Packet Delay-Aware Scheduling in Input Queued Switches

Yihan Li, Shivendra Panwar, H. Jonathan Chao and Jong ha Lee
Electrical and Computer Engineering Department, Polytechnic University, Brooklyn, NY
Email: yli@photon.poly.edu, panwar@catt.poly.edu, chao@poly.edu, jlee25@utopia.poly.edu

***Abstract*— Virtual Output Queuing is widely used by high-speed packet switches to overcome head-of-line blocking. This is done by means of matching algorithms. In fixed-length VOQ switches, variable-length IP packets are segmented into fixed-length cells at the inputs. When a cell is transferred to its destination output, it will stay in the reassembly buffer and wait for the other cells of the same packet before the entire packet can depart the system. The delay a packet suffers in the system includes the waiting time in the VOQ, the widely studied cell delay, and the waiting time at the output reassembly buffer, the reassembly delay often ignored in many papers. Among all existing matching algorithms, Maximum Weight Matching (MWM) has the lowest average *cell delay*. In this paper, we investigate the average *packet delay*, one of the key performance measure for an input buffered packet switch. A new class of matching algorithms, PDA-MWM, is defined and proved to be stable under all admissible traffic. Three PDA-MWM matching algorithms are studied by simulation. We show that, in order to achieve low packet delay, there is a tradeoff between the cell delay performance and the reassembly delay performance. If both of them are carefully considered, a matching scheme can greatly reduce the packet delay as compared to MWM.**

***Index Terms*— Switching, scheduling, Virtual Output Queueing, stability, delay performance.**

I. Introduction

SWITCHING technology continues to be one of the bottlenecks in the development of broadband networks as the traffic continues to double every year. Fixed-length switching technology is widely accepted as an approach to achieving high switching efficiency for high-speed packet switches. Packet switches based on Input Queuing (IQ) are desirable for high speed switching, since the internal operation speed is only moderately higher than the input line speed. However, an Input Queuing switch has a critical drawback [1]: the throughput is limited to 58.6% due to the head-of-line (HOL) blocking phenomena. Output Queuing (OQ) switches have optimal delay-throughput performance for all traffic distributions, but the N-times speed-up for the memory operation speed limits the scalability of this architecture. Virtual Output Queuing (VOQ) is used to overcome these drawbacks and combine the advantages of the Input Queuing and the Output Queuing. In a VOQ switch, each input maintains N queues, one for each output. By using VOQ and an appropriate matching algorithm, no additional speedup is required and HOL blocking can be eliminated.

This work is supported in part by the New York State Center for Advanced Technology in Telecommunications (CATT), and also in part by the National Science Foundation under grants ANI0081527 and ANI0081357.

Matching algorithms are used to resolve output contention by scheduling the input-output connections in each time slot. One key performance of a matching algorithm is the throughput. It has been proved that by using a *maximum weight matching algorithm* (MWM), 100% throughput can be reached for independent, identically distributed (i.i.d.) arrivals (uniform or nonuniform) [2], [3]. Because of the high complexity of MWM, *maximal matching algorithms* [4], [5], [6], [7], [8], [9], [10], [13] have been proposed to approximate MWM, but they cannot guarantee 100% throughput without speedup. Recently, some matching algorithms [11], [12], [13], [14], which are not MWM but can still guarantee 100% throughput without speedup have been devised.

Another important performance measure for a matching algorithm is the average delay. In fixed-length switches, variable-length IP packets are segmented into fixed-length cells at the inputs by the Input Segmentation Module (ISM), and then the cells are placed in the corresponding VOQ. When a cell is transferred to its destination output, it will stay in the Output Reassembly Module (ORM), and wait for the other cells of the same packet. After the complete reception of all the cells of the same packet, these cells will be reassembled into a packet. The delay a packet suffers before it is delivered to its destination output line includes the waiting time in the VOQ, and the waiting time at the output reassembly buffer. The waiting time in the VOQ is actually the widely studied cell delay, while the reassembly delay is often ignored in many papers. Among all existing matching algorithms, MWM is conjectured to have the optimal cell delay performance. However, the packet delay performance has not been well studied in previous work. In [13] and [14], a simple matching algorithm, HE-iSLIP, has been shown to achieve packet delay performance comparable to MWM. In this paper, we investigate the issue of how to improve the packet delay performance in VOQ switches, and present a new class of stable matching algorithms which have the same implementation complexity as MWM and can provide lower average packet delay than MWM. Simulation result gives us the important insight that, in order to achieve low packet delay, there is a tradeoff between the cell delay and reassembly delay. When both of them are carefully considered, the average packet delay of a matching algorithm can be much lower than that of MWM. Note that we do not look at the implementation complexity in this paper and solely focus on the issue of how much the packet delay performance can be improved.

