Paper

SMM Clos-Network Switches under SD Algorithm

Janusz Kleban[1] and Jarosław Warczyński[2]

[1] *Faculty of Electronics and Telecommunications, Poznan University of Technology, Poznań, Poland*
[2] *Faculty of Electrical Engineering, Poznan University of Technology, Poznań, Poland*

https://doi.org/10.26636/jtit.2018.123017

Abstract—This paper is devoted to evaluating the performance of Space-Memory-Memory (SMM) Clos-network switches under a packet dispatching scheme employing static connection patterns, referred to as Static Dispatching (SD). The control algorithm with static connection patterns can be easily implemented in the SMM fabric due to bufferless switches in the first stage. Stability is one of the very important performance factors of packet switching nodes. In general, a switch is stable for a particular arrival process if the expected length of the packet queues does not increase without limitation. To prove the stability of the SMM Clos-network switches considered under the SD packet dispatching scheme the discrete Markov chain model of the switch is used and Foster's criteria to extend Lyapunov's second (direct) method of stability investigation of discrete time stochastic systems are used. The results of simulation experiments, in terms of average cell delay and packet queue lengths, are shown as well.

Keywords— *Clos-network switch, packet dispatching algorithms, packet switching network, stability of switching network*

1. Introduction

Connecting paths between input and output ports in switches/routers are provided by switching fabrics, which are the main part of every packet switching node. The switching fabrics replace buses which are too slow, mainly in medium-sized and high-end routers and switches. They can establish connections between input ports and requested output ports, while simultaneously transmitting packets. Single-stage switching fabrics known as crossbar switches are used mainly in medium-sized routers/switches [1].

Basically, an $N \times N$ crossbar switch consists of a square array of N^2 individually operated crosspoints (N represents the number of inputs and outputs). Each crosspoint has two possible states: cross (default) and bar, and corresponds to the input-output pair. A connection between input port i and output port j is established by setting the (i, j)-th crosspoint to the bar state, while letting other crosspoints along the connecting paths remain in the cross state. The crossbar switch can transfer up to N cells from different input ports to different output destinations within the same time slot. The control algorithm for the crossbar fabric is very simple, as the bar state of the crosspoint can be triggered individually by each incoming packet when its destination matches the output address. Crossbar fabrics are complex in terms of the number of crosspoints, which grows as N^2. The arbitration process that has to choose packets to be sent from inputs to outputs within each time slot can also become the system's bottleneck, as the switch size increases.

In high-end routers, multi-stage or even multi-stage and multi-plane switching fabrics are used. These types of switching fabrics are currently used by network equipment vendors in core routers, e.g. Cisco's CRS series, Juniper's T series, and Brocade's BigIron RX Series. For example, in Cisco's new router called Carrier Routing System-X (CRS-X), a multi-stage and multi-plane switching fabric is used. This family of routers focuses on the extreme scale. One standard deployment of a 7-ft rack chassis of CRS-X routers can deliver up to 12.8 terabits per second. The system can be clustered together in a massive configuration of up to 72 chassis, which would deliver up to 922 Tb/s of throughput [2].

Clos-network switches are a very attractive solution for core routers because of their modular and scalable architecture. The Clos-network fabric is composed of crossbar switches arranged in stages [3]. According to the required combinatorial properties, it is possible to build [4]:

- strict-sense nonblocking (SSNB),
- wide-sense nonblocking (WSNB),
- rearrangeable (RRNB),
- repackable (RPNB) non-blocking networks.

In SSNB [3] networks, no call is blocked at any time. WSNB [5], [6] networks are able to connect any idle input and any idle output, but a special path-searching algorithm must be used. RRNB [5] networks can also establish the required connections between any idle input-output pair, but a rearrangement of some existing connections to other connecting paths may be needed to change the network state in order to unblock a blocked call. RPNB [7], [8] networks employ rearrangements after call termination to prevent the switching fabric from entering blocking states. The presented classes of switching networks were proposed in the past, when circuit-switching telephone exchanges supported voice traffic.

Currently, telecommunication networks focusing on packet services and high-speed switching fabrics adopt the use of fixed-length packets called cells. All incoming variable-length packets (e.g. IP packets) are segmented at ingress line cards into fixed-size cells. Next, they are transmitted within time slots through the switching fabric, and reassembled into packets at egress line cards, before they depart [1]. In high-speed routers it is not necessary to use SSNB switching fabrics, because a new set of connecting paths may be set up for each time slot. RRNB fabrics are sufficient to satisfy all requirements related to one-to-one connections between all sources and destinations. They can establish connecting paths for all possible permutations of input-output pairs. These connections can be established on a call-by-call basis or simultaneously for a given set of inputs and outputs. The former technique employs rearrangements of the existing paths when a new call cannot be set up. In the latter method, parallel processing of all required input-output connections is carried out and, next, the connecting paths are set up simultaneously in the switching fabric.

The main difference between circuit switching and packet switching fabrics is that in circuit switching systems, when the output port is busy, the call is lost. In packet switching fabrics, when outputs are busy, cells are buffered and wait in queues. This means that queues of cells destined for, let's say, a very popular output port, may grow, because many cells should be sent to the same output port, but only one cell can be sent out within one time slot.

