

# Performance of Load Balancing Algorithms in Clos Packet Switches

Aleksandra Smiljanić, *Member, IEEE*

**Abstract**—The size of a single-hop cross-bar fabric is still limited by the technology, and the fabrics available on the market do not exceed the terabit capacity. A multihop fabric such as Clos network provides higher capacity by using smaller switching elements (SE). When the traffic load is balanced over the switches in a middle stage, all the traffic would get through the fabric, as long as the switch outputs are not overloaded. However, the delay that packets experience through the Clos switch depends on the granularity of flows that are balanced. We examine the maximum fabric utilization under which a tolerable delay is provided for various load balancing algorithms, and derive its general formula in terms of the number of flows that are balanced. We show that the algorithms which balance flows with sufficiently coarse granularity provide both high fabric utilization and delay guarantees to the most sensitive applications.

## I. INTRODUCTION

Clos circuit switch has been proposed by Clos in 1953 at Bell Labs [4]. Figure 1 shows the connections between switching elements (SE) in a symmetric Clos three-stage switch. This interconnection rule is: the  $x$ th SE in some switching stage is connected to the  $x$ th input of each SE in the next stage [4], [5], [6]. Here, all connections have the same bandwidths. It has been shown that a circuit can be established through the Clos switching fabric without rearranging existing circuits as long as the number of SEs in the second stage is at least twice the number of inputs of an SE in the first stage, i.e.  $l \geq 2 \cdot n$ . It has also been shown that a circuit can be established through the Clos switching fabric as long as the number of SEs in the second stage is no less than the number of inputs of an SE in the first stage, i.e.  $l \geq n$ . In the latter case, the number of required SEs and their total capacity are smaller due to the fact that the existing circuits can be rearranged. While the complexity of the switching fabric hardware is reduced, the complexity of the algorithm for a circuit setup is increased. In both cases, non-blocking property of the Clos architecture has been proven assuming the specific algorithms for circuit setup [6].

The Clos switching fabric can be used for increasing capacity of packet switches as well. The interconnection of SEs would be the same as in the circuit switch case. However, these SEs should be reconfigured in each cell time slot based on the outputs of outstanding cells. Here, packets are split into cells of a fixed duration, which is typically 50ns (64 bytes at 10Gb/s). A scheduling algorithm that configures SEs in a Clos packet switch implies higher processing complexity than the scheduling algorithms that configure a cross-bar packet switch [3], [8], [10], [11]. Few heuristics have been proposed to configure SEs in Clos packet switches [7], [9]. However, a

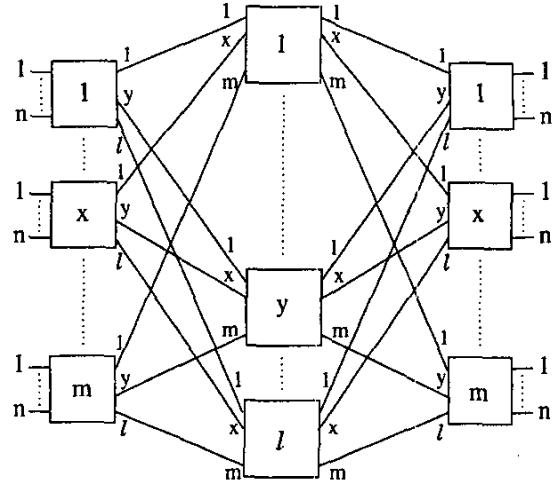


Fig. 1. Clos switching fabric

practical scheduling algorithm that provides non-blocking in these switches has not been designed. Algorithms for circuit setup in Clos circuit switches cannot be readily applied in Clos packet switches. First, all SEs should be synchronized on a cell-by-cell basis. Then, an implementation of the algorithm that rearranges connections on a cell-by-cell basis in SEs of a rearrangeable non-blocking Clos switch would be prohibitively complex [5]. So, the Clos fabric with the larger hardware,  $l = 2 \cdot n$ , is needed for a non-blocking packet switch.

Clos packet switches in which the traffic load is balanced across the SEs may provide non-blocking. Such an architecture has been described in [2], [14]. Turner showed that the architecture is non-blocking if the traffic of each multicast session is balanced over the SEs in Benes packet switch [14]. Here the multicast session carries the information between end users in the network. However, the delay that can be guaranteed to the flows has not been assessed. The worst-case delay increases with the number of flows that are separately balanced, because different flows may transmit cells over the same SEs at similar times. It will be shown that for a fixed tolerable delay, the allowed fabric utilization may unacceptably decrease as the number of flows increases. First, we will describe several options for load balancing of flows with different granularities, and then assess their performance. Either inputs or input SEs may balance traffic, and flows to either output SE or outputs may be balanced separately. Formula for the fabric utilization in terms of the number of flows, and tolerable delay will be derived. Based on this performance analysis, the adequate load

balancing algorithms will be proposed at the end.

