# Distributed scheduling policies for networks of switches with a configuration overhead

Claus Bauer

Dolby Laboratories, San Francisco, CA, 94103, USA,
cb@dolby.com

**Abstract.** Optical switching cores are fast gaining importance for deployment in internet switches/routers. The reconfiguration of these switches requires a large timely switching overhead. The design of efficient algorithms that take into account the configuration overhead has been widely researched. However, all previous research solely focuses on switching algorithms that optimize the performance features of a single switch with configuration overhead. This paper is the first which designs classes of switching policies for a network of switches with a configuration overhead that guarantee the stability of the network. We also show that networks of switches with configuration overhead are stable if different classes of policies are deployed at different switches simultaneously.

## 1 Introduction

The introduction of new optical transmission technologies such as Dense Wavelength Division Multiplexing (DWDM) have dramatically increased the transmission capacity of optical fibers. As a consequence, there is a need for switches and routers that work at or above the speed of the high-speed optical links connecting them.
Today, most high-performance routers/switches use an electronic core that deploys a Virtual Output Queueing Scheme and a crossbar switching core. It is not expected that electronic switches will be able to meet the performance requirements imposed by future optical transmission capacities. Thus, optical switching cores have increasingly gained importance. At this time, it is neither possible to buffer packets in the optical domain nor to evaluate the packet header in the optical domain. Therefore, most researchers ([7], [12])) propose *hybrid* electronic-optical architectures: Packets that arrive on optical input links are converted into an electric signal. The header evaluation and the eventual buffering are performed in the electronic domain. In order to forward the packet through the optical switching core, the packet is reconverted into the optical domain and sent through the switch. If the switch uses output buffers, the packet is again reconverted into electronics at the output, buffered electronically, and reconverted into optics when it leaves the switch. Obviously, it is desirable to find ways to perform the header evaluation and buffering in the optical domain in order to save the numerous electronic-optical and optical-electronic conversions.
The MEMS technology is a favorite candidate for an optical switching core

([6],[13]). Compared to electronic switches, that reconfigure themselves in few nanoseconds or less, the reconfiguration time of a MEMS based switch is typically quoted as being between 1 and 10 ms. Depending on the link speed, these switches take up to 20,000 cell times to reconfigure. This long reconfiguration time introduces large packet delays and requires scheduling algorithms that take this configuration overhead into account and optimize the delay and loss characteristics of the resulting schedule.
Recently, various researchers have proposed different scheduling algorithms for switches for a configuration overhead. Two approaches can be distinguished: In the Timeslot Assignment based approach ([9], [10], [15]), incoming traffic is accumulated during a predefined accumulation period. During each cycle, the arriving traffic is buffered in an $N \times N$ traffic matrix. Using different algorithms, the traffic matrix is decomposed into permutation matrices that determine the configurations of the switch. The Single Scheduling approach ([10],[11]) can be considered as a slow version of scheduling algorithms for switches without configuration overhead. Similar to packet-based based scheduling ([14]), a schedule is generated and maintained for several timeslots. Most approaches require the switching core to work at a speedup $S > 1$ compared to the linkspeed.
Previous research on scheduling algorithms for switches with configuration overhead has investigated scheduling algorithms that optimize the performance features stability and delay for a single switch. So far, no research on scheduling algorithms that stabilize networks of switches with a configuration overhead has been performed. For switches without configuration overhead, it has been shown in [3] for the example of a maximum weight matching algorithm ([7]) that scheduling algorithms that guarantee the stability of individual switches do not necessarily guarantee the stability of networks of switches. Following the argument in [3], it can be shown that the Single Scheduling $LQF + holding$ algorithm proposed in [10] can lead to instabilities in a network of switches with configuration overhead.
For switches without configuration overhead, in [2] and [3], local scheduling algorithms that stabilize networks of switches, but require signaling traffic between adjacent switches have been proposed. Only recently, in [1] a scheduling algorithm has been proposed that does not require signaling traffic, but stabilizes a network of switches. This algorithm requires non-local information to be transported in the packet header.
This paper is the first effort to investigate local scheduling algorithms for networks of input-queued switches with a configuration overhead that stabilize the entire network.
The rest of the paper is organized as follows. In the next section, we develop a model for a network of switches. In section 3, we define local scheduling policies and prove the stability of networks that deploy any of those policies at all switches. In section 4, we prove that networks of switches that deploy different classes of these policies simultaneously at different switches of the network are stable as well. We conclude in section 5.

