On the Speedup Required to Achieve 100% Throughput for Multicast over Crossbar Switches

Can Emre Koksal
Department of Electrical and Computer Engineering
The Ohio State University, Columbus, OH, 43210
e-mail: koksal@ece.osu.edu

Abstract— **We show an isomorphism between maximal matching for packet scheduling in crossbar switches and strictly non-blocking circuit switching in three-stage Clos networks. We use the analogy for a crossbar switch of size $n \times n$ to construct a simple multicast packet scheduler of complexity $\mathbf{O}(n \log n)$ based on maximal matching. We show that, with this simple scheduler, a speedup of $\mathbf{O}(\log n / \log \log n)$ is necessary to support 100% throughput for any admissible multicast traffic. If fanout splitting of multicast packets is not allowed, we show that an extra speedup of 2 is necessary, even when the arrival rates are within the admissible region for mere unicast traffic. Also we revisit some problems in unicast switch scheduling. We illustrate that the analogy provides useful perspectives and we give a simple proof for a well known result.**

I. Introduction

Applications requiring QoS support for *multicast traffic* remain to be important in small and large scale networks. The problem of providing quality of service guarantees for multicast traffic over crossbar switches has received a limited attention, despite the popularity of its counterpart for unicast traffic. One of the main reasons for this is the difficulty of the task. Indeed, it was shown in [1] that "optimal scheduling" of multicast packets is NP-hard over a crossbar switch and in [2] it was proved that the resource speedup necessary to achieve 100% throughput for all admissible multicast traffic grows unbounded with increasing switch size. These results hold even when the crossbar switch is *multicast capable*, i.e., it is capable of connecting an input to multiple outputs. It is also stated in [2] that "the numerical evaluation of the necessary speedup is prohibitive" and no scaling law has been given as to how the speedup scales with the switch size.

In this paper we first illustrate the difficulty of multicast scheduling by showing that a speedup of 2 is necessary to support 100% throughput to multicast traffic, even if the rates are within the admissible region of the mere unicast traffic. Then, we present a simple algorithm to provide 100% throughput for all admissible multicast traffic and specify a scaling law for the necessary speedup to achieve this task.

Our main tool in the development will be an analogy between middle stage switch configurations of three-stage Clos networks and schedules for a crossbar switch. A similar analogy was first exploited in [3] for certain set of TDMA schedulers. We illustrate that strict-sense non-blocking in a three-stage Clos network is analogous to *maximal matching* in the crossbar switch scheduling problem. Consequently the result [4] for unicast traffic that, maximal matching is sufficient to provide 100% throughput[1] with a speedup of 2, becomes straightforward. Further, applying a theorem given in [5] for multicast circuit switching in Clos networks, we show that the speedup necessary for 100% throughput for multicast traffic scales as $\mathrm{O}(\log n / \log \log n)$ with maximal matching, which has an associated complexity of $\mathrm{O}(n \log n)$.

II. Switch Model and Multicast Traffic

We consider the combined input and output queued (CIOQ) switch architecture with a single crossbar fabric. We assume that the crossbar fabric is *multicast capable*, i.e., at any given time, an input can be connected to multiple outputs simultaneously, but an output can only be connected to an input. We call a given set of connections, a *switch configuration*.

We define a time slot as the time in which a cell can be transmitted over a link. In case an internal speedup, s, is used, up to s switch configurations can be set up in a time slot and hence up to s cells can be transferred to an output. We call the time in which a switch configuration remains active, a *schedule slot*. Hence a schedule slot is $1/s$ of a time slot. We assume links with identical capacities and packets arriving over an input link are fragmented into fixed sized cells.

