Performance Analysis of a Practical Load Balanced Switch

Yanming Shen, Shivendra S. Panwar, H. Jonathan Chao
Department of Electrical and Computer Engineering
Polytechnic University

***Abstract*— The load balanced (LB) switch proposed by C.S. Chang et al [1], [2] consists of two stages. First, a load-balancing stage spreads arriving packets equally among all linecards. Then, a forwarding stage transfers packets from the linecards to their final output destination. The load balanced switch does not need any centralized scheduler and can achieve 100% throughput under a broad class of traffic distributions. In this paper, we analyze a practical load balanced switch, called the Byte-Focal switch [3], which uses packet-by-packet scheduling to significantly improve the delay performance over switches of comparable complexity. We analyze the average delay for different stages in the Byte-Focal switch. We show that the average queueing delay is roughly linear with the switch size N and although the worst case resequencing delay is N^2, the average resequencing delay is much smaller. This means that we can reduce the required resequencing buffer size significantly.**

I. Introduction

Due to the memory speed constraint, most high speed packet switches use input buffering alone or in combination with other options, such as output buffering or cross-point buffering [4]. Input-buffered switches are required to resolve output contention, that is, find a match between inputs and outputs per time slot (see e.g., [5]–[10]). Thus, the issue of how to schedule packets efficiently to achieve high throughput and low delay for a large-capacity switch has been one of the main research topics in the past few years. Although several practical scheduling schemes have been proposed or implemented (for instance [11]–[14]), most of them require a centralized packet scheduler, increasing the interconnection complexity between the line cards and the packet scheduler, while some schemes need a speedup of up to two [15], [16] to compensate for deficiencies in packet scheduling. Due to the difficulty of scheduling its switch fabric, commercial high-speed switches typically cannot guarantee 100% throughput for all arrival traffic patterns, and this goal will become even more difficult in the future as the number of interfaces and the interface line speeds increase.

Recently, C.S. Chang et al introduced the Load Balanced Birkhoff-von Neumann switch architecture (LB-BvN) [1], [2]. This architecture is based on spreading packets uniformly inside the switch before forwarding them to their correct destination. As shown in Figure 1, the basic LB-BvN switch consists of two identical switches and does not need any scheduler. Each of the two switching stages goes through the same pre-determined cyclic shift configuration. Therefore, each input is connected to each output exactly $\frac{1}{N}th$ of the time, regardless of the arriving traffic. The first stage is a load-balancer that spreads traffic over all the second stage Virtual Output Queues (VOQs). The second stage is an input-queued crossbar switch in which each VOQ is served at a fixed rate. The LB-BvN switch does not require a centralized scheduler and at the same time, it can guarantee 100% throughput for a broad class of traffic. Therefore, load balanced switches are not subject to the two main problems commonly present in previous switch architectures: centralized scheduling and the lack of throughput guarantees. This makes the load balanced switch an appealing architecture to study.

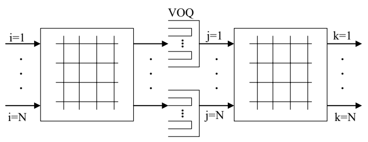

Fig. 1: The architecture of the load balanced Birkhoff-von Neumann switch

However, the FIFO discipline might be violated for packets from the same input in the load balanced switch. In its simplest form, the switch mis-sequences packets by an arbitrary amount. Several solutions have been proposed to tackle the unbounded out-of-sequence problem and can be categorized into two different approaches. The first approach is to prevent packets from being received out-of-sequence at the outputs (e.g., FFF (Full Frames First) [17], Mailbox switch [18]). The second approach is to limit the maximum resequencing delay to an upper bound, e.g., $O(N^2)$, and then add a resequencing buffer (RB) at the output to reorder the packets. Such schemes include FCFS (First Come First Serve) [19], EDF (Earliest Deadline First) [19], FOFF (Full Ordered Frames First) [20], and the Byte-Focal switch [3].

In this paper, we analyze the performance of the Byte-Focal switch, which is simple to implement and highly scalable. The rest of the paper is organized as follows. In Section II, we briefly present the Byte-Focal switch architecture. In Section III, we analyze the average delay at different stages. Section IV provides the simulation results and Section V concludes the paper.

