0, \ldots, N-1$. Then $\Lambda = [\lambda] = \sum_{k=0}^{N-1} \gamma_k \Pi^k$, where $\gamma_k = \lambda = \rho/N, \forall k$.*

From the proof of Theorem 1, it is clear that $\delta = 1 - \sum_k \gamma_k$, and hence

$$\begin{aligned} \delta &= 1 - \sum_k \gamma_k \\ &= 1 - \max_i \sum_j \lambda_{ij} \end{aligned} \quad (5)$$

Now consider the other terms in (4). Since $A_{ij}(t)$ is Bernoulli i.i.d. with mean probability $\lambda_{ij}$ of being 1, we have

$$\begin{aligned} E[A_{ij}(t)] &= \lambda_{ij}, \\ E[A_{ij}(t)^2] &= \lambda_{ij}, \end{aligned} \quad (6)$$

Under the strongly stable algorithm $B$, the switch state becomes a discrete time, irreducible, aperiodic Markov chain which has a stationary distribution. Hence for large enough $t$,

$$\begin{aligned} E[S_{ij}(t)] &= E[A_{ij}(t)] = \lambda_{ij} \\ E[S_{ij}(t)^2] &= E[A_{ij}(t)^2] = \lambda_{ij}, \end{aligned} \quad (7)$$

Further, since $S(t)$ depends on arrivals till time $t$, $S(t)$ is independent of $A(t+1)$ under i.i.d. assumption on input traffic. Using this, we obtain,

$$\begin{aligned} E[S_{ij}(t)A_{ij}(t+1)] &= E[S_{ij}(t)]E[A_{ij}(t+1)] \\ &= \lambda_{ij}^2 \end{aligned} \quad (8)$$

From (6)-(8), we obtain

$$\begin{aligned} &\sum_{ij} E[A_{ij}(t+1)^2] + E[S_{ij}(t)^2] - 2E[A_{ij}(t+1)S_{ij}(t)] \\ &\quad = \sum_{ij} (\lambda_{ij} + \lambda_{ij} - 2\lambda_{ij}^2) \\ &\quad = 2\tilde{\Lambda}_1 - 2\tilde{\Lambda}_2 \end{aligned} \quad (9)$$

where, $\tilde{\Lambda}_1 = \sum_{ij} \lambda_{ij}$ and $\tilde{\Lambda}_2 = \sum_{ij} \lambda_{ij}^2$. We denote, $\tilde{\Lambda} \triangleq (\tilde{\Lambda}_1 - \tilde{\Lambda}_2)$. Observe that, $W^\star(t) \geq \frac{1}{N}\sum_{ij} Q_{ij}(t) = \frac{1}{N}\|Q(t)\|_1$. To see why this is true, suppose the opposite is true, that is $\|Q(t)\|_1 > NW^\star(t)$. Consider the $N$ disjoint permutations $\Pi^k, 0 \leq k \leq N-1$, we used in the Example 1. Note that $\sum_k \Pi^k = [1]$. The weight of the schedule corresponding to these permutations is $W_{\Pi^k} = \langle \Pi^k, Q(t) \rangle$, and hence,

$$\begin{aligned} \sum_k W_{\Pi^k} &= \sum_k \langle \Pi^k, Q(t) \rangle \\ &= \langle \sum_k \Pi^k, Q(t) \rangle \\ &= \langle [1], Q(t) \rangle \\ &= \|Q(t)\|_1 > NW^\star(t) \end{aligned}$$

Hence, by pigeon-hole principle, we must have one $k$ such that $W_{\Pi^k} > W^\star$, which is a contradiction. Hence, $W^\star(t) \geq \frac{\|Q(t)\|_1}{N}$.

