Birkhoff-von Neumann Input-Buffered Crossbar Switches for Guaranteed-Rate Services

Cheng-Shang Chang Wen-Jyh Chen Hsiang-Yi Huang
Dept. of Electrical Engineering
National Tsing Hua University
Hsinchu 300, Taiwan, R.O.C.
Email: cschang@ee.nthu.edu.tw
{wjchen, huhuang}@gibbs.ee.nthu.edu.tw

Abstract

Based on a decomposition result by Birkhoff and von Neumann for a doubly substochastic matrix, in this letter we propose a scheduling algorithm that is capable of providing guaranteed-rate services for input-buffered crossbar switches. Our guarantees are uniformly good for all non-uniform traffic. The computational complexity to identify the scheduling algorithm is $O(N^{4.5})$ for an $N \times N$ switch. Once the algorithm is identified, its on-line computational complexity is $O(\log N)$ and its on-line memory complexity is $O(N^3 \log N)$.

Keywords

input-buffered switches, rate guarantees, scheduling, generalized processor sharing

This research is supported in part by the National Science Council, Taiwan, R.O.C., under Contracts NSC-88-2213-E007-004 and NSC-87-2219-E007-005. Part of the results in this letter was presented in IEEE IWQoS'99

I. Introduction

Due to their scalability, crossbar switches have received a lot of attention recently (see e.g., [5], [7] and references therein). An input-buffered crossbar switch with N input ports and N output ports has a segregated buffer for each input port. In such a switch, time is slotted and synchronized so that packets from different input buffers can be read out simultaneously within a time slot. In a time slot, a crossbar switch sets up a connection pattern corresponding to a permutation matrix. As a permutation matrix is a one-to-one mapping from input ports to output ports, packets destined to the same output ports cannot be transmitted at the same time in a crossbar switch.

In this letter, we propose an algorithm that is capable of providing uniform rate guarantees for all non-uniform traffic in a crossbar switch. To be precise, let $r_{i,j}$ be the rate assigned to the traffic from input i to output j for an $N \times N$ input-buffered crossbar switch, and $C_{i,j}(t)$ be the cumulative number of slots that are assigned to the traffic from input i to output j by time t. We show that if

$$\sum_{i=1}^{N} r_{i,j} \leq 1, \quad j = 1, 2, \ldots, N, \quad \text{and} \tag{1}$$

$$\sum_{j=1}^{N} r_{i,j} \leq 1, \quad i = 1, 2, \ldots, N, \tag{2}$$

then there is a scheduling algorithm such that

$$C_{i,j}(t) - C_{i,j}(s) \geq r_{i,j}(t-s) - s_{i,j}, \tag{3}$$

for all i, j, $s \leq t$, and some $s_{i,j} \leq N^2 - 2N + 2$. The two conditions in (1) and (2), called "no overbooking" conditions in [5], are intuitively clear as they simply state that neither the total rate to an output port nor the total rate coming out from an input port can be larger than 1. The computational complexity to identify a scheduling algorithm to guarantee (3) is $O(N^{4.5})$ for an $N \times N$ input-buffered crossbar switch. Once the algorithm is identified, its on-line computational complexity is $O(\log N)$ and its on-line memory complexity is $O(N^3 \log N)$.

II. The Main Result

Our main idea of providing rate guarantees in input-buffered crossbar switches is to use the capacity decomposition approach by Birkhoff [1] and von Neumann [10]. To explain the idea, let $R = (r_{i,j})$ be the rate matrix with $r_{i,j}$ being the rate requested by the traffic from input i to output j for an $N \times N$ input-buffered crossbar switch. A nonnegative matrix R that satisfies the conditions in (1) and (2) is known to be *doubly substochastic* (see e.g., Marshall and Olkin [8]). For a doubly substochastic matrix, there exists a set of positive numbers ϕ_k and permutation matrices P_k, $k = 1, \ldots, K$ for some $K \leq N^2 - 2N + 2$ that satisfies

