Enhanced Birkhoff–von Neumann decomposition algorithm for input queued switches

J.Li and N.Ansari

Abstract: The enhanced Birkhoff–von Neumann decomposition (EBVND) algorithm, a new class of scheduling arbitrators for input queued (IQ) crossbar switches that is based on the Birkhoff–von Neumann decomposition algorithm, is introduced. Theoretical analysis shows that the performance of EBVND is better than the Birkhoff–von Neumann decomposition algorithm in terms of throughput and cell delay, and can also provide rate and cell delay guarantees. Also, the weighted rate filling algorithm (WRFA), a new algorithm that can be ueed to construct doubly stochastic matrices from doubly substochastic matrices with less complexity and better fairness, is proposed. The wave front Birkhoff–von Neumann decomposition (WFBVND) algorithm and its simplified version WFBVND with log$N$ iterations (WFBVND-log$N$), the special cases of EBVND, are also introduced and evaluated. Simulations show that the WFBVND and WFBVND-log$N$ algorithms have much lower average cell delay as compared to the Birkhoff–von Neumann decomposition algorithm.

1 Introduction

Input queued (IQ) switching architecture is attractive for a high-speed network owing to its scalability. Nevertheless, a IQ switch using a single FIFO queue in each input has the problem of head-of-line (HOL) blocking that limits its throughput to approximately 58.6% [1]. Previous research shows that by adopting virtual output queueing (VOQ) [2], in which multiple VOQs directed to different outputs are maintained at each input, 100% throughput can be achieved under all admissible independent traffic [3]. However, few IQ scheduling algorithms can provide delay guarantees. The Slepian–Duguid [2] algorithm can guarantee cell delay under the (r, T) traffic model with a fixed schedule that is pre-computed when connections are set up. However, it has problems such as computational complexity and rate granularity limitation.

Recently, Chang *et al.* [4] proposed the Birkhoff–von Neumann decomposition algorithm, which can provide rate and cell delay guarantees, based on a decomposition result by Birkhoff and von Neumann for a doubly stochastic matrix. In this paper, we will introduce the enhanced Birkhoff–von Neumann decomposition (EBVND) algorithm, which is better than the Birkhoff–von Neumann decomposition algorithm in terms of throughput and cell delay, and can also provide rate and delay guarantees.

2 Background

Consider an $N \times N$ VOQ switch consisting of N inputs, N outputs, and a non-blocking switch fabric such as a crossbar. The packets, which may have variable length, are broken into fixed length cells when they arrive at the inputs. After the cells have crossed the fabric, they are reassembled to the original variable length packets. A time slot is defined as the time required to transmit a cell at the line rate. The VOQ directed to output j at input i is denoted by $Q_{i,j}$. If $Q_{i,j}$ is not empty, there will be a request from input i to output j. The basic objective of scheduling a VOQ switch is to find a contention-free match based on the connection requests, i.e., at most one input can be matched to each output, and *vice versa*. Let $\boldsymbol{S} = (S_{i,j})$ be the matching matrix, which indicates the match between inputs and outputs. If input i and output j are matched, then $S_{i,j} = 1$; otherwise, $S_{i,j} = 0$. At the end of the time slot, a cell is transmitted from input i to output j if $S_{i,j} = 1$ and $Q_{i,j}$ is not empty.

Suppose the rate assigned to the traffic from input i to output j is $r_{i,j}$, which is also the arrival rate of $Q_{i,j}$. The traffic is admissible if and only if the following inequalities are satisfied:

$$\sum_{j=0}^{N-1} r_{i,j} \leq 1, \forall i \quad (1)$$

$$\sum_{i=0}^{N-1} r_{i,j} \leq 1, \forall j \quad (2)$$

Define matrix $\boldsymbol{R} = (r_{i,j})$, then $\boldsymbol{R}$ is a doubly substochastic matrix. For any doubly substochastic matrix $\boldsymbol{R}$, there exists [4, 5] a doubly stochastic matrix $\tilde{\boldsymbol{R}} = (\tilde{r}_{i,j})$ such that $r_{i,j} \leq \tilde{r}_{i,j}$, $\forall i, j$. Matrix $\tilde{\boldsymbol{R}}$ is a doubly stochastic matrix if it satisfies

$$\sum_{j=0}^{N-1} \tilde{r}_{i,j} = 1, \forall i \quad (3)$$

and

$$\sum_{i=0}^{N-1} \tilde{r}_{i,j} = 1, \forall j \quad (4)$$

IEE Proceedings online no. 20010618

DOI: 10.1049/ip-com:20010618

Paper first received 17th February and in revised form 20th July 2001

The authors are with the Advanced Networking Laboratory, Electrical and Computer Engineering Department, New Jersey Institute of Technology, University Heights, Newark, NJ 07102, USA