## II. LOAD BALANCING ALGORITHMS IN THE CLOS PACKET-SWITCHES

Obviously, when cells reach the center SEs (SEs in the second stage), they are further routed according to their output addresses. So, load balancing can be only performed at the input SEs (SEs in the first stage). We will discuss four different load balancing algorithms. In the first load balancing algorithm, cells from some input bound for the particular output SE (SE in the third stage) are spread equally among center SEs. In the second case, cells from some input bound for the particular output are spread equally among center SEs. Then, the load can be balanced by input SEs: an arbiter associated with each input SE determines to which center SE a cell will be transmitted. So, in the third algorithm, cells transmitted from an input SE to some output SE would be spread equally across the center SEs. In the fourth algorithm, cells transmitted from an input SE to some output would be spread equally across the center SEs.

In the first load balancing algorithm, input  $i$ ,  $0 \leq i < N$ , has  $m$  different counters associated with different output SEs,  $c_{ij}$ ,  $0 \leq j < m$ . Here  $N = nm$  is the number of switch input and output ports. A cell arriving to input  $i$  and bound for the  $j$ th output SE will be marked to be transmitted through the  $c_{ij}$ th output of its SE, i.e. to be transmitted through the  $c_{ij}$ th center SE. Then, the counter in question is incremented modulo  $l$ , namely  $c_{ij} \leftarrow (c_{ij} + 1) \bmod l$ . In the second load balancing algorithm, input  $i$ ,  $0 \leq i < N$ , stores  $N$  counters associated with different switch outputs,  $c_{ij}$ ,  $0 \leq j < N$ . A cell arriving to input  $i$  and bound for the  $j$ th switch output will be marked to be transmitted through the  $c_{ij}$ th output of its SE, i.e. to be transmitted through the  $c_{ij}$ th center SE. Then, the counter in question is incremented modulo  $l$ . In the third load balancing algorithm, input SE  $i$ ,  $0 \leq i < m$ , stores  $m$  different counters associated with different output SEs,  $c_{ij}$ ,  $0 \leq j < m$ . A cell arriving to input SE  $i$  and bound for the  $j$ th output SE will be marked to be transmitted through the  $c_{ij}$ th output of its SE, i.e. to be transmitted through the  $c_{ij}$ th center SE. Then, the counter in question is incremented modulo  $l$ . In the fourth load balancing algorithm, input SE  $i$ ,  $0 \leq i < m$ , stores  $N$  counters associated with different switch outputs,  $c_{ij}$ ,  $0 \leq j < N$ . A cell arriving to input SE  $i$  and bound for the  $j$ th switch output will be marked to be transmitted through the  $c_{ij}$ th output of its SE, i.e. to be transmitted through the  $c_{ij}$ th center SE. Then, the counter in question is incremented modulo  $l$ .

Let us examine the blocking nature of a Clos packet switch based on a load balancing. Let  $SE_{ij}$  denote the  $j$ th SE in stage  $i$  throughout the text. In all algorithms, each input, or input SE, will transmit the traffic at equal rates through the connections from input to center SEs, and, consequently the rate,  $R_{12}$ , transmitted through any of these connections is:

$$R_{12} = \sum_{i' \in SE_{1i}} s_{i'} / l \leq n \cdot R / l, \quad (1)$$

where  $s_{i'}$  is the rate at which input  $i'$  sends the traffic. If  $r_{i'k}$  denotes the rate at which input  $i'$  sends the traffic to output

$k'$ , then the rate,  $R_{23}$ , transmitted through a connection from a center (second stage) SE to an output (third stage) SE, say  $SE_{3k}$ , is:

$$R_{23} = \sum_{i'} \sum_{k' \in SE_{3k}} r_{i'k'} / l \leq nR / l \quad (2)$$

using condition that the outputs are not overloaded. So, the maximum rate,  $R_c$ , supported by a connection in the fabric should fulfill:

$$R_c \geq n \cdot R / l, \quad (3)$$

because equality may be reached in (1,2). So, non-blocking is provided without link speedup if  $l \geq n$ .