In section II, a new class of matching algorithms, Packet

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

Delay-Aware MWM (PDA-MWM) is presented. We prove that PDA-MWM is stable under any admissible arrival traffic. In section III, three member algorithms of the PDA-MWM class are investigated, and their delay performance is compared to that of MWM by simulation. We show that all these schemes can efficiently reduce the average packet delay by carefully selecting functions and parameters used in the schemes. Section IV concludes this paper.

II. Packet Delay-Aware MWM

At time t, let $Q(t) = [q_{ij}(t)]_{N\times N}$, where $q_{ij}(t)$ is the queue length of VOQ_{ij} at time t. The weight of a schedule $M(t)$, which is the sum of the lengths of all matched VOQs, is denoted by

$$W(t) = \langle M(t), Q(t)\rangle. \tag{1}$$

Theorem 1 ([15]) Let $W^*(t)$ denote the weight of maximum weight matching scheduling at time t, with respect to switch state $Q(t)$. Let $W^B(t)$ denotes the weight of a scheduling algorithm B at time t. Further, B has property that, $W^B(t) \geq W^*(t) - f(W^*(t))$, for all t, where $f(.)$ is a sub-linear function. Then, the scheduling algorithm B is stable under any admissible Bernoulli i.i.d. input traffic.

We consider a new class of matching scheme as follows. Define $\Psi(t) = [\psi_{ij}(t)]_{N\times N}$, and

$$\Psi(t) = Q(t) + C(t), \tag{2}$$

where $C(t) = [c_{ij}(t)]_{N\times N}$, and $0 \leq c_{ij}(t) \leq K$, $0 \leq i, j \leq N$, $K < \infty$. In each time slot, a maximum weight match, which is calculated with respect to $\Psi(t)$ (not to $Q(t)$ as in MWM), is used to schedule the packet transmission.

Theorem 2 A matching algorithm as described above is stable under any admissible Bernoulli i.i.d. input traffic.

Proof: According to equation (2), we have

$$\psi_{ij}(t) = q_{ij}(t) + c_{ij}(t). \tag{3}$$

At time t, denote the match using the new scheme as $\widehat{M}(t)$, and the match using MWM as $M^*(t)$. We therefore denote the weight of $\widehat{M}(t)$ as

$$\widehat{W}(t) = \langle \widehat{M}(t), Q(t)\rangle, \tag{4}$$

and the weight of $M^*(t)$ as

$$W^*(t) = \langle M^*(t), Q(t)\rangle, \tag{5}$$

Then,

$$\begin{aligned}
&\widehat{W}(t) = \langle \widehat{M}(t), Q(t)\rangle \\
= \;& \langle \widehat{M}(t), \Psi(t) - C(t)\rangle \\
= \;& \langle \widehat{M}(t), \Psi(t)\rangle - \langle \widehat{M}(t), C(t)\rangle \\
\geq \;& \langle M^*(t), \Psi(t)\rangle - \langle \widehat{M}(t), C(t)\rangle \\
= \;& \langle M^*(t), Q(t) + C(t)\rangle - \langle \widehat{M}(t), C(t)\rangle \\
= \;& \langle M^*(t), Q(t)\rangle + \langle M^*(t), C(t)\rangle - \langle \widehat{M}(t), C(t)\rangle \\
= \;& W^*(t) + \langle M^*(t), C(t)\rangle - \langle \widehat{M}(t), C(t)\rangle.
\end{aligned} \tag{6}$$

Since $0 \leq c_{ij}(t) \leq K$,

$$-NK \leq \langle M^*(t), C(t)\rangle - \langle \widehat{M}(t), C(t)\rangle \leq NK. \tag{7}$$