From (4), above discussion and replacing $\epsilon = \frac{\delta}{N}$, we obtain,

$$\begin{aligned} E[V(Q(t+1)) - V(Q(t))|Q(t)] &\leq -2\epsilon\|Q(t)\|_1 + \\ &\quad C_b + 2\tilde{\Lambda} \end{aligned} \quad (10)$$

Now consider the following,

$$\begin{aligned} E[V(Q(t+1)] &= E[V(Q(t+1)) - V(Q(t)) \\ &\quad + V(Q(t))] \\ &= E[E[V(Q(t+1)) - V(Q(t))|Q(t)]] \\ &\quad + E[V(Q(t))], \\ &\leq -2\epsilon E[\|Q(t)\|_1] + C_b \\ &\quad + 2\tilde{\Lambda} + E[V(Q(t))] \end{aligned} \quad (11)$$

Now summing up (11) from $t = 0$ to $t = T-1$, we obtain,

$$\begin{aligned} E[V(Q(T))] &\leq T(C_b + 2\tilde{\Lambda}) + E[V(Q(0))] \\ &\quad -2\epsilon \sum_{t=0}^{T-1} E[\|Q(t)\|_1] \end{aligned}$$

Assuming, we start with empty system, $E[V(Q(0))] = 0$. Thus, for any $T$, we obtain,

$$\begin{aligned} \frac{1}{T}\sum_{t=0}^{T-1} E[\|Q(t)\|_1] &\leq \frac{1}{2\epsilon}(C_b + 2\tilde{\Lambda}) \\ &\quad - \left(\frac{1}{T}\right) E[V(Q(T))] \end{aligned} \quad (12)$$

Since $V(\cdot) \geq 0$, we obtain from (12),

$$\frac{1}{T}\sum_{t=0}^{T-1} E[\|Q(t)\|_1] \leq \frac{1}{2\epsilon}(C_b + 2\tilde{\Lambda}) \quad (13)$$

Under the i.i.d. arrival process, the switch state $Q(t)$ is a discrete time Markov chain. It is an irreducible aperiodic Markov chain. Hence it is ergodic, that is, the left term in (13) converges to expected value of $\|Q(t)\|_1$ under equilibrium distribution. Hence from (13) we obtain,

$$\begin{aligned} \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} E[\|Q(t)\|_1] &= \lim_{T\to\infty} E[\|Q(T)\|_1] \\ &\leq \frac{1}{2\epsilon}(C_b + 2\tilde{\Lambda}) \end{aligned} \quad (14)$$

Using this, we obtain the following theorem.

**Theorem 3.** *Let $B$ be a scheduling algorithm. The weight of schedule obtained by $B$ at time $t$ is denoted by $W^B(t)$ and let the weight of MWM schedule for the same switch state at time $t$ be $W^\star(t)$. For any time $t$, $W^B(t) \geq W^\star(t) - C_b$. Then under admissible Bernoulli i.i.d. traffic with the arrival rate matrix $\Lambda = [\lambda_{ij}]$, the average of the sum of all the queue lengths in stationarity is bounded above as*

$$\lim_{t\to\infty} E[\|Q(t)\|_1] \leq \frac{N\tilde{\Lambda}}{\delta} + \frac{NC_b}{2\delta}$$

*where, $\delta = 1 - \max_i\{\sum_j \lambda_j\}$ and $\tilde{\Lambda} = [\sum_{ij}(\lambda_{ij} - \lambda_{ij}^2)]$.*

For MWM, we have $C_b = 0$, which means that the bounds are

$$E[\|Q(t)\|_1] \leq \frac{N\tilde{\Lambda}}{\delta}$$

For uniform traffic, we have $\lambda_{ij} = \lambda = \rho/N, \forall i, j$. Hence,

$$\tilde{\Lambda} = \frac{\rho}{N}\left(1 - \frac{\rho}{N}\right) N^2$$
$$= N\rho\left(1 - \frac{\rho}{N}\right)$$

This gives,

$$E[\|Q(t)\|_1] \leq N^2 \frac{\rho}{(1-\rho)}\left(1 - \frac{\rho}{N}\right)$$

These bounds are exactly same as the ones obtained in [5].

Note that, from Theorem 3, the important implication is that, the bound of approximate algorithms on average queue length are proportional to their "approximation distance" from maximum weight matching (MWM). The simulation study shows that this qualitative statement on bounds is actually tight. Thus, the approximation distance of a scheduling algorithm from maximum weight matching gives a metric to differentiate performance of algorithms with respect to each other.