$$R \leq \sum_{k=1}^{K} \phi_k P_k, \quad \text{and} \tag{4}$$

$$\sum_{k=1}^{K} \phi_k = 1. \tag{5}$$

An organized proof for the above Birkhoff-von Neumann decomposition can be found in Chapter 2 of the book [8] (Theorems 2.A.2 and 2.C.1). There, a detailed algorithm, originated from Dulmage and Halperin [4], is also given to obtain the set of positive numbers ϕ_k and permutation matrices P_k. The computational complexity of the Dulmage-Halperin algorithm is $O(N^{4.5})$. We note that the Dulmage-Halperin algorithm was also rediscovered by Inukai [6] for the time slot assignment problem in a satellite-switched time-division multiple access (SS/TDMA) system.

In view of (4) and (5), the amount of time that the crossbar switch sets its connection pattern according to the permutation matrix P_k should be proportional to ϕ_k. As time is slotted, we can use the Packetized Generalized Processor Sharing (PGPS) algorithm in Parekh and Gallager [9] (or the Weighted Fair Queueing (WFQ) in Demers, Keshav, and Shenkar [3]) to approximate the scheme.

Algorithm 1:

(i) Assign a class of tokens for each permutation matrix P_k, $k = 1, \ldots, K$.

(ii) At time 0, generate a token for each class. Assign the virtual finishing time of the first class k token by $F_k^1 = 1/\phi_k$. Sort these K tokens in an increasing order of the virtual finishing times.

(iii) The switch serves the tokens in an increasing order of the virtual finishing times. Once a class k token is served in a time slot, the switch sets its connection pattern according to the permutation matrix P_k for that time slot.

(iv) After the $(\ell - 1)^{th}$ class k token is served, the switch generates the ℓ^{th} class k token and assign the virtual finishing time by $F_k^\ell = F_k^{\ell-1} + 1/\phi_k$. Insert the newly generated token in the sorted token list and repeat the algorithm from (iii).

The complexity of Algorithm 1 is to insert every new token into the sorted token list. As there are K tokens in the sorted list and $K \leq N^2 - 2N + 2$, the computational complexity of Algorithm 1 is $O(\log N)$. Since it takes $N \log_2 N$ bits to describe a permutation matrix and there are $O(N^2)$ permutation matrices needed to be stored in Algorithm 1, the memory complexity of Algorithm 1 is $O(N^3 \log N)$.

Now we present the main result of this letter. Let $C_{i,j}(t)$ be the cumulative number of slots that are assigned to the traffic from input i to output j by time t. Also, let $p_{k,i,j}$ be the element at the (i,j) position of the permutation matrix P_k. Note that $p_{k,i,j}$ only takes values 0 and 1. Define $E_{i,j} = \{k : p_{k,i,j} = 1\}$ to be the subset of $\{1, 2, \ldots, K\}$ such that for all k in $E_{i,j}$, the permutation matrix P_k has a nonzero element at the (i,j) position.

Theorem 1 *For any rate matrix $R = (r_{i,j})$ that satisfies (1) and (2), Algorithm 1 is a scheduling policy that guarantees*

$$\sum_{k \in E_{i,j}} \phi_k(t-s) - s_{i,j} \le C_{i,j}(t) - C_{i,j}(s) \le \sum_{k \in E_{i,j}} \phi_k(t-s) + s_{i,j}, \tag{6}$$

where

$$s_{i,j} = \min[K, |E_{i,j}| + \sum_{k \in E_{i,j}} \phi_k(K-1)], \tag{7}$$

and $|E_{i,j}|$ is the number of elements in the set $E_{i,j}$.