## 2 Terminology and Model

### 2.1 Model of a network of queues

In this section, we follow an approach in [1] to describe our model of a queueing system. We assume a system of $J$ physical queues $\tilde{q}^j$, $1 \leq j \leq J$ of infinite capacity. Each physical queue consists of one or more logical queues, where each logical queue corresponds to a certain class of customers within the physical queue. Whenever a packet moves from one physical queue to another, it changes class and therefore also changes logical queue. We denote a logical queue by $q^k$, $1 \leq k \leq K$, where $K \geq J$. A packet enters the network via an edge switch, travels through a number of switches and leaves the network via another edge switch. We define a function $L(k) = j$ that defines the physical queue $\tilde{q}^j$ at which packets belonging to the logical queue $q^k$ are buffered. The inverse function $L^{-1}(j)$ returns the logical queues $q^k$ that belong to the physical queue $\tilde{q}^j$.

Throughout this paper, the time $t$ is described via a discrete, slotted time model. Packets are supposed to be of fixed size and a timeslot is the time needed by a packet to arrive completely at an input link.

We define a row vector $X_n = (x_n^1, ..., x_n^K)$, where the $k$-th vector $x_n^k$ represents the number of packets buffered in the logical queue $q^k$ in the $n$-th timeslot. We define $E_n = (e_n^1, ..., e_n^K)$, where $e_n^k$ equals the number of arrivals at the logical queue $q^k$ in the $n$-th timeslot. Analogously, we define $D_n = (d_n^1, ..., d_n^K)$, where $d_n^k$ expresses the number of departed packets from $q^k$ in the $n$-th timeslot. Thus, we can describe the dynamics of the system as follows:

$$X_{n+1} = X_n + E_n - D_n. \tag{1}$$

Packets that arrive at a logical queue $q^k$ either arrive from outside the system or are forwarded from a queue within the system. Thus, we can write:

$$E_n = A_n + T_n,$$

where $A_n = (a_n^1, ..., a_n^K)$ denotes the arrivals from outside the system and $T_n = (t_n^1, ..., t_n^K)$ denotes the arrivals from inside the system.
We define a routing matrix $R = [r_{i,j}]$, $1 \leq i, j \leq K$, where $r_{i,j}$ is the fraction of customers that depart from the logical queue $q^i$ and are destined for the logical queue $q^j$. Assuming a deterministic routing policy, there holds, $r_{i,j} \in \{0, 1\}$, $\sum\limits_{1 \leq i < K} r_{i,j} < 1$, $\sum\limits_{1 \leq j \leq K} r_{i,j} \leq 1$. We set $r_{i,j} \neq 0$, if $q^j$ follows $q^i$ along the route.
Noting that $T_n = D_n R$ and writing $I$ for the identity diagonal matrix, we find

$$X_{n+1} = X_n + A_n - D_n(I - R). \tag{2}$$

We assume that the external arrival processes are stationary and satisfy the Strong Law of Large Numbers. Thus,

$$\lim_{n \to \infty} \frac{\sum\limits_{i=1}^{n} A_i}{n} = \Lambda \qquad \text{w.p.1}, \tag{3}$$

where $E[A_n] = \Lambda = (\lambda^1, .., \lambda_K)\,, \forall n \geq 1$[1]. Noting that $(I-R)^{-1} = I+R+R^2+...$, we find that the average workload $W = (w^1, ..., w^K)$ at the logical queues $q^k$ is given by $W = \Lambda(I-R)^{-1}$.

Finally, we give a stability criteria for a network of queues as proposed in [2].