## III. GENERALIZED PERFORMANCE ANALYSIS OF LOAD BALANCING ALGORITHMS

Traffic of each individual flow is balanced independently across the SEs. If there are many flows that transmit cells across some SE at the same time, the cells will experience long delay. Many applications, e.g. voice and video, require rate and delay guarantees. We will assess the worst case utilizations for balancing algorithms that provide rate and delay guarantees.

Time is divided into frames of  $F$  cells, and each input-output pair is guaranteed a specified number of time slots per frame, for example  $a_{ij}$  time slots are guaranteed to input-output pair  $(i, j)$ ,  $0 \leq i, j < N$ . Each input, and each output can be assigned at most  $F_u$  time slots per frame, i.e.

$$\sum_k a_{ik} \leq F_u, \quad \sum_k a_{ki} \leq F_u. \quad (4)$$

We will evaluate  $F_u$  in terms of  $F$  and  $N_f$  for which all cells of one frame pass the stage in the next frame when various load balancing algorithms are applied. Here  $N_f$  is the maximum number of flows passing through some connection that are separately balanced. We will assume that the fabric is non-blocking, i.e. that  $l = n$ .

We assume that there is a coarse synchronization in a switch, i.e. that at some point of time the input ports schedule cells belonging to the same frame. The delay that a cell may experience through Clos switch is three times the frame duration  $D = 3FT_c$ . The coarse synchronization may introduce an additional delay smaller than the frame duration, but may also simplify the controller implementation. Otherwise, SEs should give priority to the earlier frames which complicates their schedulers, also cell resequencing becomes more complex because the maximum jitter is increased.

Let  $F_c$  denote the number of cells per frame sent from a given input SE through a given center SE. We will calculate  $F_c \leq F$  in terms of  $F_u$ , and then the maximal utilization  $F_u / F$  of the connections from input to center SEs. Because of the symmetry, utilization is the same for the connections from center to output SEs, as will be explained. Let  $N_f$  further denote the largest number of flows sourced by  $SE_{1i}$  that pass through the links from this SE to center SEs; and,  $f_{ig}$ ,  $0 \leq g < N_f$ , denote the number of time slots per frame that are

guaranteed to the individual flows sourced by  $SE_{1i}$ . It follows:

$$\begin{aligned} F_c &\leq \sum_g \lceil f_{ig}/n \rceil \Rightarrow \\ F_c &< \sum_g f_{ig}/n + N_f \Rightarrow \\ F_c &< F_u + N_f, \end{aligned} \quad (5)$$

where  $\lceil x \rceil$  is the smallest integer no less than  $x$ , i.e.  $\lceil x \rceil < x + 1$ . Let us find the maximum number of cells sourced by  $SE_{1i}$  that may happen to be transmitted through the given center SE, say  $SE_{2j}$ . Assume that out of  $N_f$  flows sourced by  $SE_{1i}$ ,  $N_f - n$  flows are assigned one time slots per frame, and the remaining  $n$  flows are assigned  $\max(0, nF_u - (N_f - n))$  time slots per frame. If it happens that first cells in a frame of all flows are sent through  $SE_{2j}$ , the total number of cells per frame transmitted through  $SE_{2j}$  from  $SE_{1i}$  is maximized, and is easily shown to be:

$$\max(N_f, F_u + (n - 1) \cdot N_f/n - w), \quad (6)$$

where  $0 \leq w < n$ . Note that in this case  $F_c$  almost reaches the upper bound in (5) since  $F_u \geq N \gg n$ . The maximum number of cells transmitted per frame from a center to an output SE is the same if  $N_f$  is the number of flows bound to the output SE that pass through the links from center SEs to this output SE. In that case  $f_{kg}$ ,  $0 \leq g < N_f$ , is the number of cells guaranteed to flow  $g$  bound to  $SE_{3k}$ . Again, out of  $N_f$  flows bound for  $SE_{3k}$ ,  $N_f - n$  flows may transmit one cell per frame that pass through  $SE_{2j}$  etc. Since  $F_c \leq F$  for any of the internal connections in the fabric, it follows that

$$F_u \leq \max(\lceil N_f/n \rceil, F - (n - 1) \cdot N_f/n + w), \quad (7)$$

and the maximum utilization for which cells pass through the switch within designated frames is approximately:

$$U = \begin{cases} 1 - N_f/F & F \geq N_f \\ 0 & F < N_f, \end{cases} \quad (8)$$

where  $N_f$  is the maximum number of flows sourced by any input SE or bound to any output SE, i.e. the maximum number of flows passing through some fabric internal connection that are separately balanced.