Each cell arriving at an input queue has a *fanout set*, i.e., the set of the output links that the cell needs to be forwarded to. Unicast cells have a fanout set of unit cardinality. To avoid head of the line (HOL) blocking [6], we assume the presence of virtual output queueing (VOQ) at each input for every possible fanout set. Further, virtual output queueing at a per fanout set level is referred to as *multicast virtual output queueing* (MC-VOQ) in [2]. Note that in an $n \times n$ switch, for a given input, there exist $2^n - 1$ possible fanout sets. Due to this exponential growth, MC-VOQ has issues of scalability and consequently it is merely a theoretical tool used to investigate the limitations of IQ switches under multicast traffic.

A scheduler may choose not to place an arriving cell with a certain fanout set F directly to the associated MC-VOQ. It is also possible that it duplicates the cell and place a copy to the MC-VOQ with a fanout set F' and the other copy to the MC-VOQ with a fanout set F'' such that $F' \cup F'' = F$ and $F' \cap F'' = \emptyset$. Hence, these two copies are transferred to the corresponding outputs at different times. We call this process

We assume that the cell arrivals are rate ergodic and each VOQ is associated with a certain cell arrival rate (after possible

[1] In fact [4] shows work conservation, which is stronger than 100% throughput.

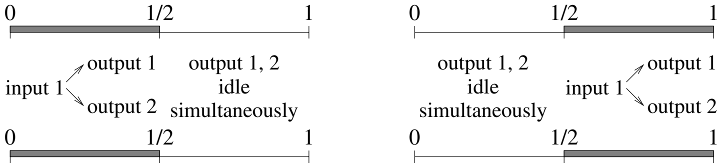


Fig. 1. No matter where the multicast flow is served, both outputs 1 and 2 will be idle simultaneously.

fanout splitting). For a given set of rates to be admissible, the total rate of cells arriving at each input link or destined to each output link cannot exceed 1 cell per time slot. Note that it may be possible that, after fanout splitting, the total rate of cells arriving at the VOQs of an input exceed 1 cell per time slot.

Let us first focus on the case with only unicast cells. In this scenario, at an input, there are n virtual output queues, one for each output. These n^2 VOQ arrival rates can be written in the form of an $n \times n$ *rate matrix* R. It can be shown (see e.g., [7], [8]) that 100% throughput is achievable if and only if R lies in the convex hull of the set of permutation matrices, i.e., there exists a frame, $\pi_1, \pi_2, \ldots, \pi_T$ of (containing possibly identical elements) permutation matrices such that $R \leqslant \frac{1}{T}\sum_{k=1}^{T} \pi_k$ for some possibly infinite T.

For multicast traffic, the *rate matrix* R is such that R_{ij} is, as a fraction of the link capacity, the rate at which input i wants to be connected to output j. Hence it is possible that $\sum_{j=1}^{n} R_{ij} > 1$. For instance suppose in every time slot only input 1 receives cells each of which is to be broadcast to all outputs. Then $\sum_{j=1}^{n} R_{1j} = n$. On the other hand, for all output j, $\sum_{i=1}^{n} R_{ij} \leqslant 1$, since no output can be oversubscribed.

The fundamental difference for the multicast traffic is that, the existence of a frame, $c_1, c_2, \ldots, c_T$ of configuration matrices for which $R \leqslant \frac{1}{T}\sum_{k=1}^{T} c_k$ does not necessarily imply 100% throughput is achievable. Consequently even with a multicast capable crossbar, speedup is necessary for 100% throughput. Following is an example.

Example 1: Consider the following rate matrix:

$$R = \begin{bmatrix} 0 & 0 & 0.5 \\ 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} + \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The first and the second component contain the rates of the unicast cells and the rate of the multicast cells respectively. Thus half of the cells arriving at input 1 are multicast cells with a fanout set of $\{1, 2\}$ and the other half are unicasts with the destination 3. Even though the sum of the rates in the first row of R is 1.5, the total rate of the cells arriving at the first input is 1. Indeed, no input or input or output link is oversubscribed under this traffic. In fact input links 1 and 2 are fully subscribed. Therefore to achieve 100% throughput without any speedup, at any point in time, input 2 must be connected to either output 1 or output 2, but not both, since all the cells are unicast at the second input. On the other hand input 1 needs to be connected to these two outputs simultaneously half the time to transfer multicast cells. This implies that these two outputs can be let free by the first input only half of the time as shown in Fig. 1, where the time period illustrated can be arbitrarily long. Consequently whenever the first input serves a multicast cell, input 2 must remain idle. However, since input 2 is fully utilized, some speedup is necessary.

