Exhaustive Service Matching Algorithms for Input Queued Switches

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

Abstract— **Virtual Output Queuing is widely used by fixed-length high-speed switches to overcome head-of-line blocking. This is done by means of matching algorithms. *Maximum matching algorithms* have good performance, but their implementation complexity is quite high. *Maximal matching algorithms* need speedup to guarantee good performance. *Iterative matching schemes*, such as iSLIP and DRRM, use multiple iterations to converge on a maximal match. The objective of matching algorithms is to reduce the matching overhead for each time slot. In this paper, *Exhaustive Service Matching* is presented as a way to amortize the cost of a match over multiple time slots, thus significantly improving switch performance. In an Exhaustive Service Matching switch, cells belonging to the same packet are transferred to the output continuously, which leads to good packet delay performance and simplifies the implementation of packet reassembly. To avoid unfairness under some extremely unbalanced traffic pattern, Limited Service Matching and Exhaustive Service Matching with Hamiltonian Walk (EMHW) are presented. We show that Limited Service Matching achieves better fairness under unbalanced traffic patterns, and in some cases improves the delay performance, while retaining low implementation complexity and a scalable architecture. We prove that EMHW is stable under all admissible traffic. All these schemes can be applied to existing matching algorithms, such as iSLIP and DRRM, to achieve high switching efficiency with low implementation complexities.**

Index Terms— **switching, scheduling, Virtual Output Queueing, polling, exhaustive service, limited service, Hamiltonian walk.**

I. Introduction

FIXED-LENGTH switching technology is widely accepted as an approach to achieving high switching efficiency for high speed packet switches. Variable-length IP packets are segmented into fixed-length "cells" at the inputs and are reassembled at the outputs. 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. However, an Input Queuing switch has a critical drawback [1], [2]: 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 in the fabric limits the scalability of this architecture. Virtual Output Queuing (VOQ) is used to overcome these drawbacks and combine the advantages of an Input Queuing switch and an Output Queuing switch. In a VOQ switch, each input maintains N queues, one for each output. By using VOQ, no additional speedup is required and HOL blocking can be eliminated.

Considerable work has been done on scheduling algorithms for VOQ switches. 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) [3], [4]. But maximum weight matching is not practical to for hardware implementation. A number of practical *maximal matching algorithms* have been proposed [5], [6], [7], but maximal matching algorithm cannot achieve as high a throughput as maximum matching algorithms. Iterative algorithms such as PIM [8], iSLIP [9], and Dual Round Robin Matching (DRRM) [10], [11], [12] use multiple iterations to converge on a maximal matching. Recently, a class of matching algorithms, which are not MWM and can guarantee 100% throughput without speedup have been devised. One approach is to use a randomized scheduling algorithm, presented in [13], which has low complexity but very high delay. In [14], ALGO3 achieves stability by using a Hamiltonian walk, and APSARA, LAURA and SERENA further improve the delay performance at the cost of higher complexity. Compared to APSARA and LAURA, SERENA has similar performance but with a lower time complexity of $O(N)$. These matching algorithms have typically been cell-based. That is, in every time slot, a new matching set is generated and the switch fabric is updated to connect the new set of matched inputs and outputs.

In this paper, a class of matching algorithms, Exhaustive Service Matching and its variations, are presented to achieve good performance and stability with low implementation complexity. Unlike many other matching algorithms, which try to find the best match possible in each time slot, Exhaustive Service Matching achieves efficiency by minimizing the matching overhead over time. Additionally, note that cells forwarded to outputs are held in reassembly buffers can only leave the switch when all cells belonging to the same packet are received so that the packet is reassembled. Thus the total delay a packet suffers, from the time it arrives at the input to the time it departs at the output, includes the cell delay incurred traversing the switch and the time needed for packet reassembly. As we shall see, in Exhaustive Service Matching, since all the cells belonging to the same packet are transferred to the output continuously, the packet delay is significantly reduced.