We calculated the maximum utilization when different flows bound for the same SE are not desynchronized, so they might send cells within a given frame starting from the same center SE. Alternatively, flow  $g$  of  $SE_{1i}$  resets its counter at the beginning of a frame to  $c_{ig} = (i + g) \bmod n$ . We will calculate the maximum number of cells that are transmitted from  $SE_{1i}$  through  $SE_{2(n-1)}$  in the middle stage, and the same result would hold for any other center SE. The number of cells in flow  $g$  transmitted from  $SE_{1i}$  through  $SE_{2(n-1)}$  is  $\lfloor (f_{ig} + (i + g) \bmod n)/n \rfloor$ , where  $\lfloor x \rfloor$  is the smallest integer not greater than  $x$  i.e.  $\lfloor x \rfloor \leq x$ . So, the number of cells from  $SE_{1i}$  through  $SE_{2(n-1)}$  is:

$$\begin{aligned} F_c &= \sum_{0 \leq g < N_f} \lfloor (f_{ig} + (i + g) \bmod n)/n \rfloor \\ &\leq \sum_{0 \leq g < N_f} (f_{ig} + (i + g) \bmod n)/n \\ &\leq F_u + \lceil N_f/n \rceil \cdot (n - 1)/2 \approx F_u + N_f/2. \end{aligned} \quad (9)$$

Equality in (9) is reached iff:

$$f_{ig} = n - (i + g) \bmod n + n \cdot y_{ig}, \quad (10)$$

where  $y_{ig} \geq 0$  are integers. Values  $f_{ig}$  that satisfy condition (10) exist if it holds that:

$$\begin{aligned} nF_u &= \sum_{0 \leq g < N_f} f_{ig} \\ &\geq \sum_{0 \leq g < N_f} n - (i + g) \bmod n = \lceil N_f/n \rceil n(n + 1)/2 \Leftrightarrow \\ F_u &\geq \lceil N_f/n \rceil \cdot (n + 1)/2 \approx N_f/2. \end{aligned} \quad (11)$$

If inequality (11) holds, equality in (9) may be reached. Since  $F_c \leq F$ , it follows that:

$$\begin{aligned} F_c &= F_u + N_f/2 \leq F \Rightarrow \\ U_r &= F_u/F \leq 1 - N_f/(2F) \\ F &\geq N_f. \end{aligned} \quad (12)$$

However, if inequality (11) does not hold, then:

$$\begin{aligned} sz(z + 1)/2 &\leq nF_u < s(z + 1) \cdot (z + 2)/2 \Rightarrow \\ z &\approx \sqrt{2nF_u/s}, \end{aligned} \quad (13)$$

where  $s = \lceil N_f/n \rceil$ , and  $0 \leq z < n$  is an integer. It is easy to understand that  $F_c$  will be maximal for:

$$f_{ig} = \begin{cases} n - k & n - z \leq k = (i + g) \bmod n < n \\ 0 & 0 \leq (i + g) \bmod n < n - z. \end{cases} \quad (14)$$

Then,

$$F_c = sz \approx \sqrt{2F_u N_f}. \quad (15)$$

If (11) does not hold, from  $F_c \leq F$  follows that:

$$\begin{aligned} F_c &= \sqrt{2F_u N_f} \leq F \Rightarrow \\ U_r &= F_u/F \leq \min(N_f/(2F), F/(2N_f)). \end{aligned} \quad (16)$$

Note that formulas (12,16) hold if  $F_c$  is the number of cells transmitted from a center to an output SE and  $N_f$  is the number of flows bound to the output SE, while in the derivation  $f_{kg}$  denotes the number of cells in flow  $g$  transmitted to  $SE_{3k}$  starting from  $SE_{2j}$ . So, the maximum utilization when counters are reset each frame is:

$$U_r = \begin{cases} 1 - N_f/(2F) & F_u \geq N_f/2 \\ \min(N_f/(2F), F/(2N_f)) & F_u < N_f/2. \end{cases} \quad (17)$$

From (12,16,17), it follows that:

$$U_r = \begin{cases} 1 - N_f/(2F) & F \geq N_f \\ F/(2N_f) & F < N_f. \end{cases} \quad (18)$$