A doubly stochastic matrix $\tilde{\boldsymbol{R}}$ can be expressed as a linear combination of permutation matrices [4]:

$$\tilde{\boldsymbol{R}} = \sum_k \phi_k \boldsymbol{P}_k \qquad (5)$$

where $\boldsymbol{P}_k$ is a permutation matrix, and $0 < \phi_k \leq 1$ such that $\Sigma_k \phi_k = 1$.

The Birkhoff–von Neumann decomposition algorithm schedules the cells by setting the matching matrix $\boldsymbol{S}$ to the permutation matrix $\boldsymbol{P}_k$ with probability ϕ_k according to a modified packetised generalised processor sharing algorithm [4]. Let $C_{i,j}(n)$ be the cumulative number of time slots for transmission that are assigned to $Q_{i,j}$ by time slot n. Denote $A_{i,j}(n)$ as the total number of cells arrived in $Q_{i,j}$ at the end of time slot n. Then the Birkhoff–von Neumann decomposition algorithm can guarantee

$$C_{i,j}(m) - C_{i,j}(n) \geq (m-n)r_{i,j} - u_{i,j} \qquad (6)$$

for all i, j, $m > n$, where $u_{i,j}$ is a real number less than or equal to $N^2 - 2N + 2$, if eqns. 1 and 2 are satisfied [4]. Eqn. 6 implies that if $A_{i,j}(n)$ conforms to $(\sigma_{i,j}, r_{i,j})$, i.e.

$$A_{i,j}(m) - A_{i,j}(n) \leq (m-n)r_{i,j} + \sigma_{i,j} \qquad (7)$$

then the cell delay from input i to output j is bounded by $\lceil(\sigma_{i,j} + u_{i,j})/r_{i,j}\rceil$ using the Birkhoff–von Neumann decomposition algorithm [4].

3 Enhanced Birkhoff–von Neumann decomposition algorithm

An algorithm to construct a doubly stochastic matrix $\tilde{\boldsymbol{R}}$ from a doubly substochastic matrix $\boldsymbol{R}$ is provided in [4], with a computational complexity of $O(N^3)$. In this paper, we introduce the weighted rate filling algorithm (WRFA) to perform this task.

Weighted rate filling algorithm (WRFA):

1. Define $p_i = 1 - \Sigma_{j=0}^{N-1} r_{i,j}$. Calculate p_i for all i.
2. Define $q_j = 1 - \Sigma_{i=0}^{N-1} r_{i,j}$. Calculate q_j for all j.
3. Calculate $\Delta = N - \Sigma_{i=0}^{N-1} \Sigma_{j=0}^{N-1} r_{i,j}$.
4. Let $\tilde{r}_{i,j} = r_{i,j} + p_i q_j / \Delta$.

Theorem 1: Matrix $\tilde{\boldsymbol{R}} = (\tilde{r}_{i,j})$ constructed from doubly substochastic matrix $\boldsymbol{R} = (r_{i,j})$ with WRFA is a doubly stochastic matrix.

Proof: Since $\boldsymbol{R} = (r_{i,j})$ is a doubly substochastic matrix, we have $p_i \geq 0$, $\forall i$, $q_j \geq 0$, $\forall j$, and $\Delta \geq 0$. Thus, $\tilde{r}_{i,j} = r_{i,j} + p_i q_i / \Delta \geq 0$, $\forall i, j$. For any j, we have

$$\sum_{i=0}^{N-1} \tilde{r}_{i,j} = \sum_{i=0}^{N-1} r_{i,j} + \sum_{i=0}^{N-1} \left(\frac{p_i q_j}{\Delta}\right)$$
$$= 1 - q_j + \frac{\sum_i (1 - \sum_j r_{i,j}) q_j}{\Delta} = 1$$

For any i, we also have $\Sigma_{j=0}^{N-1} \tilde{r}_{i,j} = 1$. Thus, matrix $\tilde{\boldsymbol{R}} = (\tilde{r}_{i,j})$ is a doubly stochastic matrix.