II. The Byte-focal Switch Architecture

We briefly describe the Byte-Focal switch architecture in this section. The Byte-Focal switch is based on packet-by-packet scheduling to maximize the bandwidth utilization of

0-7803-9569-7/06/$20.00 ©2006 IEEE

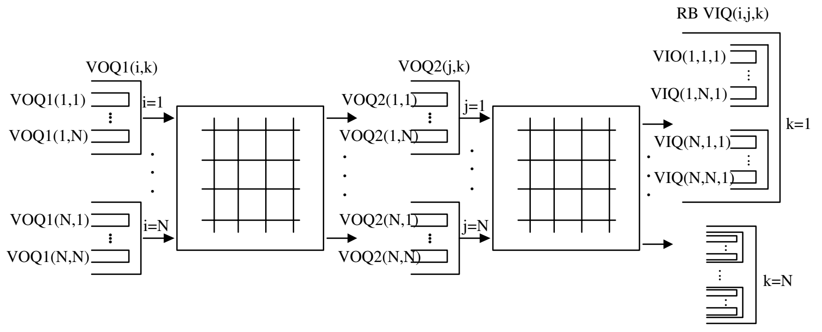

Fig. 2: The Byte-Focal switch architecture

the first stage and thus improve the average delay performance. Figure 2 shows the Byte-Focal switch architecture. It consists of two deterministic switch fabrics and three stages of queues, namely input queues i, center stage queues j, and output resequencing buffers (RB) k, where $i, j, k = 1, 2, \dots, N$. The two stages both use a deterministic and periodic connection pattern. That is, at the first stage, at any time slot t, the connection pattern (i, j) is given by

$$j = (i + t) \bmod N, \qquad (1)$$

$i = 1, \dots, N$ and $j = 1, \dots, N$. Similarly, the connection pattern (j, k) at the second stage satisfies

$$k = (t - j) \bmod N, \qquad (2)$$

$j = 1, \dots, N$ and $k = 1, \dots, N$.

A. First Stage Load-balancing

There are two stages of VOQs in the Byte-Focal switch, VOQ1 and VOQ2 for the first and second stage switch, respectively. We define the flow f_{ik} as the packets arriving at the input port i and destined to output port k. Packets from f_{ik} are placed in VOQ1(i, k). Each VOQ1(i, k) has a *J pointer* (see Figure 3) that keeps track of the last second stage input to which a packet was transferred, and the next packet is always sent to the next second stage input. As a head of line (HOL) packet departs from a VOQ1(i, k), its J pointer value increases by one mod N. When input i is connected with j, each VOQ1(i, k) whose J pointer value is equal to j sends a request to the arbiter, and the arbiter needs to select one of them to serve. Since at each time slot, the input port at the first stage is connected to the second stage cyclically, the packets in VOQ1(i, k) are sent to the N second stage input ports in a round-robin manner and are placed in VOQ2$(1, k)$, VOQ2$(2, k), \dots,$ VOQ2(N, k) according to their final destination. As a consequence of its mode of operation, the Byte-Focal switch guarantees that the cumulative number of packets sent to each second stage input port for a given flow differs by at most one.

In improving the average delay performance, the scheduling scheme used by the input arbiters at the first stage plays a very important role in the Byte-Focal switch. Since the packets in VOQ1(i, k) are cyclically distributed to the second stage, as

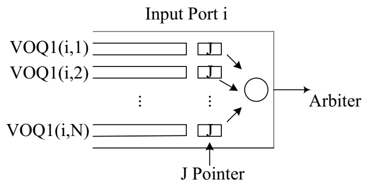

Fig. 3: Scheduling schemes at the first stage

a result, when the first stage input port i is connected to the second stage input port j, more than one of the VOQ1s at i may be eligible to send packets to j. The following algorithms for choosing a VOQ1 are presented in [3]:

- Round-robin: Among all eligible VOQ1s, the arbiter at each input port selects one of them in round-robin order.
- Longest queue first: At each time slot, the arbiter at each input port chooses to serve the longest queue from all eligible VOQ1s.
- Fixed threshold scheme: Set a predetermined threshold (TH), N. First serve the VOQ1s with queue length greater than N, if there is no such queue, then pick round-robin among the queues.
- Dynamic threshold scheme: Similar to fixed threshold scheme except that now the threshold is set to be $\lfloor Q(t)/N \rfloor$, where $Q(t)$ is the total VOQ1 queue length at an input port at time t.