Note that K and ϕ_k's are the parameters obtained from the Birkhoff-von Neumann decomposition. As $K \le N^2 - 2N + 2$, we have $s_{i,j} \le N^2 - 2N + 2$. Also, we have from (4) that

$$\sum_{k \in E_{i,j}} \phi_k = \sum_{k=1}^{K} \phi_k p_{k,i,j} \ge r_{i,j}. \tag{8}$$

The proof of Theorem 1 is based on the following lemma. Let τ_k^ℓ be the index of the time slot that the ℓ^{th} class k token is served, and $D_k(t)$ be the cumulative number of class k tokens served by time t. These two quantities are related via the following inversion formula

$$D_k(t) = \sup\{\ell : \tau_k^\ell \le t\}. \tag{9}$$

Lemma 2 *For all $k = 1, \ldots, K$,*

$$\lfloor \phi_k t \rfloor \le D_k(t) \le \lfloor \phi_k(t + K - 1) \rfloor, \tag{10}$$

where $\lfloor x \rfloor$ is the largest integer that is less than or equal to x.

Using the inequality $x - 1 < \lfloor x \rfloor \le x$, one has

$$\phi_k t - 1 < D_k(t) \le \phi_k(t + K - 1). \tag{11}$$

Proof. As Algorithm 1 serves the tokens in an increasing order of the virtual finishing times, in the worst case the ℓ^{th} class k token is served after all the tokens with virtual finishing times not greater than F_k^ℓ are served. Note that $F_k^\ell = \ell/\phi_k$. The total number of tokens that have virtual finishing times not greater than F_k^ℓ is

$$\ell + \sum_{j \ne k} \lfloor (\ell \phi_j)/\phi_k \rfloor. \tag{12}$$

As the algorithm serves a token per time slot, we have

$$\tau_k^\ell \leq \ell + \sum_{j\neq k} \lfloor (\ell\phi_j)/\phi_k \rfloor \leq \ell + \sum_{j\neq k} (\ell\phi_j)/\phi_k = \ell/\phi_k = F_k^\ell.$$

Thus, every token is served not later than its virtual finishing time. From the inversion formula in (9), we then have $D_k(t) \geq \sup\{\ell : F_k^\ell \leq t\} = \lfloor \phi_k t \rfloor$. We prove the second inequality in (10) by contradiction. Suppose that for some k and some t, $D_k(t) > \lfloor \phi_k(t + K - 1) \rfloor$. As $D_k(t)$ is an integer, this is equivalent to $D_k(t) \geq \lfloor \phi_k(t + K - 1)\rfloor + 1$. From the inversion formula in (9), the $\lfloor \phi_k(t+K-1)\rfloor + 1^{th}$ class k token must be served not later than t, i.e.,

$$\tau_k^{\lfloor \phi_k(t+K-1)\rfloor+1} \leq t.$$

As the algorithm serves the tokens in an increasing order of their virtual finishing times, those tokens with virtual finishing times less than $F_k^{\lfloor \phi_k(t+K-1)\rfloor+1}$ must also be served not later than t. From the strict inequality $\lfloor x \rfloor + 1 > x$, it follows that the total number of tokens served not later than t is strictly larger than (cf. (12))

$$\phi_k(t + K - 1) + \sum_{j\neq k} \lfloor (\phi_k(t + K - 1)\phi_j)/\phi_k \rfloor.$$

Since there are exact t tokens served by time t, we have

$$\begin{aligned} t &> \phi_k(t + K - 1) + \sum_{j\neq k} \lfloor (\phi_k(t + K - 1)\phi_j)/\phi_k \rfloor \\ &> \phi_k(t + K - 1) + \sum_{j\neq k} ((t + K - 1)\phi_j - 1) = t, \end{aligned}$$

and we reach a contradiction. ■

Proof. (Theorem 1) We first prove the lower bound in (6). Note that

$$C_{i,j}(t) = \sum_{k\in E_{i,j}} D_k(t).$$

Thus, it suffices to show that

$$\sum_{k\in E_{i,j}} D_k(t) - \sum_{k\in E_{i,j}} D_k(s) \geq (\sum_{k\in E_{i,j}} \phi_k)(t - s) - K, \tag{13}$$

and that

$$\sum_{k\in E_{i,j}} D_k(t) - \sum_{k\in E_{i,j}} D_k(s) \geq (\sum_{k\in E_{i,j}} \phi_k)(t - s) - |E_{i,j}| - \sum_{k\in E_{i,j}} \phi_k(K - 1). \tag{14}$$