Under an extremely unbalanced arrival traffic, an exhaustive service policy may lead to unfairness and starvation. Two variations of Exhaustive Service Matching, Limited Service

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

Matching and Exhaustive Service Matching with Hamiltonian walk (EMHW), are therefore presented. We show that, without additional implementation complexity, Limited Service Matching achieves better fairness under an unbalanced traffic pattern, and in some cases improves the delay performance. We prove that EMHW is stable under all admissible traffic, regardless of what matching algorithm is used, and the service under EMHW is always as good as or better than under ALGO3 in [14].

Exhaustive or limited schemes can be used in conjunction with existing matching algorithms, such as iSLIP and DRRM [15], [12]. Only one iteration is enough and no speedup is needed to achieve high switching efficiency. In this paper Exhaustive service iSLIP (E-iSLIP), Limited service iSLIP (L-iSLIP) and E-iSLIP with Hamiltonian walk (HE-iSLIP) are presented. Their implementation complexity is $O(logN)$. Simulation results show that these schemes have better performance than iSLIP. HE-iSLIP is compared to ALGO3 and SERENA, which are stable as well. The implementation complexity of ALGO3 is $O(logN)$, and that of SERENA is $O(N)$. HE-iSLIP exhibits much lower packet delay than ALGO3. Under uniform traffic and for some nonuniform traffic patterns, the packet delay performance of HE-iSLIP is better than that of SERENA, while under *diagonal traffic*, the delay of SERENA is lower.

Exhaustive Service Algorithms, Limited Service Algorithms and EMHW are described in section II, III and IV. Simulated performances are presented in section V.

II. Exhaustive Service Matching Algorithms

In the switches under consideration in this paper, packets with variable length are segmented into fixed-size cells when they arrive and are put into VOQs according to their destination output. In most of the previous work, when an input and an output are matched, only one cell is transferred from the input to the matched output. This behavior is similar to the *limited service policy* with a limit of 1 [20] in a polling system.

In order to improve the performance under nonuniform traffic and bursty traffic, we modify the limit-1 service policy so that whenever an input is matched to an output, all cells in the corresponding VOQ will be transferred in the following time slots before any other VOQ at the same input can be served. This is called the *exhaustive service policy* [20] in polling systems. In exhaustive service matching algorithms, the matching overhead, consisting of unused time slots, is amortized over all packets served continuously under the same match.

In an Exhaustive Service Matching Algorithm, we say that in a given time slot, each input and output is either *free* or *busy*. An input (output) is busy if it is matched to an output (input), otherwise it is free. At the beginning of a time slot, the matches for busy inputs and outputs are the same as those in the previous time slot, and free inputs and outputs will be matched by the matching algorithm.

We apply the exhaustive service policy to iSLIP, and call the new schemes Exhaustive service iSLIP (E-iSLIP).

In iSLIP, there are three steps in each iteration. First, each input sends requests for all of its nonempty VOQs. Then, each output selects one request to grant in round robin order. Finally, each input accepts one of the multiple grants, also in round robin order.

For E-iSLIP, at the beginning of each time slot, a busy input (output) with a VOQ that has just emptied, (1)increments its pointer to one location beyond the matched output (input), and (2)changes its state to free. If the corresponding VOQ is not empty, the arbiter pointer of a busy input (output) always points to the matched output (input). A detailed description of the three step E-iSLIP algorithm follows:

Step 1: Request. Each free input sends a request to every output for which it has a queued cell. Each busy input sends a request to the matched output.

Step 2: Grant. If an output (either free or busy) receives any requests, it chooses the one that appears next in a fixed, round-robin schedule starting from the highest priority element. The output notifies each input whether or not its request was granted.

Step 3: Accept. If an input receives a grant, it sets its state to busy, accepts the grant that appears next in a fixed, round-robin schedule starting from the highest priority element. The input pointer then points to the matched output. If an output receives an accept, it sets its state to busy, and its pointer points to the matched input.