**Definition 1:** A system of queues is rate stable if

$$\lim_{n \longrightarrow \infty} \frac{X_n}{n} = \lim_{n \longrightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} (E_i - D_i) = 0 \qquad \text{w.p.1.}$$

A necessary condition for the rate stability of a system of queues is that the average number of packets that arrive at any physical queue $\tilde{q}^j$ during a timeslot is less than 1. We formalize this criteria as follows:

**Definition 2:** For a vector $Z \in \mathbb{R}^K$ , $Z = (z^1, .., z^K)$, and the function $L^{-1}(k)$ as defined in this subsection, we set:

$$||Z||_{maxL} = \max_{j=1,..,J} \left\{ \sum_{k \in L^{-1}(j)} z^k \right\}. \tag{4}$$

The necessary condition for rate stability can now be formalized as follows:

$$||W||_{maxL} < 1. \tag{5}$$

## 2.2 Model of a network of switches

In this section, we apply the terminology of the previous section to a network of switches. We assume that the switching core is an $N \times N$ input-queued or combined input/output-queued (IQ/CIOQ) switch that deploys a Virtual Output Queue buffer structure ([7]. A network of IQ/CIOQ switches can be conceived as a queueing system as defined in the previous section where the virtual output queues are considered as the physical queues. In this model, we neglect the output queues of the switches because instability can only occur at the Virtual Output Queues (see [1]).

We say that packets that enter the network via the input of a given switch and leave the network via the output of a given switch belong to the same flow. Packets belonging to the same flow travel through the same sequence of physical queues and are mapped to the same logical queues at each physical queue, i.e., a flow can be mapped biunivocally to a series of logical queues.

We assume that each logical queue behaves as a FIFO queue and assume a *per-flow* scheduling scheme. It has been shown in [1] how stability results for *per-flow* scheduling schemes can be used to design less complex and stable *per-virtual output queue* schemes.

The network consists of $B$ switches and each switch has $N_b$, $1 \leq b \leq B$, inputs and outputs. If the total number of flows in the system is $C$, we do not have more than $N_b^2$ physical queues and $CN_b^2$ logical queues at switch $b$. We can model

[1] Throughout the paper, we abbreviate "with probability 1" by "w.p.1."

the whole network of switches as a system of $\sum\limits_{1\leq b\leq B} CN_b^2$ logical queues. For the sake of simplicity, we suppose that $N_b = N$, $1 \leq b \leq B$ and set $K = CN^2B$. Finally, we define $Q_I(b,i)$ as the set of indexes corresponding to the logical queues at the $i$-th input of the $b$-switch. Analogously, $Q_O(b,i)$ denotes the set of indexes corresponding to the logical queues directed to the $i$-th output of the $b$-switch. We use these definitions to adapt the norm $||Z||_{maxL}$ to a network of switches:

**Definition 3:** Given a vector $Z \in \mathbb{R}^K, Z = \{z^k, k = CN^2b + CNi + Cj + l,\ 0 \leq b < B,\ 0 \leq i,j < N, 0 \leq l < C$, the norm $||Z||_{IO}||$ is defined as follows:

$$||Z||_{IO} = \max_{\substack{b=1,..,B\\ i=1,..,N}} \left\{ \sum_{m\in Q_I(b,i)} |z^m|\,, \sum_{m\in Q_O(b,i)} |z^m| \right\}.$$

As we assume a deterministic routing policy, the necessary condition for rate stability given in (5) can be written for a network of switches as follows:

**Definition 4:** For a network of IQ/CIOQ switches, a traffic and routing pattern $W$ is admissible if and only if:

$$||W||_{IO} = ||\Lambda(I-R)^{-1}|| < 1. \tag{6}$$

In the rest of this paper, we will only consider traffic and routing patterns that satisfy the condition (6). We will say that a network which is rate stable under condition (6) achieves 100% throughput.