While a cell is being routed in a packet switching fabric, it can face a contention problem resulting from the fact that two or more cells compete for a single resource. Algorithms that can solve contentions, are usually called packet dispatching schemes. Cells that have lost contention must be either discarded or buffered. Buffers are also used to alleviate the complexity of packet dispatching algorithms and to absorb possible contentions. According to buffer allocation schemes, Clos-network packet switches are classified as: Space-Space-Space (SSS or S^3), Memory-Memory-Memory (MMM), Memory-Space-Memory (MSM), and Space-Memory-Memory (SMM) switches. MSM Clos-network switch seems to be the best architecture investigated. The basic packet dispatching algorithms for this kind of switching fabrics were proposed in [9], [10]. A modified MSM Clos-network switch was proposed and investigated in [11].

In this paper, we analyze the SMM Clos-network switch, where bufferless modules are used in the first stage and buffered crossbars in the second and third stages. Due to bufferless modules in the first stage, a very simple control algorithm may be implemented to distribute cells to the central modules, e.g. static dispatching (SD). The SMM architecture was proposed in [12], where an analytical analysis for admissible traffic was performed. In [13], different kinds of backpressure schemes between central modules and input modules are evaluated, in terms of maximum buffer usage in central modules. The packet dispatching scheme proposed in [14] uses static dispatching patterns and internal backpressure signals. It is dedicated for SMM Clos-network switches, where the second and third stages are made of crosspoint queued (CQ) switches [15]. In [16], a fault-tolerant desynchronized static round-robin (FT-DSRR) cell dispatching algorithm was proposed. The FT-DSRR algorithm is an adaptation of the DSRR algorithm to SMM Clos-network switches, where serious crosspoint faults induced by harsh space radiation environment may take place. It may be used to control onboard switches.

This paper deals with an SMM Clos-network switch, where output queued switches are used in the second and third stages. The main contribution of this paper is the proof of stability of the SMM Clos-network switch using the Discrete Time Markov Chain (DTMC) model and an analytical approach based on Foster's stochastic criteria, analogous to the direct method of Lyapunov which was aimed at inferring about the stability of deterministic dynamic systems. The theoretical results were verified by simulation investigations of RRNB and SSNB architectures of a network under uniform and non-uniform traffic distribution patterns.

The remainder of this paper is organized as follows. Section 2 introduces some background knowledge concerning the SMM Clos-network switch and the SD algorithm. In Section 3 input traffic analysis is performed. Using a stochastic Lyapunov-like analytical method, we prove that the investigated switching fabric is stable under the SD packet dispatching scheme in Section 4. Section 5 presents simulation results obtained for the SD scheme. We conclude this paper in Section 6.

2. SMM Clos-Switching Fabric and SD Scheme

The three-stage Clos switching fabric architecture is denoted by $C(m,n,r)$, where the parameters m, n, and r entirely determine the structure of the network. There are r input modules (IM) of capacity $n \times m$ in the first stage, m central modules (CM) of capacity $r \times r$, and r output modules (OM) of capacity $m \times n$ in the third stage. The capacity of this switching system is $N \times N$, where $N = nr$. The three-stage Clos-network switch is strictly non-blocking if $m \geq 2n-1$ and rearrangeable non-blocking if $m \geq n$.

In the basic SMM Clos-network switch (shown in Fig. 1), the first stage consists of r bufferless IMs with n input ports (IPs) each. The second stage consists of m CMs, and each of them has r FIFO buffers (COQs), one per output. A maximum of r cells from r IMs may arrive at one COQ buffer, so it must work r times faster than the line rate. The third stage consists of r OMs, where each output port $OP(j,h)$ has FIFO output buffer (OQ). A maximum of m cells from m CMs may arrive at one OQ, so to store all cells during one time slot it must work m times faster than the line rate. The interstage links between IMs and CMs are denoted by $L_I(i,k)$, where i represents the number of IMs, and k – the number of CMs, whereas $L_C(k,j)$ denotes interstage links between $CM(k)$, and $OM(j)$. Instead of using shared-memory CM and OM modules, it is possible to employ CQ switches, where speed-up is not necessary [15].

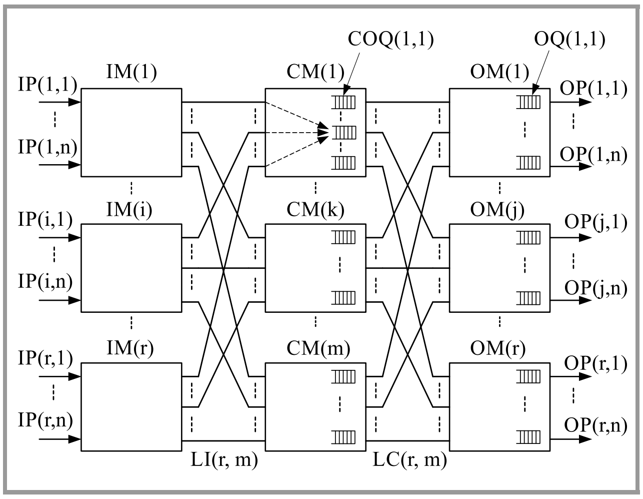

Fig. 1. An SMM Clos-network switch.

The SD scheme investigated in this paper seems to be the simplest packet dispatching algorithm that can be implemented in the SMM Clos-network switch. It is an adaptation of the Static Round-Robin Dispatching (SRRD) to the SMM Clos-network switch, and is less demanding in terms of hardware, in comparison with other proposed schemes (e.g. [14]). The SD scheme does not need any special arbitration, e.g. the handshaking processes, to distribute cells to the CMs. The key idea of the scheme is based around static connection patterns which are used in each IM. The consecutive static connection patterns used in IMs are shown in Fig. 2.