Combining equations (6) and (7), we have

$$\widehat{W}(t) \geq W^*(t) - f(W^*(t)), \tag{8}$$

where $f(W^*(t)) = \langle \widehat{M}(t), C(t)\rangle - \langle M^*(t), C(t)\rangle$ is a sub-linear function. Therefore, according to Theorem 1, the new matching algorithm is stable under any admissible Bernoulli i.i.d. arrival traffic. ■

In fixed-length switches, when a cell is transferred to its destination output, it will stay in a buffer and wait for the other cells of the same packet. After the complete reception of all the cells of the same packet, these cells will be reassembled into a packet. The delay a packet suffers before it is reassembled into a packet and delivered to its destination includes the cell delay, and the waiting time at the output reassembly buffer, which is often ignored in many papers. Among all existing matching algorithms, MWM achieve the lowest average cell delay. Other schemes, which do not achieve maximum weight match in each time slot, may not give as low an average cell delay as MWM. In the new class of algorithms discussed above, if $C(t)$ is carefully selected, it is possible to greatly reduce the reassembly delay at the cost of slightly higher cell delay, and therefore achieve lower average packet delay than MWM. We therefore refer to these algorithms as being members of the Packet Delay-Aware MWM (PDA-MWM).

III. Delay Performance of Packet Delay-Aware MWM algorithms

In this section, we investigate three PDA-MWM algorithms and compare their delay performance to MWM. For a more realistic evaluation of switch performance, we consider the following average delays in our simulation.

- Cell delay: the delay a cell suffers from the time it enters the system to the time it is transferred from the input to its destined output.
- Reassembly delay: the delay a cell suffers from the time it is transferred to it destined output to the time it is reassembled and departs the system.
- Packet delay: as in [13] and [14], the packet delay of a packet is measured from the time when the last cell of a packet enters the system to the time it departs.

All simulation results shown in this paper are for a 32x32 switch under uniform traffic. Two different packet patterns are considered in is paper, as follows.

- Pattern 1, the packet length is fixed with a size of 10 cells.
- Pattern 2 is based on the Internet traffic measurements from [16], where 60% of the packets are 44 bytes, 20% are 552 bytes, and the rest are 1500 bytes. In our simulation, we assume a cell payload of 44 bytes and define the packet size distribution as follows: the size of 60% of the packets is 1 cell, the size of 20% of the packets is 13 cells, and the size of other packets is 34 cells. The average packet size is 10 cells.

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

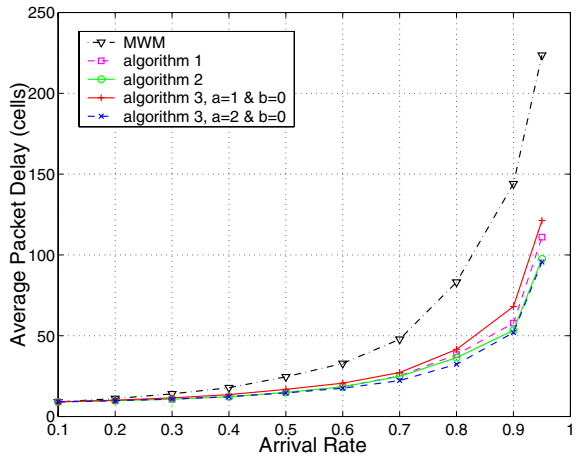

Fig. 1. The average packet delay for packet pattern 1.

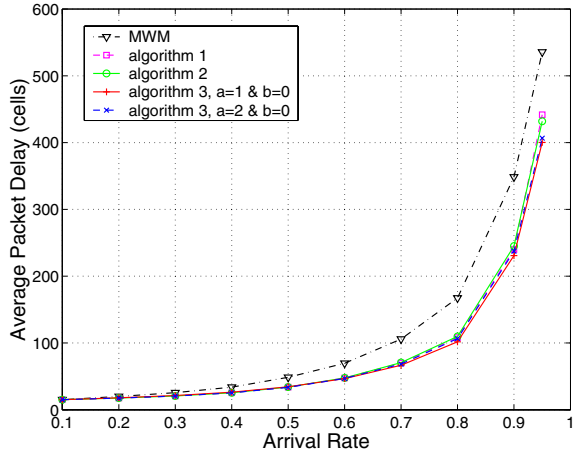

Fig. 2. The average packet delay for packet pattern 2.