The other alternative is the fanout splitting of the multicast cells. With fanout splitting, the total rate of cell arrivals at the VOQs of input 1 exceeds 1; consequently some speedup is necessary to accommodate them. We conclude that without a speedup, R is not supportable, with or without fanout splitting.

This example illustrates that multicast scheduling problem is much more complicated than unicast scheduling. Even with a multicast capable crossbar, some speedup is necessary for 100% throughput. This is valid despite the fact that matrix R can be written as a convex combination,

$$R = 0.5 \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 0.5 \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

of valid configurations matrices (R is in the convex hull of configuration matrices) for multicast enabled crossbar.

In this example, speedup $s = 1.5$ is necessary and sufficient for 100% throughput for the given traffic matrix. In [2] it was proved that the speedup necessary to achieve 100% throughput for all admissible multicast traffic grows unbounded with increasing switch size. It is also stated that “the numerical evaluation of the necessary speedup is prohibitive” and no scaling law has been given for the necessary speedup for 100% throughput. Also, finding the multicast schedule that works with the minimum necessary speedup is NP-hard as shown in [1]. There are obvious ways of simplifying multicast scheduling, such as ruling out fanout splitting. However, extra speedup is necessary to make up for the lost flexibility as shown in the following theorem.

Theorem 1: If fanout splitting of multicast cells is not allowed in an $n \times n$ crossbar switch, then a speedup of $2 - \frac{1}{n}$ is necessary to support multicast traffic for which the rate matrix R is a doubly stochastic matrix.

Before the proof of the theorem, note that, if fanout splitting is allowed, no speedup ($s = 1$) is required to support a doubly stochastic rate matrix. A complete fanout splitting is sufficient to achieve 100% throughout. Thus ruling out fanout splitting costs us some extra speedup or reduced throughput at a fixed speedup.

Proof: Consider the set of rates for which input 1 receives all unicast traffic with an equal rate of $1/n$ to every output.

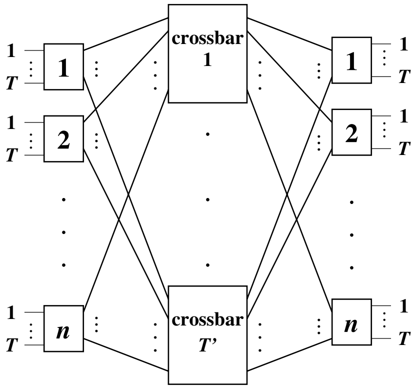


Fig. 2. Three stage Clos network.

Every other input receives cells once every n time slots to be multicast to all outputs (broadcast). For all i, j, $R_{ij} = 1/n$ and consequently the overall rate matrix is doubly stochastic.

Since fanout splitting of broadcast cells is not allowed, one schedule slot must be occupied for each broadcast cell arriving at inputs 2 to n. Along with a broadcast cell, no unicast cell can be scheduled from any of the input 1 VOQs. Thus extra n scheduling slots is necessary to accommodate input 1 traffic. As a result, to support this traffic, a total of $2n-1$ schedule slots is necessary in a span of n time slots, corresponding to a speedup of $2 - \frac{1}{n}$, completing the proof.

III. Crossbar Schedulers and Clos Networks

A three-stage Clos network is specified using three parameters (T, n, T') as shown in Fig. 2. There are T' middle stage crossbars of size $n \times n$ to connect n input stage switches of size $T \times T'$ to n output stage switches of size $T' \times T$.