Compared to the original algorithm, WRFA is simpler and fairer. For example, when

$$\boldsymbol{R} = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 \end{bmatrix}$$

the original algorithm constructs the doubly stochastic matrix

$$\tilde{\boldsymbol{R}} = \begin{bmatrix} 0.4 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.4 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.4 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix}$$

Using WRFA, we get

$$\tilde{\boldsymbol{R}} = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$$

Apparently, WRFA is fairer than the original algorithm since it shares the unreserved bandwidth more evenly among the VOQs. The complexity of WRFA is $O(N^2)$, which is smaller than the original one.

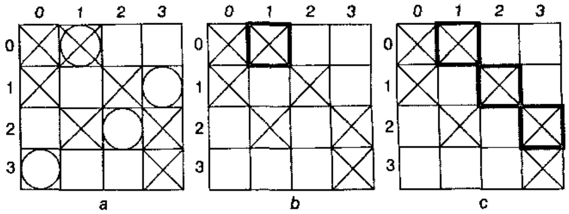

Fig. 1 *Birkhoff–von Neumann decomposition algorithm*
a Request graph and selected permutation matrix
b One cell is scheduled by Birkhoff–von Neumann decomposition algorithm
c Two more cells can be scheduled by filling 'holes'

We observed that the Birkhoff–von Neumann decomposition algorithm may select the empty VOQs for transmission because in certain time slot it sets the crossbar connection solely according to the permutation matrix which is obtained from $\tilde{\boldsymbol{R}}$, and pays no attention to the current occupancy of VOQs. Thus, it is not surprising to see that the average cell delay of the Birkhoff–von Neumann decomposition algorithm is much larger than other algorithms, such as the oldest cell first (OCF) [3]. For example, Fig. 1*a* shows the 4 × 4 VOQs in matrix form, where every square box represents a VOQ, and the non-empty VOQ is filled with a cross. Suppose the permutation matrix selected by the Birkhoff–von Neumann decomposition algorithm at the current time slot is

$$\boldsymbol{P} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

The non-zero elements of $\boldsymbol{P}$ are represented by circles in Fig. 1*a*. Among all the VOQs selected by $\boldsymbol{P}$, only VOQ $Q_{0,1}$ is non-empty, which is shown by a thick border box in Fig. 1*b*. Hence, only one cell can be scheduled by the Birkhoff–von Neumann decomposition algorithm in the current time slot. However, two more VOQs, $Q_{1,2}$ and $Q_{2,3}$, can actually send cells across the fabric in the current time slot without removing the cells scheduled by the Birkhoff–von Neumann decomposition algorithm, as shown in Fig. 1*c*.

Based on the above observation, the enhanced Birkhoff–von Neumann decomposition (EBVND) algorithm matches the inputs and outputs which are not matched by the Birkhoff–von Neumann decomposition algorithm, and attempts to make a maximal match in every time slot. Many algorithms such as parallel iterative matching (PIM) [2] and wrapped wave front arbiter (WWFA) [6], can be used to fill the 'holes' left by the Birkhoff–von Neumann decomposition algorithm. Thus, EBVND will have a higher online computational complexity than the Birkhoff–von Neumann decomposition algorithm, but is expected to have a better performance.

Suppose IQ switch B uses the Birkhoff–von Neumann decomposition algorithm while IQ switch E uses EBVND. Denote $Q_{i,j}^B$ and $Q_{i,j}^E$ as the $Q_{i,j}$ of switches B and E, respectively. Let $T_{i,j}^B(n)$ be the cumulative number of cells dequeued from $Q^B_{i,j}$ by the end of time slot n, and $T^E_{i,j}(n)$ be the number dequeued from $Q^E_{i,j}$. Define $T_{i,j}(n, m) = T_{i,j}(m) - T_{i,j}(n)$.

Lemma 1: For any integer $m > n$, if both $Q^B_{i,j}$ and $Q^E_{i,j}$ are constantly backlogged from time slot $n + 1$ to time slot m, then $T^E_{i,j}(n, m) \geq T^B_{i,j}(n, m)$.

Proof: When both $Q^B_{i,j}$ and $Q^E_{i,j}$ are not empty at a certain time slot, if a cell is dequeued from $Q^B_{i,j}$, then a cell must be dequeued from $Q^E_{i,j}$ according to the definition of EBVND.

Let $L^E_{i,j}(n)$ and $L^B_{i,j}(n)$ be the length of $Q^E_{i,j}$ and $Q^B_{i,j}$ by the end of time slot n, respectively.