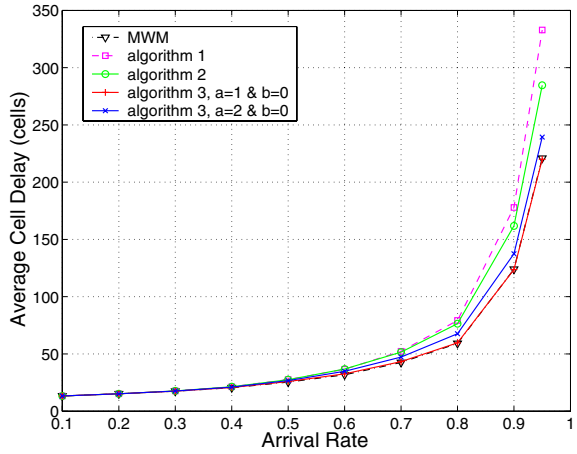

Fig. 3. The average cell delay for packet pattern 2.

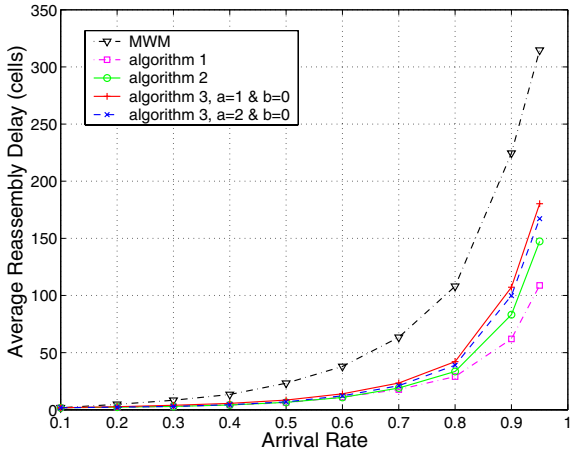

Fig. 4. The average reassembly delay for packet pattern 2.

A. PDA-MWM algorithm 1

If a VOQ was served in time slot $t-1$, and is not empty at time t, we say that this VOQ satisfies condition 1. In PDA-MWM algorithm 1, we define $C(t)$ as follows:

$$c_{ij}(t) = \begin{cases} \Gamma, & \text{if } VOQ_{ij} \text{ satisfies condition 1;} \\ 0, & \text{Otherwise.} \end{cases} \quad (9)$$

In the above equation, Γ is a constant integer number. When Γ is small, algorithm 1 is close to MWM. When Γ is large, algorithm 1 is close to an exhaustive service MWM algorithm, in which all cells in a VOQ is served continuously until the VOQ is empty if the VOQ is matched, and MWM is used in each time slot for a subset of VOQs that are not currently matched.

In this algorithm, when $\Gamma > 0$, if a VOQ is matched at time $t-1$, it has a better chance to be matched at time t. Therefore, packets in the same VOQ are likely to be served continuously, which usually leads to lower reassembly delay.

Figures 1 and 2 shows the average packet delay under packet patterns 1 and 2, respectively. For both packet patterns, algorithm 1 (with $\Gamma = 100$) gives a lower average packet delay than MWM. According to our simulation, by setting $\Gamma = 100$, if a VOQ is matched at a particular time, usually the match for that VOQ is kept unchanged until all cells in the VOQ are served. Packet pattern 2 is close to the real traffic in Internet. In figures 3 and 4, the average cell delay and average reassembly delay of different matching schemes under packet pattern 2 are compared, respectively. They show that algorithm 1, compared to MWM, has a higher cell delay. However, its reassembly delay is the lowest among MWM and all the other schemes discussed in this paper.

B. PDA-MWM algorithm 2

If a VOQ was served in time slot $t-1$, and its Head of Line (HOL) packet is in the middle of servic at time t, we say that this VOQ satisfies condition 2. In PDA-MWM algorithm 2, we define $C(t)$ as follows:

$$c_{ij}(t) = \begin{cases} \Gamma, & \text{if } VOQ_{ij} \text{ satisfies condition 2;} \\ 0, & \text{Otherwise.} \end{cases} \quad (10)$$

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

In the above equation, Γ is a constant integer number. When Γ is small, algorithm 2 is close to MWM. When Γ is large enough, all cells in a packet are likely to be served continuously.