As we can see, the Byte-Focal switch performs the first stage scheduling independently at each input port using locally available information. Thus, it does not need any communication between different linecards. It has been proved that the longest queue first, fixed threshold and dynamic threshold schemes are stable and the total queue length at each input port of the first stage is bounded by N^2 [3].

B. Second Stage Switching

Since the traffic to the second stage is uniform after the first stage load-balancing, the second stage just uses a fixed and deterministic configuration. As an example, consider the packets destined to output k. Assume at time t, VOQ2$(1, k)$ is connected to output k, then the HOL packet in VOQ2$(1, k)$ will be transfered to the output k. According to the connection pattern in equation (2), at time slot $t+1$, VOQ2$(2,k)$ is connected to output k, and the HOL packet is transfered to output k. We can see the VOQ2s are served in round-robin order. Furthermore, due to the first stage load-balancing, the queue length difference between VOQ's at the second stage is bounded by N [21].

C. Resequencing Buffer Design

Since the packets, in general, suffer different delays at the second stage, they arrive at the output out of order. The Byte-Focal switch uses the virtual input queue (VIQ) structure for the resequencing buffer(RB) (Figure 2). At each output, there are N sets of VIQs, where each set corresponds to an input port i. Within each VIQ set, there are N logical queues with each queue corresponding to a second stage input j. VIQ(i,j,k) separates each flow not only by its input port i, but also by its second stage queue j. Packets from input i destined to output k via second stage input j are stored in VIQ(i,j,k), and it is obvious that the packets in the same VIQ(i,j,k) are in order.

In each VIQ set, a pointer points to the VIQ(i,j,k) at which the next expected head of flow (HOF) packet will arrive. Because of the service discipline of the first stage switch, each input port evenly distributes packets in round-robin order into the second stage queue j. This guarantees that the HOF packet will appear as a head-of-line (HOL) packet of a VIQ set in round-robin order. Therefore, at each time slot, if the HOF packet is at the output, it is served, and the pointer moves to the next HOF packet location VIQ$(i,(j+1)\ mod\ N,k)$. Since there are N flows per output, more than one HOF packet may be eligible for service in a given time slot. Therefore, in addition to the VIQ structure, there is a departure queue (DQ) with a length of at most N entries that facilitates the round-robin service discipline. When the HOF packet of VIQ set i arrives, index i joins the tail of the DQ. When a packet departs from the DQ, its index is removed if its next HOF packet has not arrived, and joins the tail of the DQ if its next HOF packet is at the output. The advantage of using the VIQ and the DQ structure is that the time complexity of finding and serving packets in sequence is $O(1)$. At each time slot, each VIQ set uses its pointer to check if the HOF packet has arrived, while the output port serves one packet from the head of the DQ. As explained above, the VIQ structure ensures that the Byte-Focal switch will emit packets in order.

III. Average Delay Analysis

A packet in the Byte-Focal switch experiences queueing delays at the first and second stage, and also the resequencing delay at the output. In this section, we analyze the average delay of the Byte-Focal switch under uniform traffic. Throughout the analysis, we assume a uniform Bernoulli i.i.d traffic model with a rate of λ per time slot.

A. Second Stage Delay

Consider a VOQ2(j,k) and $a_{jk}(t)$ is the arrival to VOQ2(j,k) in time slot t. The input j at the second stage is connected to output k once in every N time slots. The arrival to this queue can be approximated by a Bernoulli process with a rate of $\frac{\lambda}{N}$. Assume at time T, the VOQ2(j,k) is connected to output k. Then we have the following recursive equation for VOQ2(j,k):

$$\begin{aligned}&q_{jk}(T+n)=q_{jk}(T)+\textstyle\sum_{s=1}^{n}a_{jk}(T+s),\\ &n=1,2,\ldots,N-1.\end{aligned} \tag{3}$$

and

$$q_{jk}(T+N)=\max\{q_{jk}(T)+\sum_{s=1}^{N}a_{jk}(T+s)\ -1,0\}. \tag{4}$$

This recursion has the solution [22]

$$E\{q_{jk}(T)\}=\frac{N-1}{N}\frac{\lambda^2}{2(1-\lambda)}. \tag{5}$$

Then the average queue length is

$$\begin{aligned}E\{q_{jk}(t)\}\quad &=\frac{1}{N}\sum_{n=0}^{N-1}E\{q_{jk}(T+n)\}\\ &=\frac{1}{N}\sum_{n=0}^{N-1}E\{q_{jk}(T)+\sum_{s=1}^{n}a_{jk}(T+s)\}\\ &=\frac{1}{N}\sum_{n=0}^{N-1}[E\{q_{jk}(T)\}+\frac{\lambda}{N}n]\\ &=\frac{N-1}{N}\frac{\lambda^2}{2(1-\lambda)}+\frac{\lambda(N-1)}{2N}\\ &=\frac{N-1}{N}\frac{\lambda}{2(1-\lambda)}.\end{aligned} \tag{6}$$