For a general 1-APRX algorithm $B$, with property

$$W^B(t) \geq W^\star(t) - f(W^\star(t))$$

where, $f(\cdot)$ a sublinear function. We assume that $f$ is non-decreasing function. We also know that

$$W^\star(t) \leq \|Q(t)\|_1$$

Hence, we have, $f(W^\star(t)) \leq f(\|Q(t)\|_1)$ which gives

$$W^B(t) \geq W^\star(t) - f(\|Q(t)\|_1)$$

We assume that function $f$ behaves "nicely", so that by ergodicity of $Q(t)$, we obtain,

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} f(\|Q(t)\|_1) = \lim_{T\to\infty} E[f(\|Q(T)\|_1)]$$

Putting the above discussion together, the bound of Theorem 3, for such an algorithm $B$ becomes,

$$\lim_{t\to\infty} E[\|Q(t)\|_1] \leq \frac{N\tilde{\Lambda}}{\delta} + \frac{N}{2\delta}\lim_{t\to\infty} E[f(\|Q(t)\|_1)]$$

## III. Practical 1-APRX Algorithms

In this section, we consider two 1-APRX of MWM algorithm, which are implementable versions of the MWM algorithm.

*Pipelined-MWM*

This algorithm can be described as follows: The MWM schedule is computed over more than one time-slot. This computation is done in a pipelined manner. Let $K$ be the depth of pipeline. Hence, the computation of the MWM schedule with respect to switch state at time $t$ would get over at time $t + K$. Thus at any given time, the MWM schedule used is $K$ time slots old. We denote this algorithm as *pMWM*.

The reason we chose this algorithm is because of its compatibility with hardware. We know that hardware lends itself well to pipelining. We would therefore like to understand what would happen if we were to implement a pipelined version of MWM.

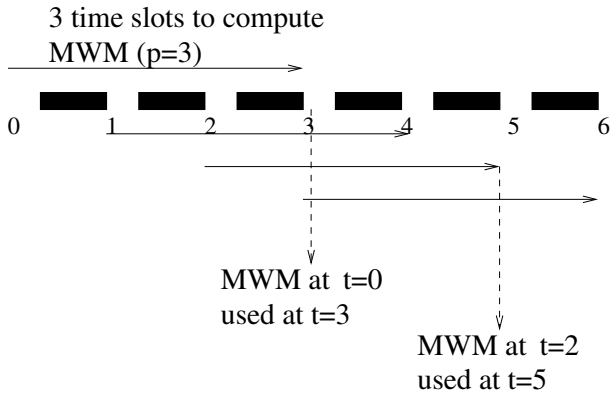


Fig. 2. Pictorial description of pMWM with pipelining p=3.

The figure 2, explains the way *pMWM* works.

**Theorem 4.** *Let $W_p$ denote the weight of schedule obtained by pMWM, and $W^*$ denote the weight of MWM schedule for the same switch state. Then,*

$$W_p \geq W^* - 2KN$$

*Proof.* The weight of a schedule can not increase by more than $N$ between consecutive time slots and can not decrease by more than $N$ between two consecutive time slots. Let $S^\star(s)$ denote the MWM schedule at time $s$. Consider any time $t$. The schedule used by *pMWM*, say $S^p(t)$ is MWM schedule with respect to switch state at time $t - K$. Thus, the weight of $S^p(t)$ at time $t - K$, $\langle S^p(t), Q(t-K)\rangle$ is at most $KN$ more than its weight, $\langle S^p(t), Q(t)\rangle$ at time $t$. Also, the weight of $S^\star(t)$ is at most $KN$ more than the weight of $S^p(t)$ at time $t - K$. Putting this together, we obtain that the weight $W^p(t)$ of the schedule used by *pMWM* is greater than $W^\star(t) - 2KN$. □

Thus pMWM is a 1-APRX algorithm.

*Bursty MWM*

This is another variant of MWM. In this version, every $K^{th}$ time slot, a MWM schedule is computed with respect to the switch state at that time. This schedule is used repeatedly for the next $K$ time slots. We call this a *Bursty MWM*, and denote by *bMWM*.