In algorithm 2, when $\Gamma > 0$, if a VOQ is matched at time $t-1$ and the HOL packet is partially served, it has a better chance to be matched at time t. Therefore, cells in the same packet are likely to be served continuously, so that the reassembly delay can be lowered.

As shown in figures 1 and 2 for the two packet patterns, the average packet delay of algorithm 2 (with $\Gamma = 100$) is much lower than MWM, and is slightly lower than algorithm 1. According to our simulation, by setting $\Gamma = 100$, if a VOQ is matched at one time, usually the match is kept until all cells in the HOL packet are served.

In figures 3 and 4, with packet pattern 2, it is interesting to see that algorithm 2 has a lower cell delay but a higher reassembly delay than algorithm 1. The reason that MWM always leads to the lowest cell delay is probably because MWM always achieves the maximum weight match (in the sense of queue length) in each time slot. It is likely that in a matching scheme, if the weight of the match (in the sense of queue length) in each time slot is close to the weight of match achieved by MWM, the average cell delay in this scheme will be lower than other schemes with smaller match weights. In algorithm 1 and algorithm 2, it is likely that VOQs satisfying condition 1 and condition 2, respectively, will keep their matches from the previous time slots, and other VOQs will be matched using MWM. Compared to algorithm 1, algorithm 2 achieves 'better' schedules because VOQs release their matches more frequently so that more VOQs are involved in being matched to each other by MWM. Therefore, the average weight of matches in algorithm 2 is larger than that of algorithm 1, which leads to a lower cell delay. On the other hand, there is always a tradeoff between the cell delay and the reassembly delay. In order to achieve low packet delay, both the cell delay and the reassembly delay performance have to be carefully considered.

C. PDA-MWM algorithm 3

If the HOL packet of a VOQ is partially served at time t, we say that this VOQ satisfies condition 3. Condition 3 is different from conditions 1 and 2 in that at one time, at most N VOQs may satisfy condition 1 and at most N VOQs may satisfy condition 2, while there could be more than N VOQs satisfying condition 3.

In PDA-MWM algorithm 3, we define $C(t)$ as follows:

$$c_{ij}(t) = \begin{cases} f(k_{ij}(t)), & \text{if } VOQ_{ij} \text{ satisfies condition 3;} \\ 0, & \text{Otherwise.} \end{cases} \tag{11}$$

$k_{ij}(t)$ is the number of remaining cells of the HOL packet of VOQ_{ij}. $f(k_{ij}(t))$ is a function of $k_{ij}(t)$ and is bounded by a finite number Λ, so that

$$0 \leq f(k_{ij}(t)) \leq \Lambda. \tag{12}$$

$f(k)$ can be defined in many ways. In this paper, we define it as follows:

$$f(k) = a(K_{max} - k) + b. \tag{13}$$

where K_{max} is the maximum length of a packet in cells, and a and b are two constants.

When $a = 0$ and $b = 0$, algorithm 3 is actually MWM.

When $a = 0$ and $b \neq 0$,

$$c_{ij}(t) = \begin{cases} b, & \text{if } VOQ_{ij} \text{ satisfies condition 3;} \\ 0, & \text{Otherwise.} \end{cases} \tag{14}$$

The larger b is, the higher priority will be given to unfinished packets.

When $a \neq 0$,

$$c_{ij}(t) = \begin{cases} a(K_{max} - k_{ij}(t)) + b, & \text{if } VOQ_{ij} \text{ satisfies condition 3;} \\ 0, & \text{Otherwise.} \end{cases} \tag{15}$$

Thus, a HOL packet which has fewer cells left is more likely to be served than those packets which have more cells left, as well as those HOL packets which have not been served. In this way, when most of the cells in a packet have been served, instead of letting them wait in the reassembly buffer for a long time, the rest of the packet (which only has a few cells) will be served before the idly. Therefore, the entire packet can depart the system earlier and the average reassembly delay can be reduced. The larger a is, the higher priority will be given to short unfinished packets.