Applying Little's formula, the average queueing delay at the second stage is

$$d_2=\frac{N-1}{2(1-\lambda)}. \tag{7}$$

However, since the queue length difference is bounded by N [21] , when the Byte-Focal switch is heavily loaded, the average delay would be smaller than (7). Consider the system formed by all second stage queues, when the total number of packets is greater than N^2-N+1, then no queue is empty, and the system is work-conserving like an output-queued switch. Thus we have

$$q_k(t+1)=\max\{q_k(t)+a_k(t+1)-1,0\},$$

where $q_k(t)=\sum_{j=1}^{N}q_{jk}(t)$. The average queueing delay for an output-queued switch is simply

$$\frac{\lambda}{2(1-\lambda)}\frac{N-1}{N}.$$

Table I shows the simulation results and the analysis for the second stage queueing delay with a switch size of $N=16$. From the table, we can see when the load is low (less than 0.59), the average delay matches $\frac{N-1}{2(1-\lambda)}$ very well. However, when the loading is extremely high (0.99), the second stage queueing delay behaves like an output-queued switch.

Load	0.09	0.19	0.29	0.39	0.49	0.59	069	0.79	0.89	0.99
Simulation	8.23	9.21	10.40	11.87	13.73	16.11	19.22	23.34	29.26	49.16
$\frac{N-1}{2(1-\lambda)}$	8.24	9.26	10.56	12.3	14.71	18.29	24.19	35.71	68.18	750
$\frac{\lambda}{2(1-\lambda)}\frac{N-1}{N}$	0.05	0.11	0.19	0.30	0.45	0.67	1.04	1.76	3.79	46.41

TABLE I: Second Stage Queueing Delay

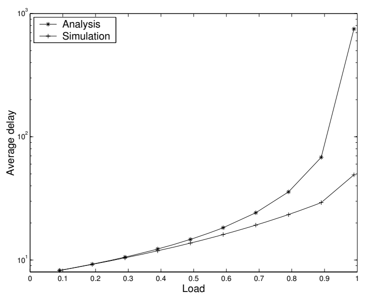

Fig. 4: First stage queueing delay

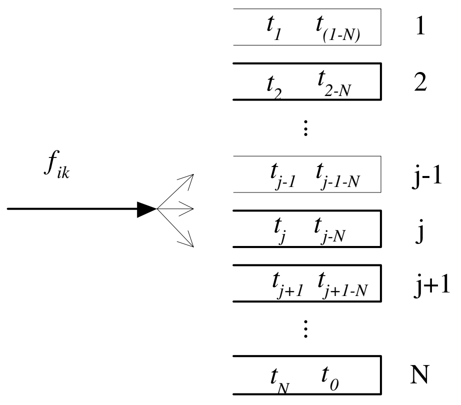

Fig. 5: Resequencing delay

B. First Stage Delay

Consider the first stage under uniform traffic. On the average, each VOQ1 is served once every N time slots, and the queues will behave similarly to a TDM system as in the second stage. Each VOQ1 has Bernoulli arrival with rate of $\frac{\lambda}{N}$ and an approximately deterministic service time of N time slots. Using the same derivation as in the second stage, the queueing delay at the first stage under uniform traffic is

$$d_1 = \frac{N-1}{2(1-\lambda)}. \tag{8}$$

Figure 4 shows the average first stage queueing delay for a switch size of $N = 16$.

C. Resequencing Delay

Consider a tagged packet (TP) [23] in flow f_{ik}. TP arrives at the second stage input j at frame time F_j (one frame time slot = N time slots). Because of the FCFS discipline, the packets which arrive at the switch after the TP cannot affect the departure time of TP from the resequencing buffer. So we only need to consider the packets which were in the switch when TP arrived. As shown in Figure 5, the packets in the same flow are cyclically distributed to the N intermediate second stage inputs, from 1 to N. The packets in the same VOQ are in order. The resequencing delay(RD) for the TP is only decided by the $N-1$ packets ahead of it, and these $N-1$ packets are sent to the second stage queue $j+1, \ldots, N, 1, \ldots, j-1$ respectively, which are denoted with $t_{j+1}, \ldots, t_{j-1}$. Let F_l denote the frame time that the packet arrived at second stage queue l, and $\hat{F}_l$ be the time that the packet leaves second stage queue l. Then the resequencing delay for the TP is

$$RD = \max(0, \max_l(\hat{F}_l - \hat{F}_j)), \tag{9}$$

where $l = j+1, \ldots, j-1$.