## 3 Local scheduling policies

### 3.1 Weight function

All scheduling policies introduced in this paper are matching policies. Any matching policy is defined relative to a specific weight. For the definition of the weights, we will make use of a family of real positive functions $f_k(x) : \mathbb{N} \to \mathbb{R}$, $1 \leq k \leq K$, that satisfy the following property:

$$\lim_{n\to\infty} \frac{f_k(n)}{n} = \frac{1}{w^k} \qquad \text{w.p.1.} \tag{7}$$

We define $\overline{d}^k(n) = \sum\limits_{m\leq n} d_m^k$ as the cumulative number of services at queue $q^k$ up to time $n$. We define the weights of the queues $q^k$s at time $n$ by

$$\phi_n^k = n - f_k(\overline{d}^k(n)) \quad \Phi_n = (\phi_n^1, .., \phi_n^K). \tag{8}$$

In [2], an example for $f_k(n)$ is given. The cumulative function of external arrivals for the logical queue $q^k$ is given by $\overline{a}^k(n) = \sum\limits_{m\leq n} a_m^k$. The inverse function $[\overline{a}^k]^{-1}(p)$ maps the packet number $p$ to the arrival slot. Setting $f_k(p) = [\overline{a}^k]^{-1}(p)$, the weight $\phi_n^k = n - [\overline{a}^k]^{-1}(p)$ denotes the time the packet has already spent in the network. At its departure time $n$, the age of the $p$-th packet is $n - [\overline{a}^k]^{-1}(\overline{d}_n^k)$.

### 3.2 The fluid methodology

The main proof of this section will make use of the fluid methodology as introduced in [4], [5]. As in [1], we consider an extension of the fluid model to a network of switches. First, we define three vectors: $X(t) = (X_{1,1}(t), ..., X_{N,N}(t))$ denotes the number of packets in the $VOQs$ at time $t$, $D = (D_{1,1}(t), ..., D_{N,N}(t))$ denotes the number of packet departures from the $VOQs$ until time $t$ and $A = (A_{1,1}(t), ..., A_{N,N}(t))$ denotes the number of packet arrivals at the $VOQs$ until time $t$. We define $\Pi = \{\pi\}$ as the set of all possible network-wide matchings and denote a specific scheduling algorithm by $\mathcal{S}$. For all $\pi \in \Pi$, we denote by $T_\pi^{\mathcal{S}}(t)$ the cumulative amount of time that the matching $\pi$ has been used up to time $t$ by the algorithm $\mathcal{S}$. Obviously, $T_\pi^{\mathcal{S}}(0) = 0 \ \forall \pi \in \Pi$. Using (2), we obtain the fluid equations of the system as follows:

$$X(t) = X(0) + \Lambda t - D(t)(I - R), \tag{9}$$

$$D(t) = \sum_{\pi \in \Pi} \pi T_\pi^{\mathcal{S}}(t), \tag{10}$$

$$\sum_{\pi \in \Pi} T_\pi^{\mathcal{S}}(t) = t. \tag{11}$$

The first two equations model the evolution of the logical queues, whereas the third counts the total number of departures from the $VOQs$. The third equation reflects the fact that in each timeslot, each input is connected to some output.

### 3.3 Maximum weight matching policies

In this section, we define a class of maximum weight matching policies that guarantee the stability of a network of IQ/CIOQ switches with configuration overhead. We introduce a set of functions $\mathcal{G}$ as follows:
**Definition** A real function $\mathcal{F}$ is said to belong to the set $\mathcal{G}$ if
a) $\dot{\mathcal{F}}(x)$ exists for all $x > 0$.
b) $\mathcal{F}$ and $\dot{\mathcal{F}}(x)$ are strictly monotonically increasing, non-negative and $\mathcal{F}(\prime) = \prime$, $\dot{\mathcal{F}}(0) = 0$.
c) $\dot{F}$ satisfies a Lipschitz condition: $\exists C_F \ s.t. \ |\dot{\mathcal{F}}(x) - \dot{\mathcal{F}}(y)| \le C_{\mathcal{F}}|x - y|, \ \forall x, y \in \mathbb{R}$.
Using the fluid methodology, we define a function $\Phi(t)$ based on the definition of the function $\Phi_n$ in (8). We set