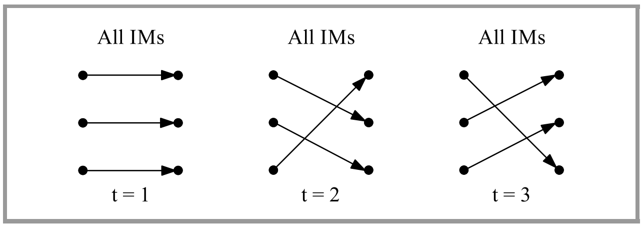

Fig. 2. A sequence in which the static connection patterns should be changed in each IM of capacity 3 × 3.

The connection patterns are the same in all IMs and are shifted to the next one in consecutive time slots. Cells arriving at each input are at once distributed to the CMs, and are stored in COQs related to the destined OMs. In the first time slot, cells from $IP(x,1)$ are sent to $CM(1)$, from $IP(x,2)$ to $CM(2)$, from $IP(x,3)$ to $CM(3)$; in the second time slot, cells from $IP(x,1)$ are sent to $CM(2)$, from $IP(x,2)$ to $CM(3)$, from $IP(x,3)$ to $CM(1)$, and so on. Arriving cells are evenly distributed to CMs, to decrease cell delay within the SMM Clos-network switch. The SD scheme may be also adopted in the MSM Clos-network switch [7].

3. Input Traffic Analysis

We assume that the traffic directed to each input port IP(i, h) can be modeled by an i.i.d. Bernoulli process, where the number of successes – which means the number of cells arriving in t time slots (in t trails) is tp_B with p_B denoting the probability of success in one trial. In such a case, the ports' arrival rate is expressed by the expected value:

$$\lambda_{IP} = \lim_{t\to\infty} \frac{t\,p_B}{t} = p_B\,. \tag{1}$$

Therefore, the input traffic arriving at one input module is equal to λ_{IM}=np_B, and at the whole switching fabric, all input modules – rnp_B. The SD algorithm balances this input load on CMs and after m time slots the central modules' arrival rate can be expressed in the following way:

$$\lambda_{CM} = \frac{n\,p_B\,r}{m}\,. \tag{2}$$

There are output queues (COQs) in each central module which stores cells destined for the predetermined OMs. When analyzing the input rate of these queues, it is easy to see that this rate can be assessed as:

$$\lambda_{COQ(i,j)} = \frac{np_B r}{m} p_{ij}\,, \tag{3}$$

where p_{ij} represents the probability of a cell arriving from the i-th input module being destined for the j-th output module. For example, with traffic uniformly distributed to the output ports and, in consequence, to the output modules OMs, $p_{ij} = 1/r$. This means that even for the maximum input port load, i.e. for $p_B = 1$, the rate $\lambda_{COQ(i,j)}$ is less than or equal to 1, if the number of OMs is $m \geq n$.

In the investigated SMM Clos-network architecture, each central module CM has one link to each OM. This ensures that in each time slot from any non-empty COQ(i, j), one cell will be sent to the appropriate OM(j), which can be described by the COQ(i, j) queue's service rate $\mu = 1$.

4. Stability Proof

The theory of stability for deterministic dynamic systems was founded by Lyapunov [18] (see also [19] for survey of stability ideas) who invented two methods of stability investigation. His second method, known as *Lyapunov's second method* or *indirect method*, turned out to be very effective in proving the stability of a very wide spectrum of deterministic systems – linear, non-linear, continuous and discrete. Later, Lyapunov's ideas were extended to stochastic systems, mainly by Foster [20]. The application of this theory to Markov chains was due to Meyn and Tweedie [21].

The term *stability*, in the context of dynamic systems described by ordinary differential equations, is commonly used to mean asymptotic stability, i.e. convergence of a system's state paths to a fixed, stable, point.

With Markovian systems, convergence must be understood in a distributional sense and, therefore, is called stochastic stability. It considers stochastic convergence in the time of a Markov chain $X = (X_n;\ n = 0, 1, \ldots)$, as $n \to \infty$.

In general terms, the Markov chain is topologically stable if there is a positive probability that it does not leave the compact center of the state space (which is called *non-evanescence* [21]), or, using a stronger condition, if the distributions of the chain as time evolves are tight (bounded in probability [21]). Meyn and Tweedie say the chain is probabilistically stable if it returns to sets of positive measure (Harris recurrence), or if there is a unique invariant probability measure for it (positive Harris recurrence) [21]. According to [20] and [21], the stability proof of stochastic systems modeled by Markov chains must show:

1. the irreducibility of the chain, which means that starting from any initial state, it is possible to arrive in subsequent transitions on any other state of the chain. Figs. 3 and 4 show examples of irreducible and non-irreducible Markov chains;
2. the positive recurrence of the chain, which can be done by demonstrating the negative drift of the Lyapunov function.