In E-iSLIP, free outputs only get requests from free inputs, and free inputs only get grants from free outputs. Thus the process of looking for a new match is similar to that of iSLIP for a smaller switch size.

The complexity of the arbitration in E-iSLIP is the same as that in iSLIP, which is $O(logN)$ [10].

It is possible that under some extremely unbalanced arrival traffic, an exhaustive service policy may lead to unfairness and starvation, where one input may occupy an output for a long time before any other input can be served by the same output. Two variations can be used to make the matching scheme stable. One uses limited service instead of exhaustive service, while for the other we introduce a Hamiltonian walk.

III. Limited Service Matching Algorithms

In limited service matching algorithms, when an input is matched to an output, a limit on the maximum number of cells that can be served continuously in the corresponding VOQ is enforced by means of a counter.

In Limited service iSLIP (L-iSLIP), each input and output maintains a counter to record the number of cells which have been served under the current match. In a given time slot, a busy input (output) (1)increments its pointer to one location beyond the matched output (input), (2)changes its state to free, and (3)sets its counter to 0, if (1) the corresponding VOQ is empty, or (2) the counter reaches LIMIT; however, if LIMIT falls in the middle of a group of cells belonging to a packet, all cells from that packet are served before the service under the current match terminates. After a cell is transferred, the counters of the corresponding input and output increment by one. The three matching steps in L-iSLIP are the same as those of E-iSLIP.

The implementation complexity of L-iSLIP is the same as that of E-iSLIP, except for the counter at each port. The last cell of a packet can be recognized by outputs for reassembly purpose, by means, for example, of an end of message bit in the header. In a Limited Service Matching algorithm, the same mechanism should be available for inputs so that all cells in a packet will be served continually without interruption. Limited Service matching algorithm does not guarantee stability under any traffic pattern. However, simulation results show that the throughput of L-iSLIP is always close to 100%. For example, the throughput of a 32x32 L-iSLIP switch is always higher than 95% for a range of LIMIT value. Therefore, 100% throughput is achieved with a speedup a little bit higher than 1. This does not lead to significant additional complexity, since a fixed-length cell switch always needs a significant speedup to compensate for segmentation overhead, e.g., the header of cells, and pad bytes needed to fill the last cell of a packet.

IV. Exhaustive Service Matching with Hamiltonian Walk

In [14], algorithms which introduced a Hamiltonian walk, were proved to be stable under all admissible Bernoulli i.i.d. inputs. In this section, we will prove that any Exhaustive Service Matching with Hamiltonian Walk (EMHW), no matter what matching algorithm is used, is stable under any admissible Bernoulli i.i.d. arrival traffic, and is always as good as or better than ALGO3, an algorithm from [14] with comparable complexity.

A Hamiltonian walk is a walk which visits every vertex of a graph exactly once. For a $N \times N$ switch, the total number of possible schedules is $N!$. If those schedules are mapped on to a graph with $N!$ vertices so that each vertex corresponds to a schedule, a Hamiltonian walk on the graph visits each vertex exactly once every $N!$ time slots. The vertex which is visited at time t is denoted by $H(t)$. The complexity of generating $H(t+1)$ is $O(1)$, when $H(t)$ is known [17].

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

ALGO3 is defined as follows:

(a) Let $S_3(t)$ be the schedule used at time t.

(b) At time $t+1$, let

$$S_3(t+1) = \arg \max_{S \in \{S_3(t), H(t+1)\}} \langle S, Q(t+1)\rangle. \tag{1}$$

EMHW is defined as follows:

(a) Let $S(t)$ be the schedule used at time t.

(b) At time $t+1$, get match $Z(t+1)$ by the Exhaustive Service Matching algorithm, based on the previous schedule $S(t)$, and $H(t+1)$ from $H(t)$ by a Hamiltonian walk.