Since each VOQ at the second stage is work-conserving at each frame time slot, any packet in f_{ik} that arrives at the second stage input queue j, with queue length $q_{jk}(F_j)$, at time F_j will leave there at

$$\hat{F}_j = F_j + q_{jk}(F_j). \tag{10}$$

Then the resequencing delay becomes

$$RD = \max(0, \max_l[(F_l + q_{lk}(F_l)) - (F_j + q_{jk}(F_j))]). \tag{11}$$

Also, we have

$$F_l + q_{lk}(F_l) - [F_j + q_{jk}(F_j)] = q_{lk}(F_l) - q_{jk}(F_l)$$
$$+F_l - F_j + [q_{jk}(F_l) - q_{jk}(F_j)].$$

Since there is at most one departure per frame slot, $F_l - F_j + [q_{jk}(F_l) - q_{jk}(F_j)] \le 0$. Therefore,

$$RD \le \max(0, \max_l q_{lk}(F_l) - q_{jk}(F_l)).$$

At each frame slot, we have [21]

$$q_{lk}(F) - q_{jk}(F) \le \sum_{i=1}^{N}[A_{ilk}(F) - A_{ijk}(F)] - [A_{ilk}(S) - A_{ijk}(S$$

where $0 \le S \le F$, and $A_{ilk}(t)$ is the cumulative number of arriving packets from input i to output k via the second stage l. Since the packets in f_{ik} are sent to the second stage in round-robin order, from 1 to N, without loss of generality, assume l is a lower index than j, then we have

$$A_{ilk}(t) - A_{ijk}(t) = 1 \text{ or } 0.$$

Define $P(A_{ilk}(t) - A_{ijk}(t) = 1) = p$ and $P(A_{ilk}(t) - A_{ijk}(t) = 0) = 1 - p$. Note that $A_{ilk}(t) - A_{ijk}(t) = 1$ means from flow f_{ik}, there is one more arrival to the second stage l

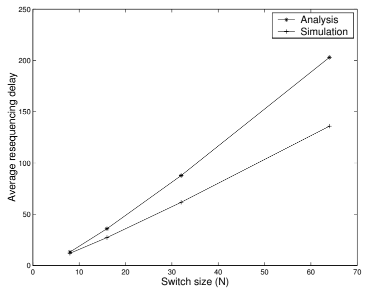

Fig. 6: Average resequencing delay vs. switch size N under loading $\lambda = 0.99$

than to the second stage k. Since the arrival rate of the flow f_{ik} is λ/N, we approximate p with $\lambda/N(1-\lambda/N)$. Define

$$X_i = [A_{ilk}(t) - A_{ijk}(t)] - [A_{ilk}(s) - A_{ijk}(s)],$$

then we have

$$X_i = \begin{cases} -1 & \text{with probability of } p(1-p) \\ 0 & \text{with probability of } p^2 + (1-p)^2 \\ 1 & \text{with probability of } p(1-p) \end{cases}$$

Let $Y_l = \sum_{i=1}^{N} X_i$, and Y_l is just the queue length difference between any two second stage VOQs. Then the average RD is

$$E\{RD\} \leq E\{(\max Y_l)^+\},\ l = 1, 2, \ldots, N-1, \qquad (12)$$

where $(X)^+ = \max\{0, X\}$.

According to central limit theorem, Y_l can be approximated as a normal distribution with mean $\eta = 0$ and variance $\sigma^2 = 2Np(1-p) = 2\lambda[1 - \lambda/N(1-\lambda/N)]$,

$$P(Y_l = k) \cong \frac{1}{\sigma\sqrt{2\pi}} e^{-k^2/2\sigma^2}, \quad k = -N, \ldots, 0, \ldots, N.$$

Then from (12), we can get the average resequencing delay bound. Figure 6 shows how the average resequencing delay grows with N. We can see that the average resequencing delay is roughly linear with the switch size N and it is much smaller than N^2.