In order to achieve low packet delay, it is very important to choose a and b carefully to keep both cell delay and reassembly delay relatively low. In figures 5, 6 and 7, we show the packet delay, cell delay and reassembly delay of algorithm 3, respectively, for different choices of a and b under arrival rate 0.95. Again, these figures clearly show the tradeoff between the cell delay performance and the reassembly delay performance. Among all pair of a and b, we look at the following cases as examples:

- case 1, when $a = 0$ and $b = 0$ (MWM)
- case 2, when $a = 1$ and $b = 0$
- case 3, when $a = 0$ and $b = 10$
- case 4, when $a = 2$ and $b = 0$
- case 5, when $a = 4$ and $b = 100$

Simulation result shows that case 1 gives the lowest cell delay, while in cases 2 and 3 the cell delay is very close to case 1. This is because that in cases 2 and 3, the VOQ weight used to calculate the match is not changed a lot from MWM, so that the matches are close to maximum weight matches. On the other hand, case 5 favors short unfinished packets the most and therefore achieves the lowest reassembly delay. Case 4 behaves somewhere between cases 2 and 3 and case 5. By taking account of both cell delay performance and reassembly delay performance, case 2 gives the lowest average packet delay.

According to the result on similar simulation for packet pattern 1, we find that the average packet delay is the lowest in case 4.

For cases 2 and 4, the packet delay performance for packet pattern 1, and the packet delay, cell delay and reassembly delay performance for packet pattern 2 are shown in figures 1, 2, 3 and 4, respectively. Among all schemes discussed in this paper, algorithm case 4 gives the lowest packet delay for

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

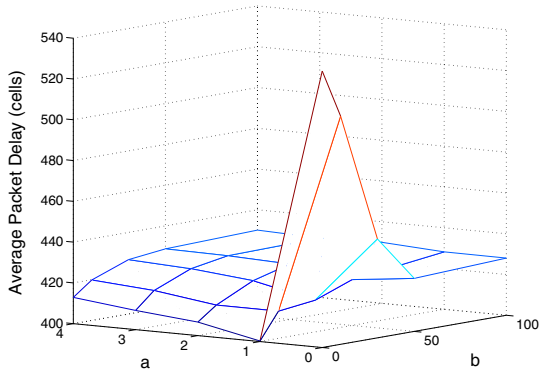

Fig. 5. The average packet delay for packet pattern 2 in algorithm 3 under arrival rate 0.95.

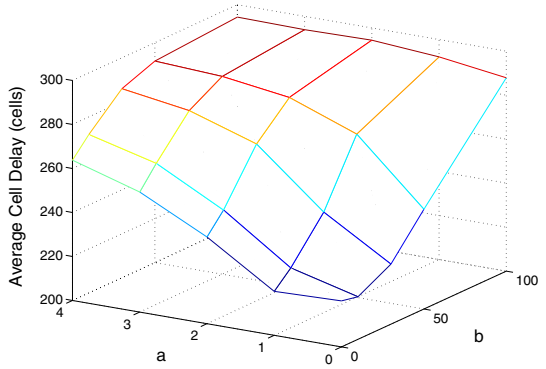

Fig. 6. The average cell delay for packet pattern 2 in algorithm 3 under arrival rate 0.95.

packet pattern 1, and case 2 gives the lowest packet delay for packet pattern 2.

IV. Conclusions

In fixed-length VOQ switches, variable-length IP packets are segmented into fixed-length cells at the inputs, and the cells are placed in the corresponding VOQ. When a cell is transferred to its destination output, it will stay in the reassembly buffer and wait for the other cells of the same packet before the entire packet can depart the system. Therefore, the delay a packet suffers in the system includes the waiting time in the VOQ (the cell delay), and the waiting time at the output reassembly buffer (the reassembly delay). The cell delay performance has been well studied in previous work, while the reassembly delay is often ignored in many paper. Among all existing matching algorithms, MWM gives the best cell delay performance. However, the question of the best achievable, i.e., optimal, packet delay performance for a matching algorithm is still open.