We now show (13). As exactly one token is served in a time slot,

$$\sum_{k=1}^{K} D_k(t) = t, \quad \forall t. \tag{15}$$

Thus, we have from (15) and the lower bound in (11) that

$$\sum_{k\in E_{i,j}} D_k(t) = t - \sum_{k\notin E_{i,j}} D_k(t) \le \sum_{k\in E_{i,j}} \phi_k t + (K - |E_{i,j}|), \quad \forall t. \tag{16}$$

Using the lower bound in (11) and (16) yields

$$\begin{aligned}&\sum_{k\in E_{i,j}} D_k(t) - \sum_{k\in E_{i,j}} D_k(s) \ge \sum_{k\in E_{i,j}} (\phi_k t - 1) - \sum_{k\in E_{i,j}} \phi_k s - (K - |E_{i,j}|)\\ &= \sum_{k\in E_{i,j}} \phi_k (t-s) - K.\end{aligned}$$

To see (14), note from the lower bound and the upper bound in (11) that

$$\begin{aligned}&\sum_{k\in E_{i,j}} D_k(t) - \sum_{k\in E_{i,j}} D_k(s) \ge \sum_{k\in E_{i,j}} (\phi_k t - 1) - \sum_{k\in E_{i,j}} (\phi_k (s + K - 1))\\ &= \sum_{k\in E_{i,j}} \phi_k (t-s) - |E_{i,j}| - \sum_{k\in E_{i,j}} \phi_k (K-1).\end{aligned}$$

To prove the upper bound in (6), we simply interchange the upper bound and the lower bound used for s and t in the proof for the lower bound in (6). ■

References

[1] G. Birkhoff, "Tres observaciones sobre el algebra lineal," *Univ. Nac. Tucumán Rev. Ser. A*, Vol. 5, pp. 147-151, 1946.

[2] C.S. Chang, W.J. Chen and H.Y. Huang, "On service guarantees for input buffered crossbar switches: a capacity decomposition approach by Birkhoff and von Neumann," *IEEE IWQoS'99*, pp. 79-86, London, U.K., 1999.

[3] A. Demers, S. Keshav, and S. Shenkar, "Analysis and simulation of a fair queueing algorithm," in *Proc. SIGCOMM'89*, pp. 1-12, Austin, TX, Sept. 1989.

[4] L. Dulmage and I. Halperin, "On a theorem of Frobenius-König and J. von Neumann's game of hide and seek, *Trans. Roy. Soc. Canada III(3)*, Vol. 49, pp. 23-29, 1955.

[5] A. Hung, G. Kesidis and N. Mckeown, "ATM input-buffered switches with guaranteed-rate property," *Proc. IEEE ISCC'98*, Athens, pp. 331-335, 1998.

[6] T. Inukai, "An efficient SS/TDMA time slot assignment algorithm," *IEEE Trans. Commun.*, Vol. 27, pp. 1449-1455, 1979.

[7] S. Li and N. Ansari, "Input-queued switching with QoS guarantees," *IEEE INFOCOM'99*, pp. 1152-1159, New York, 1999.

[8] A.W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and Its Applications.* New York: Academic Press, 1979.

[9] A.K. Parekh and R.G. Gallager, "A generalized processor sharing approach to flow control in integrated service networks: the single-node case," *IEEE/ACM Transactions on Networking*, Vol. 1, pp. 344-357, 1993.

[10] J. von Neumann, "A certain zero-sum two-person game equivalent to the optimal assignment problem, " *Contributions to the Theory of Games,* Vol. 2, pp. 5-12, Princeton University Press, Princeton, New Jersey, 1953.