(c) Let

$$S(t+1) = \arg \max_{S \in \{Z(t+1), H(t+1)\}} \langle S, Q(t+1)\rangle. \tag{2}$$

For E-iSLIP with Hamiltonian walk (HE-iSLIP), by the end of time slot t, inputs and outputs which are matched in $S(t)$ (1) set their states to busy, and (2) update their pointers to the outputs and inputs with which they are matched. Unmatched inputs and outputs set their states to free and do not update their pointers. The state updating at the beginning of a time slot and the steps to select $Z(t+1)$ are the same as those in E-iSLIP.

We will prove that EMHW, no matter what kind of matching scheme is used, is stable under any admissible traffic, and is always as good as or better than ALGO3.

Lemma 1: If $S(t)$ is the schedule at time t, and $Z(t+1)$ is the schedule at time $t+1$ chosen by an Exhaustive Service Matching Algorithm, $\langle S(t), Q(t+1)\rangle \leq \langle Z(t+1), Q(t+1)\rangle$.

Proof: According to the definition of Exhaustive Service Matching, if at time t a VOQ is matched, at $t+1$ it must be matched if it is nonempty. In other words, a VOQ which is matched at time t but not matched at time $t+1$ must be empty at time $t+1$. Therefore, both $S(t)$ and $Z(t+1)$ can be expressed by two parts,

$$S(t) = S_0(t) + S_1(t), \tag{3}$$

and

$$Z(t+1) = Z_1(t+1) + Z_2(t+1). \tag{4}$$

$S_0(t)$ denotes those input-output pairs which are matched at time t and not matched at $t+1$. That means the corresponding VOQs become empty at time $t+1$. Therefore,

$$\langle S_0(t), Q(t+1)\rangle = 0. \tag{5}$$

$S_1(t) = Z_1(t+1)$, denote those input-output pairs which are matched both at time t and $t+1$. $Z_2(t+1)$ denotes those input-output pairs which are not matched at time t but are matched at time $t+1$. Obviously,

$$\langle Z_2(t+1), Q(t+1)\rangle \geq 0. \tag{6}$$

Therefore,

$$\begin{aligned} & \langle S(t), Q(t+1)\rangle = \langle S_0(t), Q(t+1)\rangle + \langle S_1(t), Q(t+1)\rangle \\ \leq \; & \langle Z_1(t+1), Q(t+1)\rangle + \langle Z_2(t+1), Q(t+1)\rangle \\ = \; & \langle Z(t+1), Q(t+1)\rangle. \end{aligned} \tag{7}$$

■

Lemma 2: If $S(t)$ is the schedule at time t, and $S(t+1)$ is the schedule generated at time $t+1$ by an EMHW algorithm, it is always true that $\langle S(t), Q(t+1)\rangle \leq \langle S(t+1), Q(t+1)\rangle$.

Proof: If $Z(t+1)$ is the match at time $t+1$ determined by the Exhaustive Service Matching based on $S(t)$, by the definition of EMHW, we always have

$$\langle Z(t+1), Q(t+1)\rangle \leq \langle S(t+1), Q(t+1)\rangle. \tag{8}$$

Combining (8) and Lemma 1, we get

$$\langle S(t), Q(t+1)\rangle \leq \langle S(t+1), Q(t+1)\rangle. \tag{9}$$

■

Remark: EMHW is the first matching algorithm using a Hamiltonian walk that does not use the previous match $S(t)$ as a candidate match for $S(t+1)$ (see equation (11)). Lemma 1 shows that including $S(t)$ as a candidate will not increase the weight of the match.

Theorem 1: ([16]) 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.

Theorem 2: An EMHW is stable under any admissible Bernoulli i.i.d. input traffic.