In this paper, we investigate the average packet delay of a input buffered packet switch, which has not been well studied in previous work. A new class of matching algorithms, PDA-MWM, is defined and proved to be stable under all admissible Bernoulli i.i.d. arrival traffic. The packet delay, cell delay and reassembly delay performance are studied for three PDA-MWM matching algorithms. An important insight given by this work is that, in order to achieve low packet

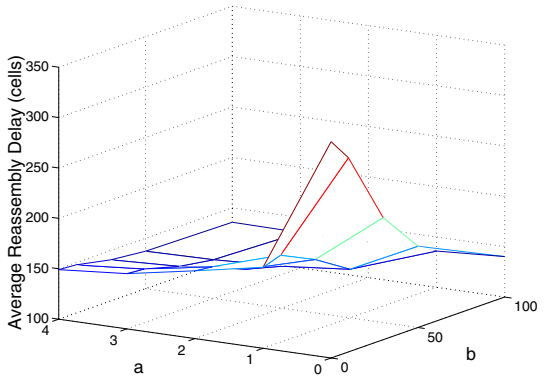

Fig. 7. The average reassembly delay for packet pattern 2 in algorithm 3 under arrival rate 0.95.

delay, there is a tradeoff between the cell delay and the reassembly delay. We show that, compared to MWM, the average packet delay can be significantly reduced if both of the cell delay performance and the reassembly delay performance are carefully considered.

References

[1] M. J. Karol, M. Hluchyj, and S. Morgan, "Input versus output queuing on a space-division packet switch," *IEEE Trans. on Communications*, vol.35, pp. 1347-1356, 1987.

[2] L. Tassiulas, A. Ephremides, "Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks," *IEEE Trans. Automatic Control*, Vol. 37, No. 2, pp. 1936-1949.

[3] N. McKeown, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE INFOCOM'96*, pp. 296-302.

[4] A. Charny, P. Krishna, N. Patel and R. Simcoe, "Algorithms for providing bandwidth and delay guarantees in Input-Buffered crossbars with speedup", *IWQOS'98*, May 1998.

[5] P. Krishna, N. S. Patel, A.Charny and R. Simcoe, "On the speedup required for work-conserving crossbar switches", *IWQOS'98*, May 1998.

[6] A. Mekkittikul and N. McKeown, "A practical scheduling algorithm to achieve 100% throughput in input-queued switches", *IEEE INFOCOM 98*, Vol 2, pp. 792-799, April 1998.

[7] T. E. Anderson, S. S. Owicki, J. B. Saxe and C. P. Thacker, "High speed switch scheduling for local area networks," *ACM Trans. on Computer Systems*, vol. 11, No. 4, pp. 319-352, Nov. 1993.

[8] N. McKeown, "The iSLIP scheduling algorithm for Input-Queued switches", *IEEE/ACM Trans. Networking*, vol. 7, pp. 188-201, April 1999.

[9] H. J. Chao, "Saturn: a terabit packet switch using Dual Round-Robin", *IEEE Communication Magazine*, vol. 38 12, pp. 78-84, Dec. 2000.

[10] Y. Li, S. Panwar, H. J. Chao, "On the performance of a Dual Round-Robin switch," *IEEE INFOCOM 2001*, vol. 3, pp. 1688-1697, April 2001.

[11] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches," *IEEE INFOCOM 1998*, vol.2, New York, 1998, pp.533-539.

[12] P. Giaccone, B. Prabhakar, D. Shah "Toward simple, high-performance schedulers for high-aggregate bandwidth switches", *IEEE INFOCOM 2002*, New York, 2002.

[13] Y. Li "Design and analysis of schedulers for high speed input queued switches," *Ph.D. Dissertation*, Polytechnic University, Jan. 2004.

[14] Y. Li, S. Panwar, H. J. Chao, "Exhaustive service matching algorithms for input queued switches," *2004 Workshop on High Performance Switching and Routing (HPSR 2004)*, April 2004.

[15] D. Shah, M. Kopikare, "Delay bounds for approximate maximum weight matching algorithms for input queued switches," *IEEE INFOCOM 2002*, New York, 2002, pp. 1024-1031.

[16] K Claffy, G. Miller, K. Thompson, "The nature of the beast: recent traffic measurements from an Internet backbone," Caida.

1-4244-0357-X/06/$20.00 ©2006 IEEE