$$\mathcal{F}(\Phi(t)) = \sum_{k=1}^{K} \mathcal{F}(\phi^k(t)).$$

We define $\Gamma = [\gamma^{(i,j)}]$ as the diagonal matrix with $\gamma^{(k,k)} = w^k$, and let $\Gamma^{-1}$ be the inverse of $\Gamma$. Further, we write the scalar product for two vectors $v_1$ and $v_2$ as $\langle v_1, v_2 \rangle = v_1 v_2^T$. Following an idea in [1], we first define a scheduling policy for input-queued switches without configuration overhead. For every function

$F \in \mathcal{G}$, we define a scheduling algorithm $MWM^{\mathcal{F}}$ as follows. At each time $t$, the scheduling algorithm $MWM^{\mathcal{F}}$ chooses the schedule $\pi^{\mathcal{F}}$ which is defined as:

$$\pi^{\mathcal{F}}(t) = \arg\max_{\pi} \left\{ \langle \pi, \dot{\mathcal{F}}(\phi(t)) \rangle \right\}. \tag{12}$$

For fixed $\mathcal{F}$, we denote the value of the matching achieved by $MWM^{\mathcal{F}}$ as $M(t) = \langle \pi^{\mathcal{F}}(t), \dot{\mathcal{F}}(\phi(t)) \rangle$. Using this terminology, we now define a class of scheduling policies $MWM_c^{\mathcal{F}}$ for switches with configuration overhead. As shown in fig. 1, a switch operates in cycles of constant length. Each cycle consists of a configu-

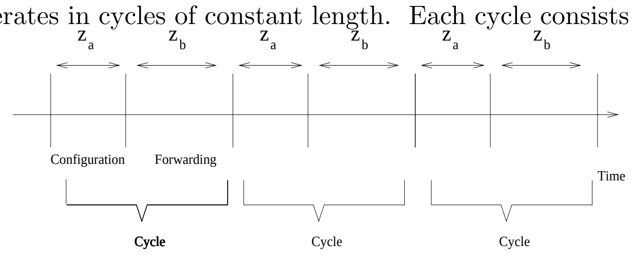

**Fig. 1.** Operation mode of the $MWM_c^{\mathcal{F}}$ algorithm

ration phase of length $z_a$-timeslots, during which the switch is reconfigured and no packets are forwarded, and a forwarding phase of length $z_b$ timelslots, during which the switch configuration remains unchanged and packets are forwarded. Scheduling decisions are made according to the policy $MWM^{\mathcal{F}}$ at instants $t_n$, where $t_{n+1} - t_n = z_a + z_b$. The chosen schedule $\pi^{\mathcal{F}}(t_n)$ is kept constant during the interval $[t_n + z_a, t_{n+1}[$. We set

$$M_c(t) = \langle \pi^{\mathcal{F}}(t_n), \dot{\mathcal{F}}(\phi(t)) \rangle \qquad \text{for } t \in [t_n, t_{n+1}[. \tag{13}$$

Now, we can formulate the main result of this section:

**Theorem 1:** *For any function $\mathcal{F} \in \mathcal{G}$, a network of IQ/CIOQ switches with configuration overhead that implements a $MWM_c^{\mathcal{F}}$ $-$policy, in which the weight $\phi_n^k$ of queue $q^k$ at time $n$ is defined as in (8) , and which deploys a speedup $S \geq \left\lceil \frac{z_a + z_b}{z_b} \right\rceil$ achieves 100% throughput.*