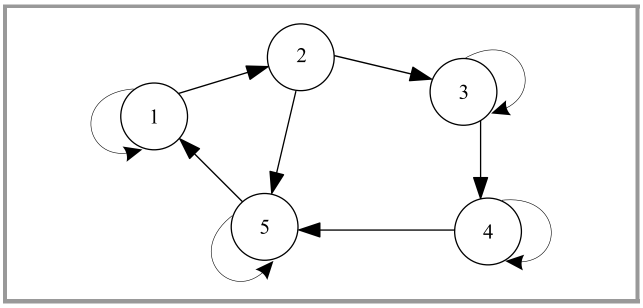

Fig. 3. Example of an irreducible Markov chain.

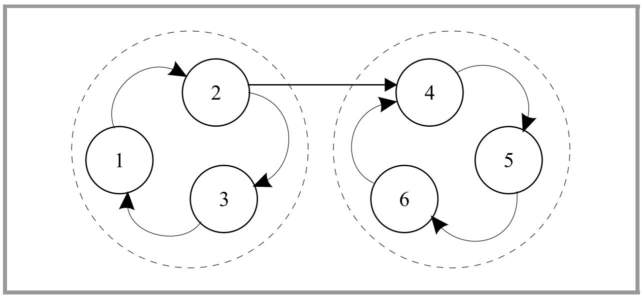

Fig. 4. Example of a non-irreducible Markov chain.

The positive recurrence of the irreducible chain's state means (see for example [21]) that

$$E(\tau_i | X_0 = i) < \infty . \tag{4}$$

That is, if state i is positive recurrent, then the chain comes back infinitely often to state i and the time τ_i between two consecutive visits is finite.

Denoting by P_i the conditional probability of the process started at state i, we say, by definition (see for example [21]), that a state i is:

1. transient if $P_i(\tau_i = \infty) > 0$,
2. null recurrent if $P_i(\tau_i < \infty) = 1$,
3. positive recurrent if it is recurrent and $E_i(\tau_i) < \infty$.

For irreducible Markov chains, condition (4) implies a positive recurrence of state i and, hence, a positive recurrence of the whole chain, i.e. in a given class, all states are either positive recurrent, null recurrent or transient.

Lemma 1: If X is irreducible, then all states are of the same type.

Proof: The proof can be based on the following fact: If X is irreducible and $j \neq k$ are any two states, then $P_j(\tau_k < \tau_j) > 0$. Now, let us assume the opposite – that this probability equals 0. Then, by the strong Markov property, the process starting from j would never visit state k. This is, however, in contradiction with the irreducibility of X.

Let S be the state space of a given DTMC and let $P \subset S$ be a finite subset of S. Denoting by τ_P the time of the first visit to set P, one can state the following generalization of condition (4) (according to the guidelines in [21]):

Lemma 2: Let $X = (X_n;\ n = 0, 1, \ldots)$ denote an irreducible DTMC with state space S and let $P \subset S$ be a finite subset of S. Chain X is positive recurrent if and only if:

$$E(\tau_P | X_0 = i) < \infty \text{ for all } i \in P . \tag{5}$$

However, it is rather difficult to determine with Eq. (5) whether a given Markov chain is positive recurrent or not. Here, the Lyapunov-Foster criteria can be used [20]:

Let $X = (X_n;\ n = 0, 1, \ldots)$ be an irreducible Markov chain defined on some countable space S with transition probabilities p_{ij}, $i, j \in S$. On the basis of [20], we can state:

Theorem 1: The Markov chain X is positive recurrent if and only if there exists a finite set $S_0 \in S$ and a function $V\!: S \to R^+$ with $\inf\{f(i) : i \in S\} > -\infty$ and a constant $\varepsilon > 0$ such that:

$$\sum_{j \in S} p_{ij} V(j) < \infty \qquad \text{for all } i \in S_0 , \tag{6}$$

and

$$\sum_{j \in S} p_{ij} V(j) \leq V(i) - \varepsilon \qquad \text{for all } i \notin S_0 . \tag{7}$$

The function $V\!: S \to R^+$ is commonly referred as the Lyapunov-Foster function.

Equations (6)–(7) can be rewritten in the equivalent form:

$$E[V(X_{n+1}) | X_n = i] < \infty \text{ for } i \in S_0 , \tag{8}$$

and

$$E[V(X_{n+1}) - V(X_n) | X_n = i] \leq -\varepsilon \text{ for } i \notin S_0 . \tag{9}$$

Looking at Eq. (8)–(9), it is easy to notice that Foster's criteria can be interpreted as conditions for the Lyapunov's function drift, which is in analogy with Lyapunov's stability direct method for dynamical systems described by ordinary differential equations.

The function fulfilling Lyapunov's conditions can be regarded as a Lyapunov candidate function (only a candidate function which allows stability proving is called a Lyapunov function). There are requirements imposed on Lyapunov candidate function $V(x)$ [18]:

1. $V(x)$ is scalar on the investigated system's state vector **x**; switching networks' states are determined by queue lengths,
2. is positive definite, i.e.: $\underset{x \neq 0}{\forall}\, V(x) > 0;\ V(0) = 0$,
3. $V(x)$ grows with the state growth of the investigated system which, in our case, means that it grows with the length of switching network queues,
4. for continuous systems: $V(x) \in C_1$.

Speaking generally, there are two levels of stability [18]–[21] – the so-called *weak stability* and the *asymptotic stability*. A proof of weak stability for a given switch network guarantees its full, 100% throughput, but does not predetermine the maximum delay of cells, which in general may be unlimited. The asymptotic stability is a more demanding level of stability, which guarantees not only full throughput of the network, but also a finite value of the maximum cell delay.