The reason we chose this algorithm is because it is a good example of our class of approximate algorithms where the weight of the matching keeps changing from being exactly equal to the weight of MWM to that of the weight that is a constant $2KN$ away from the weight of MWM.

The figure 3 explain the bMWM algorithm for $K = 3$.

**Theorem 5.** *Let $W_b$ denote the weight of schedule obtained by bMWM, and $W^*$ denote the weight of MWM schedule for the same switch state. Then,*

$$W_b \geq W^* - 2KN$$

*Proof.* From the proof of Theorem 4,

$$W^*(t) \geq W^*(t-1) - N$$

Consider the $K^{th}$ time-cycle starting from say $t_0$, one of the time slots when a new MWM schedule has just been computed

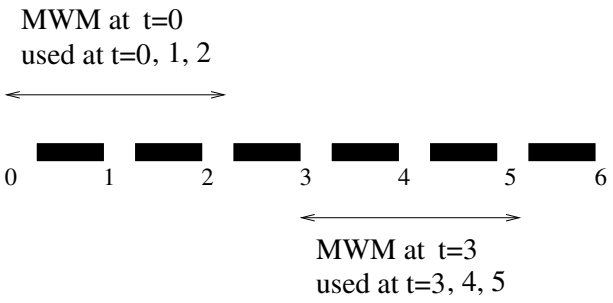


Fig. 3. Pictorial description of bMWM. Here the burst size is 3.

by the bMWM algorithm. For time $t \in [t_0, t_0+K-1]$, we keep this schedule fixed. The weight of this schedule can at most decrease by $N$ every time. Thus combining above discussion together, we obtain that for any time $t \in [t_0, t_0+K-1]$,

$$W_b \geq W^* - 2KN$$

□

### A. Simulation of Delays

We first explain the simulation settings:

**Switch**: Switch size: $N = 32$. Each VOQ can store upto 10,000 packets. Excess packets are dropped.

**Input Traffic**: All inputs are equally loaded on a normalized scale, and $\rho \in (0, 1)$ denotes the normalized load. The arrival process is Bernoulli i.i.d. The following load matrices are used.

*1. Uniform:* $\lambda_{ij} = \rho/N \ \forall i, j$. This is the most commonly used test traffic in the literature.

*2. Log-Diagonal:* This is a very skewed loading, in the sense that input $i$ has packets for different outputs with probabilities that differ exponentially. Thus $\lambda_{ij} = \frac{2^{j-i}}{sum_k(2^k)}\rho$. This traffic pattern is more difficult to schedule than uniform loading.

We measure the average cell delay for different algorithms to test the bounds obtained theoretically.

We let the simulation run until the estimate of the average delay reaches the relative width of confidence interval equal to 1% with probability $\geq 0.95$. The estimation of the confidence interval width is obtained using the *batch means* approach.

From Theorem 4 and 5, *pMWM* and *bMWM* are 1-APRX algorithms, with the approximation distance of $2KN$ for both of them. By Theorem 3, the bounds for different versions of *pMWM* and *bMWM* should increase proportional to the number of stages $K$, used by them. Thus, Theorem 3 suggests that the difference between bounds for different versions of *pMWM* and *bMWM* should be proportional to the difference between the number of stages among the versions. We shall see that the simulations verify this very well.

The figures 4 and 5 represents the average cell delay of pMWM, with different values of $K$, for different traffic load under uniform traffic and diagonal traffic respectively. Observe that, for any of the loading, under both traffic patterns, the difference between average delays are proportional to the difference in the values of $K$. For example, in figure 4 for load 0.8 the difference between average queue length for $K = 8$ and $K = 16$ is half the difference between average queue length for $K = 16$ and $K = 32$ (similar statement is true for $16, 32$ and $64$ too.). This verifies that the increase in the average queue lengths (or delays) of 1-APRX algorithm are proportional to the approximation distance of the algorithm.

The figures 6 and 7 show that similar behavior is observed for the *bMWM* algorithm too, which confirms that the bounds we obtained are tight in the qualitative sense.