Now consider a frame based scheduler with a schedule of period T. Let the speedup be $s = T'/T$, so the switch goes through T' configuration matrices, $c_1, \ldots, c_{T'}$ within the frame of T cell slots. The above scheduler is "time-analogous" to the following *circuit switching* Clos network:

If input i of the crossbar switch requires to to send k cells to output j within a frame of T slots, then, in the associated Clos network, k of the input links of input stage switch i require to make circuit connections to k of the output links of the output stage switch j. If c_m has a 1 in position (i, j), then the mth middle-stage crossbar connects input stage switch i to output stage switch j. This is illustrated in Fig. 3. Consequently the middle stage crossbars are set to have configurations $c_1, \ldots c_{T'}$. In a sense the middle-stage crossbars of the analogous Clos network replicates the entire sequence of T' configurations of the crossbar switch in space from the top to the bottom.

Any frame consisting of T' configurations corresponds to a *fixed circuit assignment* in the Clos network. The ratio of the number of middle stage crossbars to the number of input links of each input stage switch corresponds to the speedup $s = T'/T$. Note that in the Clos network, the size of the input and output stage switches grows with T. In this paper we construct the analogy for schedulers with a finite and periodic schedule of T' schedule matrices. In general it can be constructed for any given rate matrix R, as described in [9], using the concept of rate quantization.

There is one more thing we need to describe to complete the analogy for the case of multicast. In crossbar switching, in the case of fanout splitting, different copies of a multicast cell is served over different configuration matrices within a frame. If fanout splitting is not allowed, each multicast cell can be served with only a single configuration over the frame. In the analogous Clos network, to replicate fanout splitting, multicast connections in input stage switches are used as illustrated in Fig. 4. If fanout splitting is not allowed, input stage switches are only capable of unicast connections since each cell (input link) can go through only one middle stage crossbar. Note that Theorem 1 also shows that $2n-1$ middle stage crossbars is necessary to support multicast circuit switching in a Clos network in which only point to point connections are allowed at the input and output stage switches.

Based on our analogy, instead of asking the question "what is the necessary speedup to support all admissible multicast traffic in a crossbar switch?" we ask "what is the necessary number of middle stage switches to support multicast circuit switching in a Clos network?" The second question is also difficult and -to the best knowledge of the author- unanswered. However, things become simpler once we focus on strict sense non-blocking in Clos networks and it corresponds to an interesting set of schedulers based on maximal matching in switch sceduling as we discuss in the following section.

IV. Non-Blocking Switch Scheduling and Multicast Support

A network is strictly non-blocking if a connection between an idle input and an idle output can always be established, without the need for a rearrangement of the existing connections. Since there exists a middle stage crossbar to connect any idle input-output switch pair, to satisfy an incoming connection request, a simple search for that middle-stage crossbar will be sufficient. It is well known [10] that a three-stage Clos network, (T, n, T') is strictly non-blocking for unicast connections if and only if $T' \geqslant 2T - 1$.

The analogous scheduling interpretation of strictly non-blocking is interesting. At each input, there exists a frame of T unicast cells to be sent to over of the output links. For a given T, large enough for sufficient averaging of the arrival process, a speedup of $2 - \frac{1}{T}$ decouples the scheduling of cells at distinct inputs: Suppose an input has a cell to be sent to output j. It can search over the existing $2T-1$ configuration matrices for one, whose jth output is not already reserved by some input. Each input can take turns in completely assigning their cells to one of the scheduling matrices and at the end of the process, it is guaranteed that every single cell will be assigned to a matrix.