Formally, the switching system in which the packet (cell) arrival is an independent random process is characterized by the weak (in Lyapunov sense) stochastic stability if for every $\varepsilon > 0$ there exists $\delta > 0$ that:

$$\underset{\varepsilon>0}{\forall}\ \exists \delta > 0\ \lim_{t\to\infty} P\{\|q_t\| > \delta\} < \varepsilon\,, \tag{10}$$

or

$$\underset{\varepsilon>0}{\forall}\ \exists \delta > 0\ \lim_{t\to\infty} P\{\|q_t\| < \delta\} < 1 - \varepsilon\,, \tag{11}$$

where $P\{Z\}$ denotes the probability of event Z, and $||q_t||$ is any norm of q_t – the measure of queues in the system.

The asymptotic stochastic stability is defined as follows: a switching fabric in which the packet (cell) arrival is an independent random stationary process characterized by asymptotic stochastic stability if:

$$\lim_{t\to\infty} \sup E\{\|q_t\|\} < \infty\,. \tag{12}$$

Inequality (12) means that the maximum expected value of $||q_t||$ is finite. Asymptotic stochastic stability guarantees limited average queue lengths and limited cell delay times.

As shown above, the dynamics of the SMM switching fabric is determined by COQ queues (due to static connections of the central stage with the first and third stages, contentions are possible only in the COQ queues).

Let us note that the dynamics of the COQ(*i, j*) queue can be represented by the Markov chain's state diagram depicted in Fig. 5, where λ represents the queue arrival rate – $\lambda_{COQ(i,j)}$, and μ – is the queue service rate.

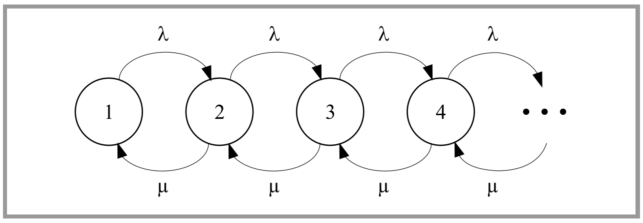

Fig. 5. State graph of a COQ(i, j) queue.

The proof of stability of this queue can be based on the Foster-Lyapunov criterion [19]–[21]. It requires that the Lyapunov candidate function $V(q_t)$, defined on the queue length, has a negative drift, strictly that:

$$\underset{\|q_t\|>\varepsilon}{\forall}\quad E\left[V\ (q_{t+1}) - V(q_t)|\, q_t\right] < -\delta\,. \tag{13}$$

In the following proof of stability, Lyapunov candidate function is chosen as the simplest possible one:

$$V(q_t) = q_t\,. \tag{14}$$

The selected function $V(q_t)$ satisfies the Lyapunov candidate function requirements specified above. After substituting the selected function $V(q_t)$ into the left-hand side of inequality (13) and taking into account the graph in Fig. 5:

$$\begin{aligned} E\left[V\ (q_{t+1}) - V(q_t)|\, q_t\right] &= E[q_{t+1}|\, q_t] - E[q_t|\, q_t] = \\ = E[q_{t+1}|\, q_t] - q_t &= \Big[\tfrac{\lambda}{\lambda+\mu}(q_t+1) + \\ + \tfrac{\mu}{\lambda+\mu}(q_t-1)\Big] - q_t &= \tfrac{\lambda-\mu}{\lambda+\mu}\,, \end{aligned} \tag{15}$$

eventually, the stability condition is:

$$\frac{\lambda-\mu}{\lambda+\mu} < 0\,. \tag{16}$$

The drift is negative when $\lambda < \mu$. For μ = 1, the system will be weakly stable (stable in Lyapunov sense) for $\lambda < 1$. It is worth noting that it does not follow that for λ = 1 the system will not be stable. The Lyapunov method proves only the stability, and if that fails, the instability of the studied system does not follow from it.

In order to prove the asymptotic stochastic stability, it should be shown that:

$$\underset{\|q_t\|>\varepsilon}{\forall}\ E\left[V\ (q_{t+1}) - V(q_t)|\, q_t\right] < -\delta\, \|q_t\|\,. \tag{17}$$

For this purpose, we need another Lyapunov candidate function $V(q_t)$ – we choose it as:

$$V(q_t) = q_t^2\,. \tag{18}$$

The drift of this function is:

$$\begin{aligned}&E\left[V\left(q_{t+1}\right)-V\left(q_t\right)\middle|q_t\right]=E\left[q_{t+1}^2\middle|q_t\right]-E\left[q_t^2\middle|q_t\right]=\\&=E\left[q_{t+1}^2\middle|q_t\right]-q_t^2=\left[\tfrac{\lambda}{\lambda+\mu}(q_t+1)^2+\right.\\&\left.+\tfrac{\mu}{\lambda+\mu}\left(q_t-1\right)^2\right]-q_t^2=\tfrac{\lambda}{\lambda+\mu}\left(q_t^2+2q_t+1\right)+\\&+\tfrac{\mu}{\lambda+\mu}(q_t^2-2q_t+1)-q_t^2=q_t^2\left(\tfrac{\lambda}{\lambda+\mu}+\tfrac{\mu}{\lambda+\mu}-1\right)+\\&+2q_t\left(\tfrac{\lambda}{\lambda+\mu}-\tfrac{\mu}{\lambda+\mu}\right)+\tfrac{\lambda+\mu}{\lambda+\mu}=\\&=2q_t\tfrac{\lambda-\mu}{\lambda+\mu}+1=2q_t\tfrac{-(\mu-\lambda)}{\lambda+\mu}+1\,.\end{aligned} \tag{19}$$

Solving the inequality:

$$2q_t\frac{-(\mu-\lambda)}{\lambda+\mu}+1<0\,, \tag{20}$$

the conditions for asymptotic stability can be determined. For $\mu = 1$ we obtain:

$$q_t > \frac{\lambda+1}{2(1-\lambda)} \text{ and } \lambda < 1\,. \tag{21}$$

This means that the asymptotic stability will only occur for a sufficiently large q_t, for example, assuming $\lambda = 0.9$, this will be an average of 10 cells, that is, when the value is reached, the cell delay will be limited and stabilized.