#### IV. PERFORMANCE OF LOAD BALANCING ALGORITHMS

It can be observed from our previous analysis that the performance of a load balancing algorithm depends on the number of flows that are separately balanced. Let  $N_f$  denote the number of balanced flows passing through some internal link. In the first algorithm  $N_f = N$ , because any input SE sources  $n^2 = N$  flows, and each of  $N$  inputs balances one flow for any output SE. In the second algorithm,  $N_f = nN$ , because any input SE sources  $nN$  flows, and each of  $N$  inputs balances  $n$  flows bound for any output SE. In the third

TABLE I  
SWITCH UTILIZATION [%]

D[ms] \ N	256	1024	4096
1	96.2	84.6	38.6
3	98.7	94.9	79.5
5	99.2	96.9	87.7

(a) Inputs balance output SE loads or input SEs balance output loads,  $T_c = 50\text{ns}$

D[ms] \ N	256	1024	4096
1	38.6	0	0
3	79.5	0	0
5	87.7	1.7	0

(b) Inputs balance output loads,  $T_c = 50\text{ns}$

TABLE II  
SWITCH UTILIZATION [%]: COUNTERS RESET EACH FRAME

D[ms] \ N	256	1024	4096
1	98.1	92.3	69.3
3	99.4	97.4	89.8
5	99.6	98.5	93.9

(a) Inputs balance SE output loads or input SEs balance output loads,  $T_c = 50\text{ns}$

D[ms] \ N	256	1024	4096
1	69.3	10.2	1.3
3	89.8	30.5	3.8
5	93.9	50.9	6.4

(b) Inputs balance output loads,  $T_c = 50\text{ns}$

algorithm,  $N_f = n$  because any input SE sources  $n$  flows, and each of  $n$  input SEs balances one flow for any output SE. In the fourth algorithm,  $N_f = N$  because any input SE sources  $n^2 = N$  flows, and each of  $n$  input SEs balances  $n$  flows for any output SE.

By substituting  $N_f$  in formula (8), we obtain the maximum utilizations for described load balancing algorithms:

$$\begin{aligned}
 U_1 = U_4 &= 1 - N/F \quad F \geq N \\
 U_2 &= \begin{cases} 1 - nN/F & F \geq nN \\ 0 & F < nN \end{cases} \\
 U_3 &\approx 1,
 \end{aligned} \quad (19)$$

because  $F < N$  is of no practical interest. So, the second load balancing algorithm is least efficient, while the third algorithm is most efficient. In order to increase the efficiency of the load balancing algorithms, the frame length should be increased. On the other side, the cell delay is proportional to the frame length. Assume that the maximum delay that can be tolerated by most sensitive applications such as interactive voice and video is  $D$ , and the cell time slot duration is  $T_c$ , then

$$F \leq D/(3T_c) \Rightarrow \quad (20)$$

$$\begin{aligned}
 U_1 = U_4 &= 1 - 3NT_c/D, \\
 U_2 &= \begin{cases} 1 - 3nNT_c/D & D \geq 3nNT_c \\ 0 & D < 3nNT_c \end{cases}
 \end{aligned} \quad (21)$$

Table I (a) shows the switch efficiency of the first and the third load balancing algorithms for various tolerable delays and switch sizes, while Table I (b) shows the switch efficiency of the second load balancing algorithm for the same parameters. It has been assumed that  $T_c = 50\text{ns}$ . One way packet delay that can be tolerated by interactive applications is around 150ms, but only 50-60ms of this allowed delay can be budgeted for the queueing. The switch delay below 3ms may be required for various reasons. For example, packets might pass multiple

packet switches from their sources to the destinations, and packet delays through these switches would add. Also, in order to provide flexible multicasting, the ports should forward packets multiple times through the packet switch, and the packet delay is prolonged accordingly [2], [12], [13], [14]. We see that the efficiency of the second balancing algorithm might decrease unacceptably as the switch size is increasing, and therefore the first and the third load balancing algorithms are superior.