Alternatively, one can construct the $T' = 2T - 1$ configuration matrices one by one as follows. For each matrix, every input (takes order and) chooses one cell destined to an output for which another cell (from another input) is not already destined to. An input discards a matrix, only if it has no

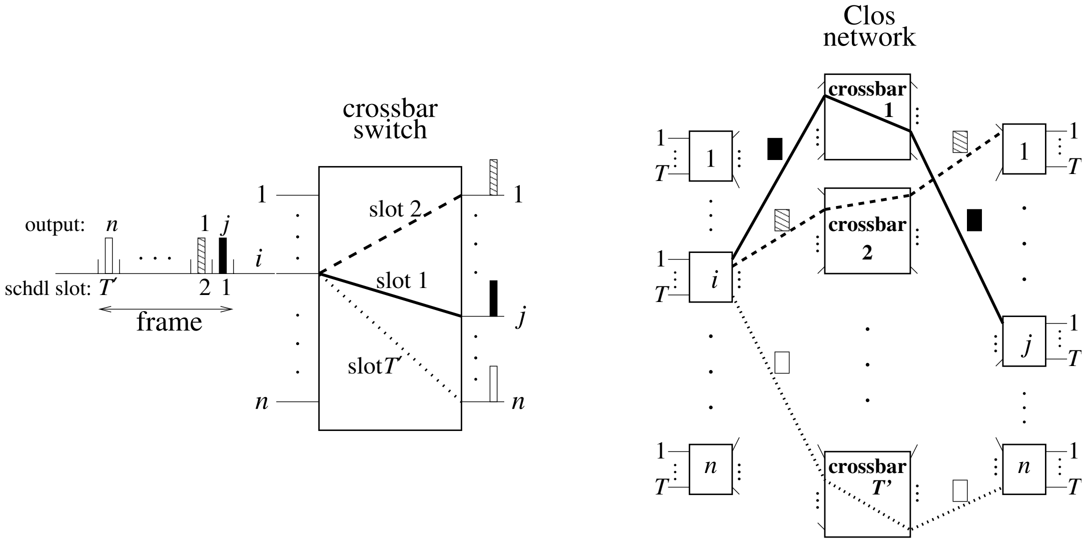


Fig. 3. The T' middle stage crossbars of the Clos network goes through the entire frame of T' configurations of the crossbar switch schedule.

cell for the available (not reserved by other inputs) outputs of the current matrix being constructed. This process leads to $2T - 1$ *maximal matchings* between inputs and the outputs since, for each matrix, no input and output both of which are idle are left unmatched if there exists a cell demanding that connection. Using the analogy, we just proved a result, which was initially showed in [4]: Since $T' = 2T - 1$ middle-stage crossbars is necessary and sufficient for strictly non-blocking Clos networks, a speedup of $\frac{T'}{T} = 2 - \frac{1}{T}$ is necessary and sufficient for 100% throughput for unicast traffic with maximal matching. Another important thing to note is that the complexity of maximal matching is $\mathrm{O}(N)$ per time slot for unicast traffic.

In complexity, there is no difference between multicast cells and unicast cells for the construction of a single configuration matrix using maximal matching. To construct each configuration matrix, every input takes turn to assign a cell to be scheduled for a transmission. This time there are multicast cells as well as unicast cells. Now suppose at an input there exists a multicast cell with a fanout set F; however only the set of outputs $F' \subset F$ is available for the matrix currently being constructed. Then the fanout set is split into two sets, F' and $F \setminus F'$. First $|F'|$ copies of the cell is multicast to the set of outputs F' and the remaining part is placed in the corresponding VOQs to be scheduled in another matrix.

In Theorem 1 of [5], it is shown that there exists a connection request pattern in a (T, n, T') Clos network such that an incoming feasible connection request cannot be met unless $T' \geqslant \Theta(T \log n / \log\log n)$. Consequently, in an $n \times n$ crossbar switch, a speedup of $s = \mathrm{O}(\log n / \log\log n)$ is necessary to achieve 100% throughput for multicast traffic with maximal matching. The number of configuration matrices constructed per time slot is proportional to s, the scheduling complexity is $\mathrm{O}(sn) \approx \mathrm{O}(n \log n)$ per time slot.

Unfortunately, parallel results do not exist for rearrangeably non-blocking multicast capable Clos networks. Therefore, $s = \mathrm{O}(\log n / \log\log n)$ is only an upper bound for the minimum necessary speedup, s_n, for 100% throughput for all admissible multicast traffic. However at a speedup s_n, the switch scheduling problem has been shown to be NP-hard, which is obviously not the case with maximal matching.