Proof: In a time slot, there can be at most one arrival and at most one departure for each input. This implies that for any match M, $k \geq 0$,

$$\begin{aligned} \langle M, Q(t)\rangle - kN &\leq \langle M, Q(t+k)\rangle \\ &\leq \langle M, Q(t)\rangle + kN. \end{aligned} \tag{10}$$

As defined before, at time t, $S(t)$ is the EMHW schedule with weight $W(t)$. Let M^* and M_0^* be the MWM at time t and $t - N!$, respectively. Because, by the definition of a Hamiltonian walk, each match will be visited exactly once within $N!$ time slots, there must be a time $T \in [t - N!, t]$ such that $H(T) = M_0^*$. Then

$$\begin{aligned} &\langle S(T), Q(T)\rangle \geq \langle H(T), Q(T)\rangle = \langle M_0^*, Q(T)\rangle \\ \geq\ &\langle M_0^*, Q(t-N!)\rangle - (T - t + N!)N. \end{aligned} \tag{11}$$

From (10) and Lemma 2, for any time t we have

$$\begin{aligned} \langle S(t), Q(t)\rangle - N &\leq \langle S(t), Q(t+1)\rangle \\ &\leq \langle S(t+1), Q(t+1)\rangle. \end{aligned} \tag{12}$$

Therefore, by (11) and (12),

$$\begin{aligned} \langle S(t), Q(t)\rangle &\geq \langle S(T), Q(T)\rangle - (t-T)N \\ &\geq \langle M_0^*, Q(t-N!)\rangle - NN! \\ &\geq \langle M^*, Q(t-N!)\rangle - NN! \\ &\geq \langle M^*, Q(t)\rangle - 2NN!. \end{aligned} \tag{13}$$

Therefore, according to Theorem 1, EMHW is stable under any admissible Bernoulli i.i.d. arrival traffic. ■

Theorem 3: Suppose the schedule at time t is $M(t)$, and at time $t+1$ the schedules generated by ALGO3 and EMHW are $S_3(t+1)$ and $S(t+1)$, respectively. Then it is always true that

$$\langle S(t+1), Q(t+1)\rangle \geq \langle S_3(t+1), Q(t+1)\rangle. \tag{14}$$

Proof: From Lemma 2, we have

$$\langle M(t), Q(t+1)\rangle \leq \langle S(t+1), Q(t+1)\rangle. \tag{15}$$

According to the definition of EMHW,

$$\langle H(t+1), Q(t+1)\rangle \leq \langle S(t+1), Q(t+1)\rangle. \tag{16}$$

From the definition of ALGO3, we know that

$$S_3(t+1) = \arg \max_{S \in \{M(t), H(t+1)\}} \langle S, Q(t+1)\rangle. \tag{17}$$

Therefore,

$$\langle S(t+1), Q(t+1)\rangle \geq \langle S_3(t+1), Q(t+1)\rangle. \tag{18}$$

■

In EMHW, a centralized controller is needed to compute the weights of the matches generated by exhaustive service matching and the Hamiltonian walk, respectively. Therefore, each input has to send the lengths of up to two VOQs to the controller. The controller will add up the VOQ lengths for each match, with time complexity $O(logN)$. The controller then compares the two weights and selects the match with larger weight. The time complexity of generating the next vertex in a Hamiltonian walk is $O(1)$. The time complexity of E-iSLIP is $O(logN)$. Therefore, the complexity of HE-iSLIP is $O(logN)$, the same as ALGO3.

V. The Simulated Performance

In this section, the simulated performance of average packet delay is presented under uniform and nonuniform traffic. The performance of E-iSLIP, L-iSLIP and HE-iSLIP are compared to iSLIP, ALGO3 and SERENA under uniform and nonuniform traffic. SERENA is a matching algorithm with complexity of $O(N)$ and is stable under any admissible arrival traffic [14]. We will see that ALGO3, with comparable implementation complexity, has poor performance compared to HE-iSLIP, and SERENA, with a higher implementation complexity, has higher delay than HE-iSLIP under uniform traffic, while the results under nonuniform traffic are mixed.