IV. Simulations

In this section, we present some simulation results for the queueing delays at the first and second stage, and the resequencing delay. We use the following traffic scenarios to test the performance of the Byte-Focal switch:

Uniform: $\lambda_{ij} = \rho/N$.

Diagonal: $\lambda_{ii} = \rho/2$, $\lambda_{ij} = \rho/2$, for $j = (i+1)\ mod\ N$. This is a very skewed loading, since input i has packets only for outputs i and $(i+1)\ mod\ N$.

Hot-spot: $\lambda_{ii} = \rho/2$, $\lambda_{ij} = \rho/2(N-1)$, for $i \neq j$. This type of traffic is more balanced than diagonal traffic, but obviously more unbalanced than uniform traffic.

Normally, for single stage switches, the performance of a specific scheduling algorithm becomes worse as the loadings become less balanced.

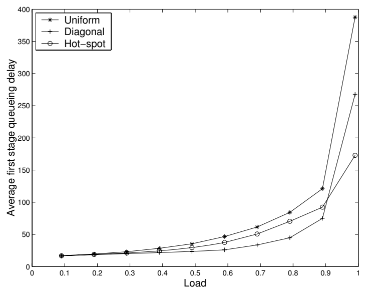

Fig. 7: Average first stage queueing delays for the dynamic threshold scheme with $N = 32$

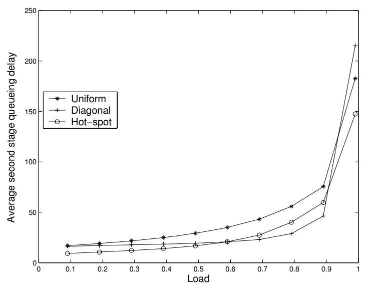

Fig. 8: Average second stage queueing delays for the dynamic threshold scheme with $N = 32$

In Figure 7 and Figure 8, we study the average first and second stage queueing delays of the dynamic threshold scheme under different input traffic scenarios for switch size $N = 32$. As the input traffic changes from uniform to hot-spot to diagonal (hence less balanced), the dynamic threshold scheme has comparable first and second stage queueing delays. Although our analysis is derived from uniform traffic, since the Byte-Focal switch performs load-balancing at the first stage, the average delays under nonuniform traffic is comparable to the uniform traffic.

In Section III-C, we analyzed the average resequencing delay. Although the worst case resequencing delay is N^2, the average resequencing delay is much smaller. This suggests that perhaps we can reduce the resequencing buffer size while maintaining an acceptable loss rate. Figure 9 shows the average resequencing delay under different input traffic for switch size $N = 32$. An interesting result is that the average resequencing delay under diagonal traffic is even lower than under the uniform traffic. This is consistent with our analysis, since under diagonal traffic, for a particular output k, only two inputs have cells destined to k, and the queue length difference at the second stage is reduced. This leads to a smaller resequencing delay. Figure 10 shows the loss rate for limited resequencing buffer sizes under uniform traffic. Note that when the loading to the switch is lower, then the loss rate would be smaller.

Figure 11 shows how the required resequencing buffer size grows with N with a loss rate at the order of 10^{-6}. We can see the required resequencing buffer size increases roughly linearly with N. This means if we want to size the

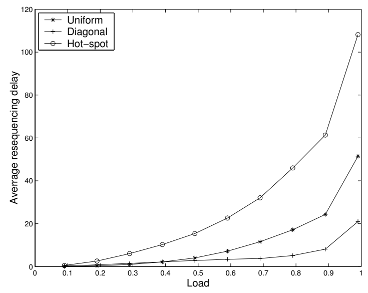

Fig. 9: Average resequencing delay for the dynamic threshold scheme with $N = 32$

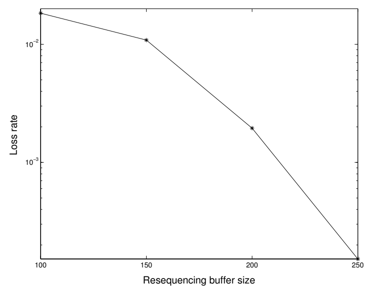

Fig. 10: Loss rate for limited resequencing buffer size for switch size $N = 32$ and $\lambda = 0.99$

resequencing buffer to meet an acceptable loss rate, e.g., 10^{-6}, then the required buffer size grows linearly instead of the worst case required resequencing buffer size N^2. This is a significant finding since it obviates one of the major drawbacks of LB-BvN switches.