## 3.4 Proof of theorem 1

Before proving the theorem, we introduce the notion of a $MWM_m^{\mathcal{F}}$ scheduling policy for switches without configuration overhead. We define the $MWM_m^{\mathcal{F}}$ scheduling policy as the scheduling policy that applies an $m - timeslots\, old$ matching of the $MWM^{\mathcal{F}}$ scheduling policy to configure the switch: If we set $\pi_m^{\mathcal{F}}(t) = \pi^{\mathcal{F}}(t - m)$, then the $MWM_m^{\mathcal{F}}$ policy calculates the match: $M_m(t) = \langle \pi_m^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t)) \rangle$. We set $\Pi^{m,\mathcal{F}}(t) = \{\pi' : \langle \pi', \dot{\mathcal{F}}(\Phi(t)) \rangle = \max_{\pi} \langle \pi, \dot{\mathcal{F}}(\phi(t-m)) \rangle\}$, and follow an argument in [4] to derive from (11):

$$\sum_{\pi \in \Pi^{m,\mathcal{F}}(t)} \dot{T}_{\pi}^{MWM_m^{\mathcal{F}}}(t) = 1. \tag{14}$$

**Lemma 1**

$$M_m(t) \geq M(t) - C(W,m), \tag{15}$$

where $C(W,m)$ is a constant that depends on $m$ and the matrix $W$.
**Proof:** We note that by (7) $\lim_{t\to\infty} f_k(t) \to t/w^k$, and that $\overline{d}^k(t) \to \infty$ for $t \to \infty$. Thus

$$\phi^k(t) \to t - \frac{\overline{d}^k(t)}{w^k}, \tag{16}$$

and

$$|\phi^k(t+m) - \phi^k(t)| \leq m\left(1 + \frac{1}{w_k}\right) =: C(W,m), \tag{17}$$

From (17) and the Lipschitz condition of the function $\dot{\mathcal{F}}$, we obtain $\langle \pi_m^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t))\rangle \geq \langle \pi_m^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t-m))\rangle - KC_{\mathcal{F}}C(W,m)$, and $\langle \pi^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t-m))\rangle \geq \langle \pi^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t))\rangle - KC_{\mathcal{F}}C(W,m)$. Further, by (12) $\langle \pi_m^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t-m))\rangle \geq \langle \pi^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t-m))\rangle$. Combining the estimates, the lemma follows with $C(W,m) =: 2KC_{\mathcal{F}}C(W,m)$.
We note that by (13) and lemma 1,

$$M_c(t) \geq M(t) - C(W, z_a + z_b). \tag{18}$$

For technical reasons, we first prove theorem 1 for the case $z_a = 0$ and then show the general case $z_a > 0$. We define the Lyapunov function: $G(t) = \langle \mathbb{I}, \mathcal{F}_1(\Phi(t))\rangle$, where $\mathcal{F}_1(x) = \Gamma\mathcal{F}(x)$. By (16),

$$\dot{\Phi}(t) = \mathbb{I} - \dot{D}(t)\Gamma^{-1}. \tag{19}$$

We want to show that for an absolute constant $B > 0$ there is $\forall t \geq 0$,

$$||\Phi(t)||_1 \leq B, \tag{20}$$

where $||\ ||_1$ denotes the $L_1$ norm. Noting that $G(0) \geq 0$, we see that if

$$\frac{d}{dt}G(t) \leq 0 \tag{21}$$

$\forall t$ such that $\langle \mathbb{I}, \Phi(t)\rangle \geq K_0$ for a fixed $K_0 > 0$, then $\forall t \geq 0$, there holds $G(t) \leq \max_{\substack{L\in\mathbb{R}^K,\\ \langle \mathbb{I}, L\rangle \leq K_0}} \langle \nVdash, \dot{\mathcal{F}}(L)\rangle$, which in turn implies (20) for a certain $B > 0$. Thus, we derive (21), from (10), (14), (18), and (19):

$$\begin{aligned}
&\frac{d}{dt}G(t) = \langle 1, \dot{\mathcal{F}}_1(\Phi(t))\dot{\Phi}(t)\rangle \\
&= \langle 1, \Gamma\dot{\mathcal{F}}(\Phi(t))\rangle(I - \dot{D}(t)\Gamma^{-1}) = \langle 1, \dot{\mathcal{F}}(\Phi(t))(\Gamma - \dot{D}(t))\rangle \\
&= \langle \Gamma, \dot{\mathcal{F}}(\Phi(t)\rangle - \sum_{\pi\in\Pi^{z_a+z_b,\mathcal{F}}(t)} \dot{T}_\pi^{MWM_{z_a+z_b}^{\mathcal{F}}} \langle \pi_{z_a+z_b}(t), \mathcal{F}(\dot{\Phi(t)})\rangle \\
&\leq \langle \Gamma, \dot{\mathcal{F}}(\Phi(t)\rangle - \langle \pi^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t))\rangle + C(W, z_a + z_b).
\end{aligned} \tag{22}$$