In fixed-length 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 a buffer and wait for the other cells in the same packet. After the complete reception of all the cells coming from 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. In order to evaluate the switch performance properly, we consider average packet delay performance in this paper. After a packet is segmented into cells, one cell will be put into the VOQ in each time slot. As in [18], the packet delay of a packet is measured from the time when the last cell of the packet enters the VOQ until the time when the same last cell of the packet is transferred to its destined output line.

A. *Under uniform traffic*

Three different packet patterns are considered in the simulation. For pattern 1, the packet length is fixed with a size of 1 cell. For pattern 2, the packet length is fixed with a size of 10 cells. Pattern 3 is based on the Internet traffic measurements from [19], where 60% of the packets are 44 bytes, 20% are 552 bytes, and the rest are 1500 bytes. In our simulation, we 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. This assumes a cell payload of 44 bytes. The average packet size is 10 cells. We compared the packet delay of E-iSLIP, L-iSLIP and HE-iSLIP to iSLIP, ALGO3 and SERENA for a 32x32 switch under uniform traffic for different packet patterns. Simulation results are shown in Figures 1, 2 and 3. Since the delay performance of E-iSLIP, HE-iSLIP and L-iSLIP (with a LIMIT of 100 cells or larger) are almost identical, the plots for all three overlap each other in the figures.

In a VOQ switch, an input wastes a certain number of slot times searching for a match and then is served during the service time. As the ratio of the service time to the searching time increases, the efficiency becomes higher. In L-iSLIP, when LIMIT increases, the service time is increased, which, under unifrom traffic, leads to a higher efficiency and better performance. On the other hand, we find that under uniform traffic, the performance of L-iSLIP does not improve much when LIMIT is increased beyond a point, e.g., a LIMIT equal to 100 cells and infinity (E-iSLIP) lead to almost identical performance. Thus LIMIT can be used to avoid unfairness under extremely unbalanced traffic patterns, and hardly effects the service under balanced traffic patterns.

We also find that the performance of HE-iSLIP is very close to that of E-iSLIP under uniform traffic. In the EMHW the transmission of a packet can be interrupted when the match by Hamiltonian walk is selected. However, this does not happen very often under uniform traffic, so that the packet delay performance is almost unaffected by introducing the Hamiltonian walk. For example, simulation results show that for a 32x32 HE-iSLIP switch under uniform traffic, when the arrival rate is 0.9 or lower, Hamiltonian walk matches are picked only 0.03% or less of the time among all matches. When the arrival rate is 0.95, the fraction of Hamiltonian walk matches increases to only 0.68%, 0.21% and 0.09% for packet patterns 1, 2 and 3, respectively.

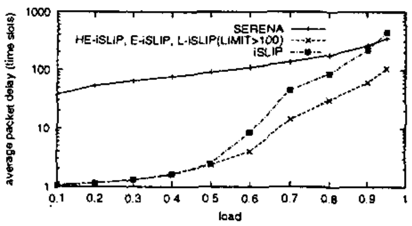

Fig. 1. The average packet delay of E-iSLIP, HE-iSLIP, iSLIP and SERENA under uniform traffic when the packet length is 1 cell.

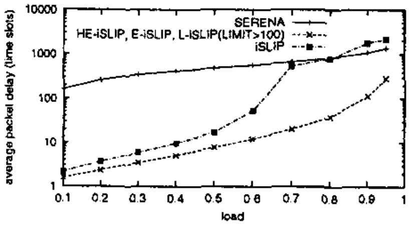

Fig. 2. The average packet delay of E-iSLIP, HE-iSLIP, iSLIP and SERENA under uniform traffic when the packet length is 10 cells.

The packet delay of ALGO3 is always much higher than 1000 cell time slots and are therefore not shown in the figures. We can see that HE-iSLIP always has the lowest packet delay, and is much lower than that of SERENA. In the figures, SERENA has the highest packet delay when the traffic load is low to moderately high. iSLIP has the highest packet delay under heavy load.