Theorem 2: If identical traffic is fed into switch B and E concurrently and no cell is dropped, then $L^E_{i,j}(n) \leq L^B_{i,j}(n)$ for any i, j and n.

Proof: Consider any i, j and n, if $L^E_{i,j}(n) = 0$, then the theorem is proved, because $L^B_{i,j}(n) \geq 0$. If $L^E_{i,j}(n) > 0$, let $n_0 < n$ be the largest number such that $L^E_{i,j}(n_0) = 0$. That is, from time slot $n_0 + 1$ to time slot n, $Q^E_{i,j}$ is constantly backlogged. Denote $T^B_{i,j}(n_0, n)_{backlog}$ as the number of cells dequeued from $Q^B_{i,j}$ between time slot $n_0 + 1$ and n inclusively when $Q^B_{i,j}$ is constantly backlogged during $(n_0, n]$. From lemma 1, we know that $T^E_{i,j}(n_0, n) \geq T^B_{i,j}(n_0, n)_{backlog}$. Thus, we have $T^E_{i,j}(n_0, n) \geq T^B_{i,j}(n_0, n)$, because $T^B_{i,j}(n_0, n)_{backlog} \geq T^B_{i,j}(n_0, n)$. Since $L^E_{i,j}(n_0) = 0$, $L^E_{i,j}(n_0) \leq L^B_{i,j}(n_0)$. From time slot $n_0 + 1$ to n, the same number of cells are enqueued into $Q^B_{i,j}$ and $Q^E_{i,j}$, but more or the same number of cells are dequeued from $Q^E_{i,j}$. Thus $L^E_{i,j}(n) \leq L^B_{i,j}(n)$ for any i, j and n.

Denote $D_c{}^B$ as the delay of a certain cell c in switch B, and $D_c{}^E$ as the delay of cell c in switch E.

Theorem 3: Assume all the VOQs in switch B and E are FIFOs. If identical traffic is fed into switch B and E concurrently and no cell is dropped, then $D_c{}^E < D_c{}^B$ for any cell c.

Proof: If a cell c which is directed to output j arrives in input i at time slot n, then both $Q^B_{i,j}$ and $Q^E_{i,j}$ will be backlogged until c is scheduled. At time slot n, $L^B_{i,j}(n) \geq L^E_{i,j}(n)$. Assume cell c departs from switch E at time slot n_1, then $T^E_{i,j}(n, n_1) \geq T^B_{i,j}(n, n_1)$. Since the arrivals of B and E are identical, cell c cannot leave switch B before time slot n_1, if all the VOQs are FIFOs. So, $D_c{}^E < D_c{}^B$ for any cell c.

Theorems 2 and 3 imply that EBVND is better than (in the worst case at least as good as) the Birkhoff–von Neumann decomposition algorithm in terms of throughput and cell delay guarantees. Thus, EBVND does provision QoS guarantees, since the Birkhoff–von Neumann decomposition algorithm was proven to provision QoS guarantees [4].

The wave front Birkhoff–von Neumann decomposition (WFBVND) algorithm, which is a special case of EBVND, finds more pairs in a match using a method similar to WWFA [6]. WFBVND divides a time slot into N phases. Assume $\boldsymbol{P}$ is the permutation matrix selected by the Birkhoff–von Neumann decomposition algorithm. In the lth phase, WFBVND calculates matrix $V_l = (V_{l,i,j})$ where $V_{l,i,j} = P_{(i+l)\mathrm{mod}N,j}$. During the lth phase, WFBVND checks the VOQs corresponding to the non-zero elements of V_l, and adds the non-empty VOQs in the match if both its input and output are unmatched. Fig. 2 shows an example, where the VOQs filled with crosses indicate the non-empty VOQs, the VOQs filled with circles indicate that the corresponding elements of V_l are 1, and the VOQs with thick border are those scheduled to transmit cells. The online computational complexity of WFBVND is $O(N^2)$.

The complexity of $O(N^2)$ may be costly for high-speed implementation. WFBVND with $\log N$ iterations (WFBVND-log$N$) is thus introduced as a simplified version of WFBVND in order to reduce the complexity. It differs from WFBVND by only having the first log$N$ phases of WFBVND. Since WFBVND-log$N$ runs less phases than WFBVND, its performance is expected to be worse than WFBVND, but still be better than the Birkhoff–von Neumann decomposition algorithm, since it is the special case of EBVND. The online computational complexity of WFBVND-log$N$ is $O(N\log N)$.