As by (6),$||W||_{IO} < 1$, we argue as in [7] to find a $W_1$ satisfying $||W_1||_{IO} < 1$ and $W \leq (1-\epsilon)W_1$ for $\epsilon > 0$. We define $\varGamma_1$ based on $W_1$ analogously to $\varGamma$. We know from [7] that (12) implies $\langle \varGamma_1, \dot{\mathcal{F}}(\varPhi(t))\rangle < \langle \pi(t), \dot{\mathcal{F}}(\varPhi(t))\rangle$. Thus, by (22):

$$\begin{aligned}\frac{d}{dt}G(t) &\leq \langle \varGamma_1, \dot{\mathcal{F}}(\varPhi(t))\rangle - \langle \pi(t), \dot{\mathcal{F}}(\varPhi(t))\rangle - \epsilon\langle \varGamma_1, \dot{\mathcal{F}}(\varPhi(t)\rangle + C(W, z_a + z_b)\\ &\leq -\epsilon \min_{\substack{1\leq k\leq K\\ w^k>0}} w^k \max_{1\leq k\leq K} \dot{\mathcal{F}}(\phi^k(t)) + C(W, z_a + z_b).\end{aligned}$$

As $\dot{\mathcal{F}}$ was supposed to be non-negative and strictly monotonically increasing, (21) follows. The relations (16) and(20) implies that: $0 < t - \frac{\overline{d}^k(t)}{w^k} + C \leq B$. Whence, $\lim_{t\to\infty} \frac{\overline{d}^k(t)}{t} = w^k$, i.e., $\lim_{t\to\infty} \frac{D(t)}{t} = W$, w.p.1, which corresponds to the rate stability condition for $X(t)$.
Following an argument in [14], we now treat the general case $z_a > 0$. For any switch that operates with a speedup $S$ and has a configuration period of length $z_a = 0$, the equation (11) changes to

$$\sum_{\pi\in\varPi} T_\pi^{\mathcal{F}}(t) = St. \tag{23}$$

As for the $MWM_c^{\mathcal{F}}$ policy a fraction $\frac{z_a}{z_a+z_b}$ of the bandwidth is lost during the configuration phase, we adjust the RHS of (23) by a factor of $\frac{z_b}{z_a+z_b}$ and obtain the equality $\sum\limits_{\pi\in\varPi} T_\pi^{\mathcal{F}}(t) = S\frac{z_b}{z_a+z_b}t$. Differentiating this equation, we follow an argument in [4] as in the derivation of (14), and find that for $S > z_a + z_b/z_b$,

$$\sum_{\pi\in\varPi^{z_a+z_b,\mathcal{F}}(t)} \dot{T}_\pi^{MWM^{\mathcal{F}}_{z_a+z_b}}(t) = S\frac{z_b}{z_a+z_b} > 1. \tag{24}$$

Arguing as for $z_a = 0$ with (24) instead of (14), the theorem follows for $z_a > 0$.

## 4 Networks of switches that deploy different scheduling policies

In the previous section, we introduced scheduling policies for networks that provide stability for a network of switches where all switches implement the same scheduling policy. Here we prove that a network of switches in which each switch deploys any of those policies also achieves 100 % throughput.
**Theorem 2** *A network of IQ/CIOQ switches with configuration overhead where each switch deploys any $MWM_c^F$ policy, $\mathcal{F} \in \mathcal{G}$, in which the weight is defined as in (8), and which deploys a speedup $S \geq \left\lceil \frac{z_a+z_b}{z_b} \right\rceil$ achieves 100% throughput.*
*Proof:* We divide the switches in the network into $M$ groups $G_i$, $i \in \{1,..,M\}$ where $G_i$ contains the switches that deploy the switching policy $MWM_c^{F_i}$. Accordingly, we can divide the departure vector $D(t)$ and the arrival rate vector $W$ in M subvectors, i.e., we write $D(t) = (D_1(t),..,D_M(t))$ and $W = (W_1,..,W_M)$.