5. Simulation Experiments

The experiments have been carried out mainly for the RRNB Clos-network switch $C(8,8,8)$ of size 64 × 64 (8 switches in each stage) under the SD algorithm. The SSNB $C(8,16,8)$ architecture, with 8 switches in the first and last stages, and 15 switches in the second stage was also investigated. A wide range of traffic loads per input port, from $p_B = 0.05$ to $p_B = 1$, with the step of 0.05, was considered in each simulation experiment. 95% confidence intervals that have been calculated after t-student distribution for ten series with 250,000 time slots (after the starting phase comprising 50,000 time slots, which enables reaching the stable state of the SMM Clos-network switch) are at least one order lower than the mean value of the simulation results, therefore they are not shown in the figures. It is assumed that in the second and third stages the switches with output buffers are used, and the size of buffers is not limited. Three main performance measures have been evaluated: average cell delay in time slots, maximum size of OQs, and throughput. A switch can achieve 100% throughput under uniform or non-uniform traffic, if the switch is stable, as it was defined in [22]. It means that the cell queues do not grow without limitation.

Two packet arrival models are considered in simulation experiments: the Bernoulli arrival model, and the bursty traffic model, where the average burst length is set to 16 cells. Several traffic distribution models (the most popular one in this area of research) have been considered, which determine probability p_{ij} that a cell, which arrives at input i, will be directed to output j. The considered cell distribution models are: uniform – $p_{ij}= p_B/N$, diagonal – $p_{ij}= 2p_B/3$ for $i = j$ and $p_{ij}= p_B/3$ for $j = (i+1) \bmod N$, and 0 otherwise, and Hot-spot: $p_{ij}= p_B/2$ for $i = j$, and $p_B/2(N\text{-}1)$ for $i \neq j$. Selected simulation results are shown in Figs. 6, 7, and 8. Figure 6 shows the average cell delay, in time slots, obtained for Bernoulli and bursty arrival models, and different kinds of cell distribution models. The SD algorithm provides 100% throughput for the investigated switching fabric only for uniform traffic and the Bernoulli arrival model. Under Bernoulli arrivals, the throughput is limited to 90% for non-uniform traffic, such as diagonal and Hot-spot. It is possible to say that the SD scheme, for uniform and non-uniform traffic distribution patterns under Bernoulli arrivals, performs quite well when the input load is lower than 0.85. In this case, the average cell delay is not greater than 10 time slots. For the bursty arrival model, the SMM Clos-network switch controlled by the SD algorithm is not

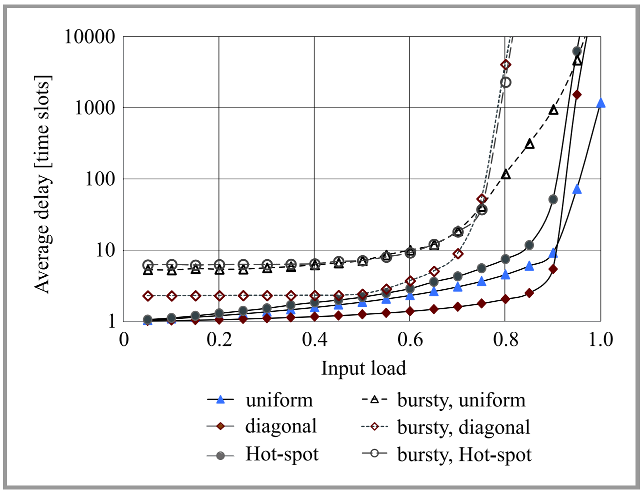

Fig. 6. Average cell delay at egress side of the SMM Clos-network switch under the SD scheme.

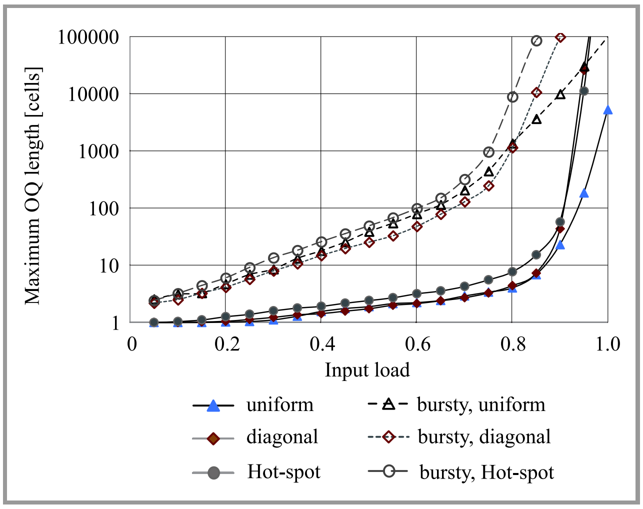

Fig. 7. Maximum OQ length in OMs under the SD scheme.

able to achieve 100% throughput for both the uniform and non-uniform traffic distribution patterns. For the uniform traffic, the throughput is close to 98%, but for the non-uniform traffic, the throughput is limited to 80%.