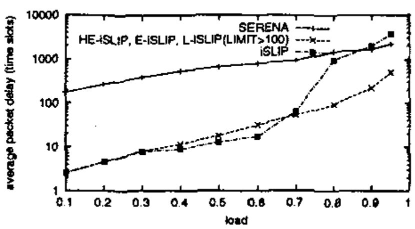

Fig. 3. The average packet delay of E-iSLIP, HE-iSLIP, iSLIP and SERENA under uniform traffic with variable packet length.

B. Under nonuniform traffic

Two typical nonuniform traffic patterns are considered in this paper. The first nonuniform traffic pattern considered in this paper is the diagonal pattern [18], [15]. The arrival rate for each input is the same. For input i a fraction f of arrivals are destined to output i, and all other arrivals are destined to output $(i + 1) modN$.

Table I shows the average delays of L-iSLIP with different LIMITs, HE-iSLIP, ALGO3 and SERENA, under diagonal traffic for all packets and for packets going to the lightly loaded and heavily loaded VOQs, respectively. The arrival rate to each input is 0.85. SERENA has the best delay performance under diagonal traffic, following by HE-iSLIP and L-iSLIP with a LIMIT of 100 cells. All three of them have similar delay when f is small. The delay of ALGO3 is much higher than those of other schemes.

Usually, the limited service policy with a large LIMIT is more efficient than that with a small LIMIT [20]. However, we can see that under diagonal traffic, a smaller LIMIT leads to better performance, especially when f is close to 0.5. The reason is as follows. With diagonal traffic, each input at most has two VOQs to be served. When f is close to 0.5, all VOQs have the same arrival rate. For a given input, if its two destination outputs are occupied by its two neighboring inputs for a long time, during which this input cannot get any service, the throughput and delay performances will suffer. If a smaller LIMIT is set, the destination outputs will become free more frequently so that the previously blocked input has a chance to be served. The performance can therefore be improved. When LIMIT is 100 cells, the delays for L-iSLIP are close to those for HE-iSLIP.

In the hotspot traffic pattern, the arrival rate for each input are identical. For input i a fraction p, $\frac{1}{N} \leq p < 1$, of arrivals are destined to output i, and other arrivals are uniformly destined to other outputs [15], [12]. Figure 4 shows the average delay of HE-iSLIP and SERENA for a 32x32 switch for different value of p when the arrival rate is 0.95 and the packet size is 1 cell. The simulation results show that HE-iSLIP always has lower delay than SERENA.

Compared to HE-iSLIP, the delay of SERENA is lower under diagonal traffic but higher under hotspot traffic pattern.

TABLE I

THE AVERAGE DELAY OF A 32X32 SWITCH UNDER DIAGONAL TRAFFIC PATTERN

f	0.1	0.2	0.3	0.4	0.5
ALGO3, all	1164	425	532	369	372
ALGO3, light	1856	1060	294	233	-
ALGO3, heavy	454	285	828	519	-
E-iSLIP, all	3.167	9.01	73.9	437	848
E-iSLIP, light	12.8	21.3	123	543	-
E-iSLIP, heavy	2.10	5.92	52.9	366	-
L-iSLIP, L=1000, all	3.17	9.01	73.7	341	464
L-iSLIP, L=1000, light	12.8	21.3	123	404	-
L-iSLIP, L=1000, heavy	2.10	5.92	52.7	300	-
L-iSLIP, L=500, all	3.17	9.01	64.7	206	251
L-iSLIP, L=500, light	12.8	21.3	107	236	-
L-iSLIP, L=500, heavy	2.10	5.92	46.7	187	-
L-iSLIP, L=100, all	3.17	9.08	30.9	53.1	56.7
L-iSLIP, L=100, light	12.8	21.5	47.0	59.4	-
L-iSLIP, L=100, heavy	2.10	5.97	22.6	48.9	-
HE-iSLIP, all	3.17	8.50	30.5	56.2	86.1
HE-iSLIP, light	12.8	20.0	48.5	70.6	-
HE-iSLIP, heavy	2.10	5.62	22.8	48.3	-
SERENA, all	3.95	6.61	8.57	9.52	10.1
SERENA, light	23.1	19.2	15.8	11.5	-
SERENA, heavy	1.82	2.47	5.47	6.56	-