V. Summary and Conclusions

In this paper we use the analogy between cell scheduling in crossbar switches and circuit switching in a three-stage Clos network to study a number of issues involving multicast support over crossbar switches.

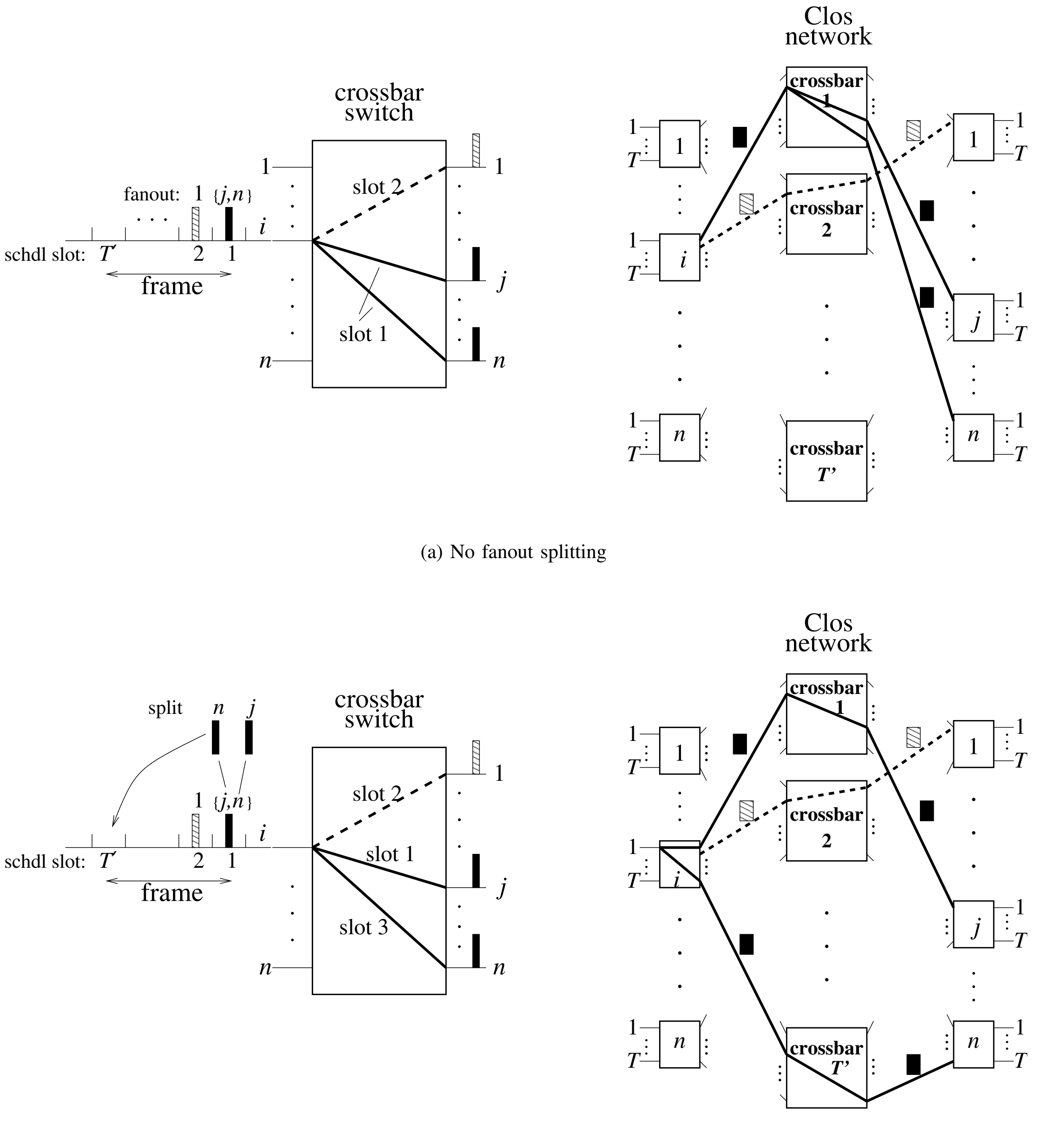
We showed that for a crossbar switch of size $n \times n$, using maximal matching, a speedup of $\mathrm{O}(\log n / \log\log n)$ is necessary to support 100% throughput for any admissible multicast traffic. Maximal matching is appealing for multicast traffic, because of its simplicity: The problem of multicast switch scheduling with the minimum necessary speedup is NP-hard, whereas the complexity of switch scheduling associated with maximal matching for multicast traffic is only $\mathrm{O}(n \log n)$ per time slot.