Figure 7 shows the maximum OQ length obtained during simulation experiments. These results are consistent with the charts presented in Fig. 6. It can be seen that for Bernoulli arrivals the OQ length rapidly grows for a heavy input load and non-uniform traffic (p_B>0.9). In bursty traffic, the OQ length increases very fast for p_B>0.75, especially for non-uniform cell distribution patterns.

Speaking generally, the SD algorithm is very simple to implement within the SMM Clos-network switch and can produce good results for the input load of p_B<0.7, both for uniform and non-uniform traffic distribution patterns. The results related to throughput are not impressive, but the complexity of this algorithm is very low.

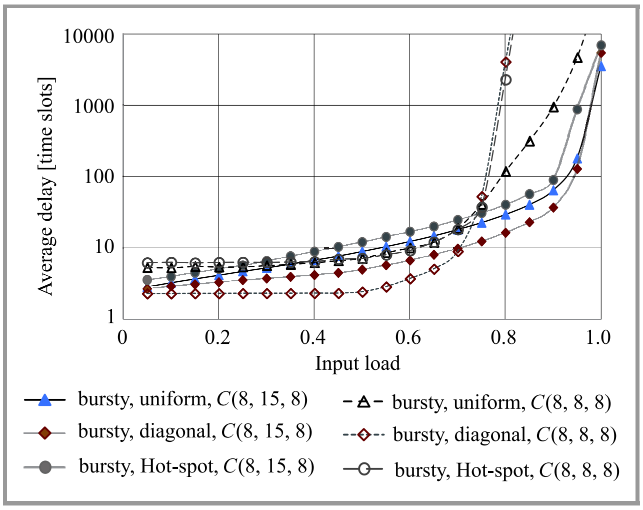

Fig. 8. Average cell delay at egress side of the SMM Clos-network switch for $C(8,8,8)$ and $C(8,15,8)$ architectures under bursty traffic.

Figure 8 shows a comparison of the average cell delay under bursty traffic for RRNB $C(8,8,8)$ and SSNB $C(8,15,8)$ architectures. As it was stated above, in packet switching nodes, connection patterns may be changed in every time slot, so the RRNB switching fabric can set up all connections possible between inputs and outputs. The SSNB architecture contains more switches in the second stage than the RRNB architecture. In this case, cells are distributed more evenly to a higher number of buffers located in the second stage, and can thus reach the destined output with a lower delay. As it is shown in Fig. 8, the SSNB fabric can offer 100% throughput and produces better results than the RRNB fabric under the SD scheme and bursty traffic. For input loads lower than 0.9, the average delay is lower than 100 time slots for all traffic distribution patterns investigated. The delay grows very fast for input load greater than 0.9, but the average delay is very high only for input loads equal to 1, and equals about 5000 time slots.

6. Conclusions

This paper aims to evaluate performance of the SMM Clos-network under the packet dispatching scheme employing static connection patterns, called SD. The system was evaluated in terms of stability and basic performance measures, such as average cell delay and packet queue lengths. In Section 4 we showed how to use the DTMC model and an analytical approach based on Foster's stochastic criteria, analogous to the direct Lyapunov's method, to prove the stability of the SMM Clos-network switch under the SD algorithm. Taking into account that the stability is proven for ideal, theoretical traffic, in Section 5 we showed simulation results obtained for uniform and non-uniform traffic distribution patterns, and for Bernoulli and bursty arrival models. Two architectures of the SMM Clos-network switch were taken into account: RRNB $C(8,8,8)$ and SSNB $C(8,15,8)$. The investigated cell dispatching scheme is very simple, but it is not able to provide satisfactory performance of the RRNB SMM Clos-network switch for very high input load, greater than 0.7, especially for bursty traffic The results are better for the SSNB architecture, but in this case more switches in the second stage must be used, and the cost of such a network will be higher. It is also impossible to provide in-sequence service under this algorithm, which results in special resequencing buffers at outputs. Furthermore, this re-sequence function makes a switch more difficult to implement, especially as the port speed and switch size increase.

Acknowledgements

The work described in this paper was financed with the use of the funds of the Ministry of Science and Higher Education for the year 2017 under Grants 08/82/DSPB/8221 and 04/45/DSPB/0162.

References

[1] H. J. Chao and B. Liu, *High Performance Switches and Routers. Wiley-Interscience*, New Jersey, USA: Wiley, 2007.

[2] Router-Switch.com, Cisco CRS-X Core Router to Offer 10 Times Capacity of Original [Online]. Available: http://blog.router-switch.com/2013/06/cisco-crs-x-core-router-to-offer-10-times-capacity-of-original/ (accessed on February 14 , 2018).

[3] C. Clos, "A Study of Non-Blocking Switching Networks", *Bell Sys. Tech. Jour.*, vol. 32, no. 2 pp. 406–424, 1953.

[4] W. Kabaciński, *Nonblocking Electronic and Photonic Switching Fabrics,* Berlin: Springer, 2005.