In order to prove rate stability, it is obviously sufficient to show that $\forall i \in \{1, .., M\}$, $\lim_{t\to\infty} \frac{D_i(t)}{t} = W_i$, w.p.1. This relation can be proved by applying the proof of theorem 1 to each group of switches $G_i$ separately.

## 5 Conclusions

This paper investigates scheduling policies for networks of switches with a configuration overhead. It proposes a class of switching policies that are based on maximum weight matchings. It is shown that network of switches that deploy either one or any mixture of those policies with a speedup of $S \geq \lceil \frac{z_a+z_b}{z_b} \rceil$ achieve 100% throughput.

## References

1. Ajmone, M.,Giaccone, P., Leonardi, E., Mellia, M., Neri, F., *Local scheduling policies in networks of packet switches with input queues,* Proc. of Infocom 2003, San Francisco, April 2003.
2. Ajmone, M.,Leonardi, E., Mellia, M., Neri, F., *On the throughput achievable by isolated and interconnected input-queued switches under multicalss traffic,* Proc. of Infocom 2002, New York City, June 2002.
3. Andrews, M., Zhang, L., *Achieving stability in networks of input queued,* Proc. of Infocom 2001, Anchorage, Alaska, April 2001.
4. Dai, J.G., Prabhakar, B., *The throughput of data switches with and without speedup,* Proc. of IEEE Infocom 2000, Tel Aviv.
5. Dai, J.G., *Stability of fluid and stochastic processing networks,* Miscellanea publication n.9, Centre for Mathematical Physics and Stochastic, Denmark (http://www.maphysto.dk), Jan. 9.
6. Lin, L.Y., *Micromachined free-space matrix switches with submillisecond switching time for large-scale optical crossconnect,* OFC Tech. Digest, 1998.
7. Keslassy, I., McKeown, N., *Achieving 100% throughput in an input queued switch,* IEEE Transactions on Communications, vol. 47, no. 8, Aug. 1999, 1260 - 1272.
8. Keslassy, Chuang, S.T., Yu, K., Miller, d., Horowitz, M., Solgaard, M., McKeown, N., *Scaling Internet Routers Using Optics,* Proc. of ACM SIGCOMM, Karlsruhe, Germany, Aug. 2003.
9. Li, X., Hamdi, M., *$\lambda$-adjust algorithm for optical switches with reconfiguration delay,* Proc. of ICC'03, Anchorage, Alaska, May 2003. New York City, June 2002.
10. Li, X., Hamdi, M., *Design and analysis of scheduling algorithms for switches with reconfiguration overhead,* Proc. of High Performance Switching and Routing (HPSR'03), Torino, Italy. June 2003.
11. Li, X., Hamdi, M., *Analysis of reduced rate scheduling for switches with reconfiguration overhead,* Proc. of Global Communications Conference (Globecom'03), San Francisco, Dec. 2003.
12. McKeown, N., *Optics inside Routers,* Proc. of ECOC 2003, Rimini, Italy, September 2003.
13. A. Neukermans and R. Ramaswami, *MEMS Technology for Optical Networking Applications*, IEEE Communications Magazine January 2001, p.62 - 69.
14. Shah, D, Gangali, Y., Keshavarzian, A., *Input-queued switches: Cell switching versus packet switching*, Proc. of IEEE Infocom 2003, San Francisco. April 2003.
15. Towles, B., Dally, W.J., *Guaranteed scheduling for switches with configuration overhead,* Proc. of Infocom 2002, New York City, June 2002.