We also showed that if fanout splitting of multicast packets is not allowed, a speedup of 2 is necessary, even when the arrival rates are within the admissible region for unicast traffic. Thus, disabling the fanout splitting of multicast cells may not be an efficient solution for the complexity problem.

Also we revisit some problems in unicast switch scheduling. We illustrate that the well known result that “a speedup of 2 is necessary for 100% throughput for all admissible unicast traffic using maximal matching” becomes a straightforward by-product of the Clos network analogy with strict sense non-blocking.

References

[1] M. Andrews, S. Khanna, and K. Kumaran, “Integrated Scheduling of Unicast and Multicast Traffic in an Input-Queued Switch,” in *Proc. of the IEEE Infocom*, 1999.



(a) No fanout splitting

(b) Fanout splitting

Fig. 4. In Fig. 4(a) the cell with the fanout set $\{j, n\}$ is sent in the first time slot without fanout splitting. In Fig. 4(b), the same cell is split and sent to outputs j and n at different times. In the associated Clos network the input stage switches are not capable of multicast in the former scenario and are capable of multicast in the latter.

[2] M.A. Marsan, A. Bianco, P. Giaccone, E. Leonardi, and F. Neri, "Multicast Traffic in Input-Queued Switches:Optimal Scheduling and Maximum Throughput," *IEEE Transactions on Networking*, vol. 11, pp. 465–476, 2003.

[3] Anujan Varma and Suresh Chalasani, "An Incremental Algorithm for TDM Switching Assignments in Satellite and Terrestrial Networks," *IEEE Journal on Selected Areas in Communications*, vol. 10, pp. 364–377, February 1992.

[4] P. Krishna, N.S. Patel, A. Charny, and R.J. Simcoe, "On the speedup required for work-conserving crossbar switches," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 6, pp. 1057–1066, Jun 1999.

[5] Y. Yang and G.M. Masson, "The necessary conditions for clos type non-blocking multicast networks," *IEEE Trans. on Computers*, vol. 48, no. 11, pp. 1214–1227, 1999.

[6] M.G. Hluchyj, M.J. Karol, and S. Morgan, "Input versus output queueing on a space division switch," *IEEE Trans. on Communications*, vol. 35, pp. 1347–1356, Dec. 1987.

[7] W.J. Chen, C.S. Chang, and H.Y. Huang, "On Service Guarantees for Input Buffered Crossbar Switches: A Capacity Decomposition Approach by Birkhoff and von Neumann," in *Proceedings of IEEE IWQoS*, 1999.

[8] C.E. Koksal, R.G. Gallager, and C. Rohrs, "Rate Quantization and Service Quality for Variable Rate Traffic over Single Crossbar Switches," in *Proc. of the IEEE Infocom*, Hong Kong, Mar. 2004.

[9] Can Emre Koksal, "On the Speedup Required to Achieve 100% Throuhgput for Multicast Traffic over Crossbar Switches," Tech. Rep., Department of Electrical and Computer Engineering, OSU, Feb. 2008.

[10] Hui J. Y., *Switching and Traffic Theory for Integrated Broadband Circuits*, Kluwer Academic Publishers, Boston, MA, 1990.