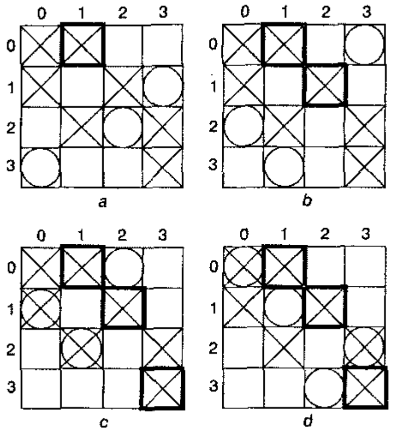

Fig. 2 *WFBVND algorithm*
a Phase 0
b Phase 1
c Phase 2
d Phase 3

4 Discussion and simulations

The performance of the new algorithms, together with that of some existing algorithms, were simulated in a 16 × 16 IQ switch. In the simulations, 256 i.i.d. flows, each belonging to a different input–output pair, were created by the Bernoulli model and filtered by a leaky bucket method. Therefore, traffic $A_{i,j}$ conforms to $(\sigma_{i,j}, r_{i,j})$ for all i, j, where $\sigma_{i,j}$ is set to be $1000r_{i,j}$. As a result, the delay bound is $1000 + \lceil u_{i,j}/r_{i,j} \rceil$ time slots for the Birkhoff–von Neumann decomposition algorithm, WFBVND, and WFBVND-log$N$.

Fig. 3 shows the distribution of the percentage of cells which experience various delays over the switch using the Birkhoff–von Neumann decomposition, OCF and WFBVND algorithms under the given traffic with a total traffic load of 84%. It shows clearly that the cell delay of the Birkhoff–von Neumann decomposition algorithm is much larger than that of OCF, while for WFBVND the delay is quite close to that of OCF.

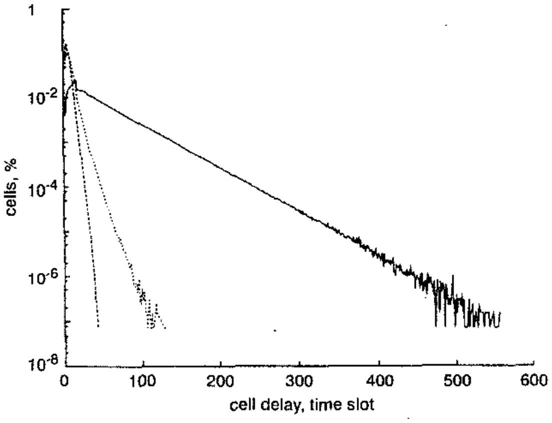

Fig. 3 *Cell delay distribution under 84% traffic load*
——— Birkhoff–von Neumann
– – – – OCF
.... WFBVND

Fig. 4 shows the average cell delay of the Birkhoff–von Neumann decomposition, WFBVND, WFBVND-log$N$ and OCF against the traffic load under i.i.d. (σ, r) traffic. Fig. 5 shows the variance of cell delay against the traffic load under the same traffic model. Figs. 4 and 5 indicate that the average cell delay and variance of WFBVND is much smaller than that of the Birkhoff–von Neumann decomposition algorithm, and close to OCF. With a reduced complexity, the average delay and the delay variance of WFBVND-log$N$ are also significantly smaller than those of the Birkhoff–von Neumann decomposition algorithm.

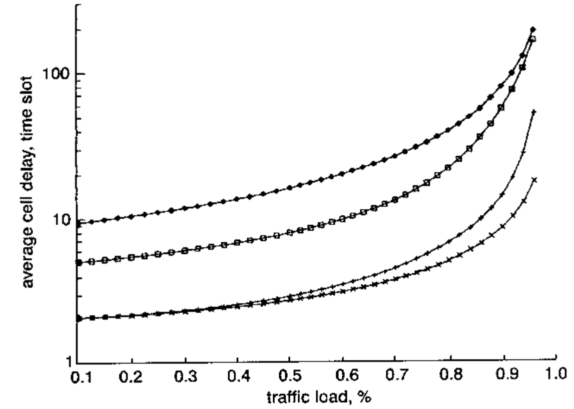

Fig. 4 *Average cell delay against traffic load under i.i.d. (σ, r) arrival*
—◇— Birkhoff–von Neumann
—+— WFBVND
—□— WFBVND-log$N$
—×— OCF