We would like to note that, the analytical bounds obtained in Theorem 3 are an upper bound and not as tight as the simulational results. Hence we have not plotted analytical bounds to compare them with the simulation bounds. But the main message of this paper is to convey the *qualitative* relationship between the performance in terms of delay and "approximation distance", and it is being verified by simulation appropriately. Thus, in the qualitative sense our analytical bounds are tight and agree with the simulation results.

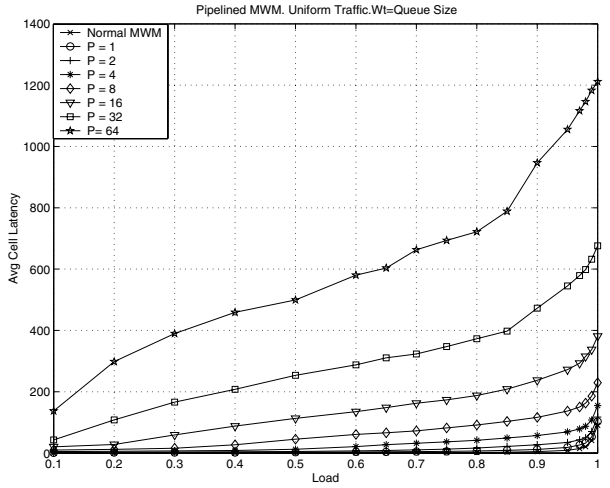


Fig. 4. Average cell-delay v/s Load for different levels of pipelining(P=level of pipelining) for pMWM under uniform traffic.

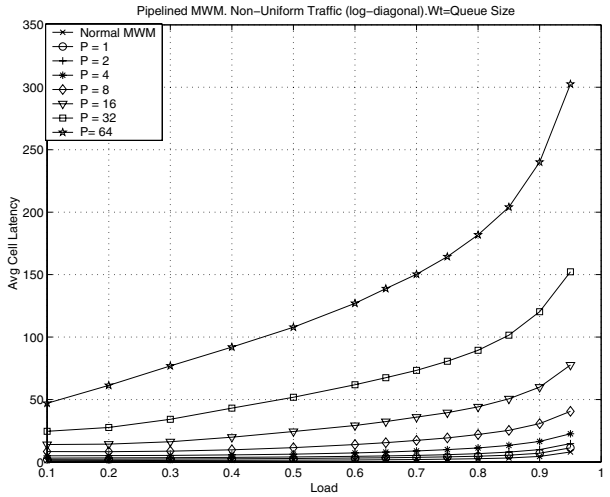


Fig. 5. Average cell-delay Latency v/s Load for different levels of pipelining under log-diagonal traffic

## IV. Conclusions

In this paper, we studied the throughput and delay properties of scheduling algorithms for IQ switches which are 1-APRX

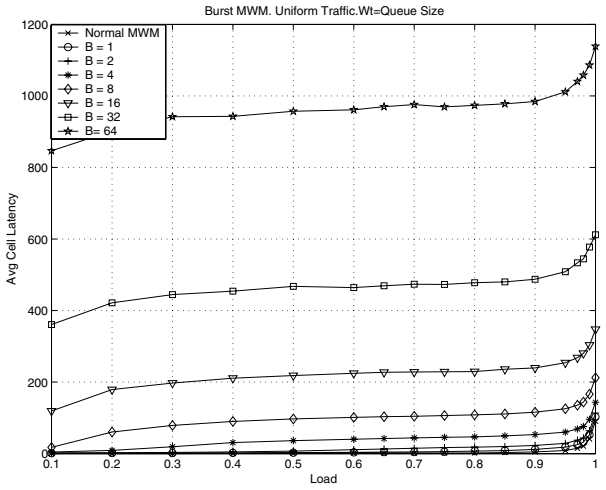


Fig. 6. Average cell-delay v/s Load for different levels of burst under uniform traffic for bMWM