If flows are balanced starting from different center SEs, the efficiency of load balancing could be increased. Namely, at the beginning of each frame, counters will be set to the appropriate values, e.g.  $c_{ij} = (i+j) \bmod n$ , where  $0 \leq i < N$ ,  $0 \leq j < n$  for the first load balancing algorithm,  $0 \leq i, j < N$  for the second algorithm,  $0 \leq i < n$ ,  $0 \leq j < N$  for the fourth algorithm. (Efficiency of the third algorithm is already close to 100%.) The guaranteed utilizations for the enhanced load balancing algorithms can be derived by substituting  $N_f$  in formula (18):

$$\begin{aligned}
 U_{r1} = U_{r4} &= 1 - N/(2F) \quad F \geq N, \\
 U_{r2} &= \begin{cases} 1 - nN/(2F) & F \geq nN \\ F/(2nN) & F < nN \end{cases}
 \end{aligned} \quad (22)$$

It follows that:

$$\begin{aligned}
 U_{r1} = U_{r4} &= 1 - 3NT_c/(2D), \\
 U_{r2} &= \begin{cases} 1 - 3nNT_c/(2D) & D \geq 3nNT_c \\ D/(6nNT_c) & D < 3nNT_c \end{cases}
 \end{aligned} \quad (23)$$

where  $D$  is the maximum delay that can be tolerated.

Table II shows the efficiency for the load balancing in which counters are reset every frame. We note that the efficiency of the second load balancing algorithm is improved, but, it is still low in large switches where cells bound for the particular output are spread equally across the center SEs. So, it is preferred that cells bound for the output SE are spread equally

across center SEs, or that input SEs spread cells across center SEs. However, the implementation of the algorithms where input SEs balance the traffic may be more complex, and, consequently, less scalable. First, inputs have to exchange the information with the SE arbiter. Secondly, counters of the arbiter should be updated  $n$  times per cell time slot, which may require advanced processing capability, and may limit the number of SE ports i.e. the total switch capacity.

## V. CONCLUSION

Clos packet switches are non-blocking when the load bound for either outputs or output SEs is balanced across the SEs in the middle stage. However, the fabric utilization under which delay requirements are met may be low in high-capacity switches. We calculated the utilization for the four load balancing algorithm in terms of the number of flows that are balanced, and various tolerable delays. The utilization was shown to be poor in large switches in which end-to-end sessions are balanced separately. However balancing the small number of flows readily provides required rate and delay guarantees.

## REFERENCES

- [1] T. E. Anderson, S. S. Owicki, J. B. Saxe, and C. P. Thacker, "High-speed switch scheduling for local-area networks," *ACM Transactions on Computer Systems*, vol. 11, no. 4, November 1993, pp. 319-352.
- [2] T. Chaney, J. A. Fingerhut, M. Flucke, J. S. Turner, "Design of a gigabit ATM switch," *Proceedings of INFOCOM 1997*, vol. 1, pp. 2-11.
- [3] H. J. Chao, "Saturn: A terabit packet switch using dual round-robin," *Proceedings of GLOBECOM 2000*, pp. 487-495.
- [4] C. Clos, "A study of non-blocking switching networks," *Bell Systems Technology Journal*, vol. 32, 1953, pp. 406-424.
- [5] J. Hui, *Switching and Traffic Theory for Integrated Broadband Networks*, Kluwer Academic Press 1990.
- [6] F. K. Hwang, *The mathematical theory of nonblocking switching networks*. World Scientific, 1998.
- [7] T. McDermott, and T. Brewer, "Large-scale IP router using a high-speed optical switch element," *OSA Journal on Optical Networking*, [www.osa-jon.org](http://www.osa-jon.org), July 2003, pp. 229-241.
- [8] N. McKeown *et al.*, "The Tiny Tera: A packet switch core," *IEEE Micro*, vol. 17, no. 1, Jan.-Feb. 1997, pp. 26-33.
- [9] E. Oki, Z. Jing, R. Rojas-Cessa, H. J. Chao, "Concurrent round-robin-based dispatching schemes for Clos-network switches," *IEEE/ACM Transactions on Networking*, vol. 10, no. 6, December 2002, pp. 830-844.
- [10] A. Smiljanić, "Flexible bandwidth allocation in terabit packet switches," *Proceedings of IEEE Conference on High Performance Switching and Routing*, June 2000, pp. 233-241.
- [11] A. Smiljanić, "Flexible Bandwidth Allocation in High-Capacity Packet Switches," *IEEE/ACM Transactions on Networking*, April 2002, pp. 287-293.
- [12] A. Smiljanić, "Scheduling of Multicast Traffic in High-Capacity Packet Switches," *IEICE/IEEE Workshop on High-Performance Switching and Routing*, May 2002, pp. 29-33.
- [13] A. Smiljanić, "Scheduling of Multicast Traffic in High-Capacity Packet Switches," *IEEE Communication Magazine*, November 2002, pp. 72-77.
- [14] J. S. Turner, "An optimal nonblocking multicast virtual circuit switch," *Proceeding of INFOCOM 1994*, vol. 1, pp. 298-305.