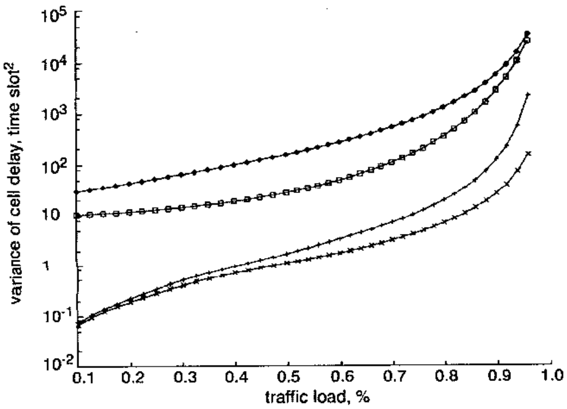

Fig. 5 *Variance of cell delay against traffic load under i.i.d. (σ, r) arrival*
—◇— Birkhoff–von Neumann
—+— WFBVND
—□— WFBVND-log$N$
—×— OCF

The algorithms are also tested under unbalanced traffic load. In the 256 flows, half of them, which are selected randomly, are set to be inactive, and the active flows are assigned rates randomly under the admission condition. Table 1 shows the average cell delay of all the active flows from input 0 to the outputs using BVND, WFBVND and WFBVND-log$N$, in which $D_{i,j}$ is the average delay of the flow from input i to output j. Table 2 tabulates the weighted average cell delay, variance of cell delay and average queue length of all the flows. The results demonstrate that our proposed algorithms have also achieved much smaller average cell delay, cell delay variance and average queue length in this traffic load.

Table 1: Average cell delay of the active flows in input 0

Algorithm	$D_{0,0}$	$D_{0,2}$	$D_{0,4}$	$D_{0,11}$	$D_{0,12}$	$D_{0,13}$	$D_{0,14}$	$D_{0,15}$
BVND	9.3	975	968	984	916	919	983	969.0
WFBVND-log$N$	8.6	4.6	11.2	984	905	552	512	138.0
WFBVND	6.6	2.9	4.0	3.5	3.4	5.1	25	107.7

Table 2: Performance of BVND and EBVND under unbalanced traffic load

Algorithm	Average cell delay (timeslot)	Variance of cell delay (timeslot × timeslot)	Average queue length (cell)
BVND	826	1.1×10^5	31
WFBVND-log$N$	296	8.3×10^4	11
WFBVND	22.6	4.1×10^3	0.43

5 Conclusions

A new class of IQ scheduling algorithms with rate and delay guarantees is proposed in this paper. Specifically, the performance of WFBVND and WFBVND-log$N$ is compared with that of OCF and the Birkhoff–von Neumann decomposition algorithm. It has been demonstrated both by simulations and theoretical analysis that the new algorithms can achieve much smaller average cell delay and delay variance, as well as provide QoS guarantees.

6 Acknowledgments

This work has been supported in part by Lucent Technologies, the New Jersey Commission on Science and Technology via the New Jersey Center for Wireless Telecommunications, and the New Jersey Commission on Higher Education via the NJI-TOWER Project.

7 References

1 KAROL, M., HLUCHYI, M., and MOGAN, S.: 'Input versus output queueing on a space-division packet switch', *IEEE Trans. Commun.*, 1987, **35**, (12), pp. 1347–1356

2 ANDERSON, T., OWICKI, S., SAXE, J., and THACKER, C.: 'High speed switch scheduling for local area networks', *ACM Trans. Comput. Syst.*, 1993, **11**, (4), pp. 319–352

3 McKEOWN, N., MEKKITTIKUL, A., ANANTHARAM, V., and WALRAND, J.: 'Achieving 100% throughput in an input-queued switch', *IEEE Trans. Commun.*, 1999, **47**, (8), pp. 1260–1267

4 CHANG, C.S., CHEN, W.J., and HUANG, H.Y.: 'On service guarantees for input buffered crossbar switches: a capacity decomposition approach by Birkhoff and von Neumann'. Proceedings of IEEE IWQoS'99, London, UK, 1999, pp. 79–86

5 von NEUMANN, J.: 'A certain zero-sum two-person game equivalent to the optimal assignment problem' *in* 'Contributions to the theory of games - vol. 2' (Princeton University Press, Princeton, NJ, USA, 1953), pp. 5–12

6 TAMIR, Y., and CHI, H.C.: 'Symmetric crossbar arbiters for VLSI communication switches', *IEEE Trans. Parallel Distrib. Syst.*, 1993, **4**, (1), pp. 13–27