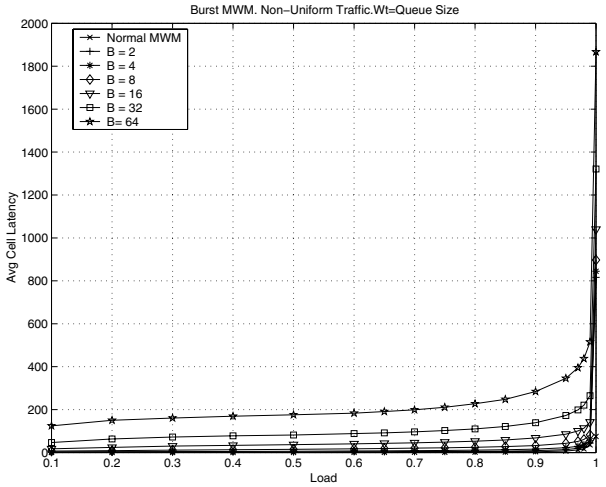


Fig. 7. Average cell-delay v/s Load for different levels of burst under diagonal traffic for bMWM

to MWM. We first showed that all such algorithms are stable. We showed that the delay bounds, an important performance metric for scheduling algorithms, are directly proportional to the difference in weight of the MWM schedule and weight of the schedule of the 1-APRX algorithm. Simulations confirmed that this qualitative relation actually holds for such algorithms. Thus, we have provided a theoretical metric that can help differentiate the performance of all stable algorithms.

We have also provided novel heuristic tighter bounds on the average queue length(delay) for MWM under uniform i.i.d. arrival traffic. These bounds can be extended for non-i.i.d. arrival traffic as well but not for non-uniform traffic.

## REFERENCES


[1] Karol M., Hluchyj M., Morgan S., "Input versus output queuing on a space division switch", *IEEE Trans. on Communications*, vol. 35, n. 12, Dec. 1987, pp. 1347-1356

[2] McKeown N., Anantharan V., Walrand J., "Achieving 100% throughput in an input-queued switch" *IEEE Infocom '96*, vol. 1,San Francisco, Mar. 1996, pp. 296-302

[3] Tassiulas L., Ephremides. A., "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks". *IEEE Trans. on Automatic Control*, vol. 37, n. 12, Dec. 1992, pp. 1936-1948.

[4] Dai J., Prabhakar B., "The throughput of data switches with and without speedup", *IEEE INFOCOM 2000*, vol. 2, Tel Aviv, Mar. 2000, pp. 556-564

[5] Leonardi E., Mellia M., Neri F., Ajmone Marsan M., "Bounds on Average Delays and Queue Size Averages and Variances in Input-Queued Cell-Based Switches", *IEEE INFOCOM 2001*, Alaska, April 2001, pp.1095-1103.

[6] McKeown N., "iSLIP: a scheduling algorithm for input-queued switches", *IEEE Trans. on Networking*, vol. 7, n. 2, Apr. 1999, pp. 188-201

[7] McKeown N., "Scheduling algorithms for input-queued cell switches", *Ph.D. Thesis*, Un. of California at Berkeley, 1995

[8] Ajmone Marsan M., Bianco A., Leonardi E., Milia L., "RPA: a flexible scheduling algorithm for input buffered switches", *IEEE Trans. on Communications*, vol. 47, n. 12, Dec. 1999, pp. 1921-33

[9] Duan H., Lockwood J.W., Kang S.M., Will J.D., "A high performance OC12/OC48 queue design prototype for input buffered ATM switches", *IEEE INFOCOM'97*, vol. 1, Kobe, 1997, pp. 20-28.

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

[11] Shah D., Giaccone P., Prabhakar B., "An efficient Randomized Algorithm for Input-Queued Switch scheduling", *To appear in Hot Interconnects*, Stanford, August 2001.

[12] Shah D., "Stable Algorithms for Input Queued Switches", *Proceedings of the Allerton Conference on Communication, Control, and Computing, Urbana, Illinois, 2001*..

[13] Gross D., Harris C.M., "Fundamentals of Queueing Theory", *Wiley Series*.

[14] Chang C.-S., Lee D.S., Jou Y.S., "Load balancing Birkhoff-von Neumann switches Part I: one stage buffering", *IEEE HPSR Conference*, Dallas, May 2001.

[15] Hajek B., "Hitting and occupation time bounds implied by drift analysis with applications", *Advances in Applied Probability*, vol. 14,pp. 502-525, September 1982.