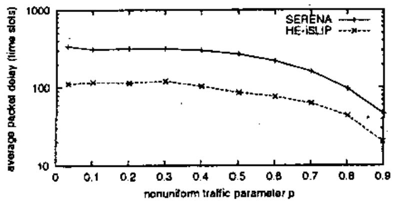

Fig. 4. The average packet delay of HE-iSLIP and SERENA under the hotspot traffic pattern.

The reason is as follows. SERENA takes the arrival pattern at each time slot into account to generate the new match. However, if there are more than one arrivals destined to the same output, only one of them, which is randomly selected, can be considered. Under diagonal traffic, only two inputs may have traffic to each output, so that the new match can adapt to the arrival pattern better. Indeed, SERENA is particularly suitable for a traffic pattern with which each output is always fed by only a few inputs. On the other hand, the performance of HE-iSLIP does not change much with different traffic patterns because the exhaustive service policy is used and the input-output matching usually adapts to the traffic pattern.

VI. CONCLUSIONS

In Exhaustive Service Matching, the cost of a match is amortized over multiple time slots, which significantly improves switch performance under uniform and nonuniform traffic. Since cells belonging to the same packet are transferred to the output continuously, the packet delay performance improves. Limited Service Matching and the Exhaustive Service Matching with Hamiltonian Walk (EMHW) class of scheduling schemes can avoid unfairness under some extremely unbalanced traffic pattern. We prove that EMHW is stable under all admissible traffic, regardless what matching algorithm is used, and is always as good as or better than ALGO3 in [14]. Exhaustive, Limited and Exhaustive with Hamiltonian walk service policies can be applied with existing matching algorithms. In this paper we apply them to iSLIP, and achieve very good packet delay performance with a low implementation complexity of $O(logN)$. We compare their performance with ALGO3 and SERENA, both of which are provably stable with implementation complexity $O(logN)$ and $O(N)$, respectively. The performance of SERENA was shown to be comparable to those of other complex but stable schemes. We show that HE-iSLIP has much better delay performances than ALGO3, while insuring stability and having comparable complexity. Compared to SERENA, which has a higher implementation complexity, HE-iSLIP has much lower delay under uniform traffic and the hotspot traffic pattern but higher delay under diagonal nonuniform traffic.

REFERENCES

[1] M. J. Karol, M. Hluchyj, and S. Morgan, "Input vs. output queuing on a space-division packet switch", *Proc. GLOBECOM* 1986, pp. 659-665.

[2] 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.

[3] 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.

[4] N. McKeown, A. Mekkittikul, V. Anantharam and J. Walrand, "Achieving *100% throughput in an Input-Queued switch", IEEE Trans. Communications*, vol. 47, No. 8, pp. 1260-1267, Aug. 1999.

[5] 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.

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

[7] 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.

[8] 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.

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

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

[11] 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.

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

[13] 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.

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

[15] Y. Li, S. Panwar, H. J. Chao, " The Dual Round-Robin Matching switch with exhaustive service," *2002 Workshop on High Performance Switching and Routing (HPSR 2002)*, May 2002.

[16] 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.

[17] A. Nijenhuis, H. Wilf, "Combinatorial algorithms: for comptuters and calulators," 2nd Edition, *Academic Press*, New York, 1978.

[18] M. A. Marsan, A. Bianco, P. Giaccone, E. Leonardi, F. Neri, "Packet Scheduling in Input-Queued Cell-Based Switches," *IEEE INFOCOM 2001*, vol. 2, pp. 1085-1094, April 2001.

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

[20] H. Takagi, "Queueing analysis of polling models: an update," *Stochastic Analysis of Computer and Communication Systems*, New York: Elsevier Science and B. V. North Holland, pp. 267-318 1990.