[5] V. E. Beneš, *Mathematical Theory of Connecting Networks and Telephone Traffic,* New York: Academic Press, 1965.

[6] V. E. Beneš, "Semilattice characterization of nonblocking networks". *The Bell System Techn. J.*, vol. 52, no. 5, pp. 697–706, 1973.

[7] H. M. Ackroyd, "Call repacking in connecting networks", *IEEE Transact. on Commun.*, vol. 27, no. 3, pp. 589–591, 1979.

[8] A. Jajszczyk and G. Jekel, "A New Concept – Repackable Networks", *IEEE Transact. on Commun.*, vol. 41, no. 8, pp. 1232–1237, 1993.

[9] E. Oki, Z. Jing, R. Rojas-Cessa, and H. J. Chao, "Concurrent round-robin-based dispatching schemes for Clos-network switches", *IEEE/ACM Transact. on Network.*, vol. 10, no. 6, pp. 830–844, 2002.

[10] J. Kleban and A. Wieczorek, "CRRD-OG: A Packet Dispatching Algorithm with Open Grants for Three-Stage Buffered Clos-Network Switches", in *Proc. 2006 Workshop on High Performance Switching and Routing HPSR2006*, Poznań, Poland, 2006, pp. 315–320.

[11] J. Kleban, "Packet dispatching using module matching in the modified MSM Clos-network switch", *Telecommun. Sys,*, vol. 66, no. 3, pp 505–513, 2017.

[12] X. Li., Z. Zhou, and M. Hamdi, "Space-Memory-Memory architecture for Clos-network packet switches". in *Proc. IEEE Int. Conf. on Commun. – ICC 2005*, Seoul, Korea (South), 2005, vol. 2, pp. 1031–1035.

[13] A. V. Manolova, S. Ruepp, A. Rytlig, M. Berger, H. Wessing, and L. Dittmann, "Internal backpressure for terabit switch fabrics", *IEEE Commun. Let.*, vol. 16, no. 2, pp. 265–267, 2012.

[14] J. Kleban and U. Suszyńska, "Static Dispatching with Internal Backpressure Scheme for SMM Clos-Network Switches", in *Proc. The Eighteenth IEEE Symp. on Computers and Commun., ISCC'13*, Split, Croatia, 2013, pp. 654–658 (doi:10.1109/iscc.2013.6755022).

[15] K. Yoshigoe, "The Crosspoint-Queued Switches with Virtual Crosspoint Queueing". in *Proc. 5th Int. Conf. on Signal Proces. and Commun. Sys. ICSPCS 2011)*, Honolulu, HI, USA, 2012, pp. 277–281.

[16] K. Liu, J. Yan, and J. Lu, "Fault-tolerant Cell Dispatching for Onboard Space-Memory-Memory Clos-Network Packet Switches", in *Proc. 16th Int. Conf. on High Performance Switching and Routing HPSR*, Budapest, Hungary, 2015 (doi:10.1109/HPSR.2015.7483090).

[17] J. Kleban and H. Santos, "Packet Dispatching Algorithms with the Static Connection Patterns Scheme for Three-Stage Buffered Clos-Network Switches", in *Proc. IEEE Int. Conf. on Commun. 2007 ICC-2007*, Glasgow, United Kingdom, 2007 (doi:10.1109/ICC.2007.1046).

[18] A. M. Lyapunov, "The General Problem of the Stability of Motion", *Int. J. of Control*, vol. 55, no. 3, pp. 531–773, 1992 (doi: 10.1080/00207179208934253).

[19] J. Kleban and J. Warczyński, "Stabilność buforowanych pól komutacyjnych Closa", *Przegląd Telekomunikacyjny i Wiadomości Telekomunikacyjne*, no. 8–9, pp. 976–981, 2016 (in Polish).

[20] F. G. Foster, "On the stochastic matrices associated with certain queuing processes". *Ann. Math. Statistics.* vol. 24, np. 3, pp. 355–360, 1953.

[21] S. Meyn and R. Tweedie, *Markov Chains and Stochastic Stability*, New York, USA: Springer, 1993.

[22] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% Throughput in an Input-queued Switch", *IEEE Transact. on Commun.,* vol. 47, no. 8, pp. 1260–1267, 1999 (doi: 10.1109/26.780463).

Janusz Kleban is an Assistant Professor at the Poznan University of Technology (PUT), the Chair of Communication and Computer Networks of the Faculty of Electronics and Telecommunications. He received his M.Sc. and Ph.D. degrees in telecommunications from PUT in 1982 and 1990, respectively. His scientific interests include packet dispatching algorithms for single and multistage switching fabrics, photonic broadband switch architectures, optical switching systems, control algorithms for lightwave networks, and Network on Chip (NoC).

E-mail: janusz.kleban@put.poznan.pl
Faculty of Electronics and Telecommunications
Poznan University of Technology
Poznań, Poland

Jarosław Warczyński received his M.Sc. degree in Electrical Engineering (Control Engineering) in 1978 and the Ph.D. degree in Computer Since (Operations research) in 1983 from Poznan University of Technology, Poland. He is a faculty member at the University's Institute of Control, Robotics and Information Engineering. His general areas of research are in operations research, robotics, control engineering and currently in high speed packet switching and routing in Clos-network switches.

E-mail: jaroslaw.warczynski@put.poznan.pl
Faculty of Electrical Engineering
Poznan University of Technology
Poznań, Poland