V. Conclusion

In this paper, we analyzed the performance of a practical high performance load balanced switch architecture: the Byte-Focal switch. The Byte-Focal switch is distributed and does not need any communication between stages or linecards. We analyzed the three stage delays and show the queueing delays at the first stage and second stage are proportional to the switch size. For the resequencing delay, although the worst case resequencing delay is N^2, the average resequencing delay and maximum buffer requirement for given loss rate are much smaller. Our analysis shows that we can reduce the required buffer size significantly.

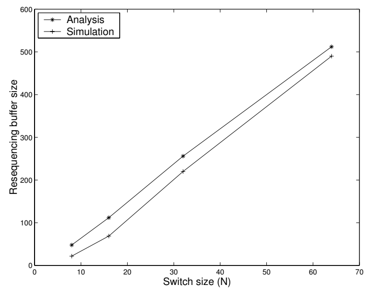

Fig. 11: Required resequencing buffer size vs. N for a loss rate of 10^{-6} and $\lambda = 0.99$

References

[1] C.S. Chang, W.J. Chen, and H.Y. Huang, "Birkhoff-von Neumann input buffered crossbar switches," in *Proceedings of IEEE INFOCOM*, 2000.

[2] C.S. Chang, D. Lee, and Y. Jou, "Load balanced Birkhoff-von Neumann switches, part I: one-stage buffering," *Computer Communications*, vol. 25, pp. 611–622, 2002.

[3] Y. Shen, S. Jiang, S. S. Panwar, and H. J. Chao, "Byte-Focal: A practical load-balanced switch," in *Proceedings of IEEE High Performance Switch and Routing*, Hong Kong, 2005.

[4] R. Cessa, E. Oki, and H.J. Chao, "CIXOB-k: Combined input-crosspoint-output buffered packet switch," in *Proceedings of IEEE Globecom Conference*, San Antonio, Texas, 2001.

[5] 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.

[6] D. Shah and M. Kopikare, "Delay bounds for approximate maximum weight matching algorithms for input queued switches," in *Proceedings of IEEE INFOCOM*, New York, 2002, pp. 1024–1031.

[7] V. Tabatabaee and L. Tassiulas, "MNCM: a new class of efficient scheduling algorithms for input-buffered switches with no speedup," in *Proceedings of IEEE INFOCOM*, 2003.

[8] J. G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," in *Proceedings of IEEE INFOCOM*, 2000, pp. 556–564.

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

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

[11] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Trans. Networking*, vol. 7, pp. 188–201, April 1999.

[12] H. J. Chao and J. S. Park, "Centralized contention resolution schemes for a large-capacity optical ATM switch," in *Proceedings of IEEE ATM Workshop*, Fairfax, VA, May 1998.

[13] Y. Li, S. S. Panwar, and H.J. Chao, "Exhaustive service matching algorithms for input queued switches," in *Proceedings of IEEE Workshop on High Performance Switching and Routing*, 2004.

[14] 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.

[15] B. Prabhakar and N. McKeown, "On the speedup required for combined input and output queued switching," *Automatica*, vol. 35, no. 12, December 1999.

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

[17] I. Keslassy and N. McKeown, "Maintaining packet order in two-stage switches," in *Proceedings of IEEE INFOCOM*, New York, 2002, vol. 2, pp. 1032–1041.

[18] C.S. Chang, D. Lee, and Y. J. Shih, "Mailbox switch: A scalable two-stage switch architecture for conflict resolution of ordered packets," in *Proceedings of IEEE INFOCOM*, Hong Kong, March 2004.

[19] C.S. Chang, D. Lee, and Y. Jou, "Load balanced Birkhoff-von Neumann switches, part II: multi-stage buffering," *Computer Communications*, vol. 25, pp. 623–634, 2002.

[20] I. Keslassy, S.T. Chuang, K. Yu, D. Miller, M. Horowitz, O. Solgaard, and N. McKeown, "Scaling internet routers using optics," in *Proceedings of ACM SIGCOMM*, Karlsruhe, Germany, Aug 2003.

[21] I. Keslassy, S.T. Chuang, K. Yu, D. Miller, M. Horowitz, O. Solgaard, and N. McKeown, "Scaling internet routers using optics (extended version)," Tech. Rep. TR03-HPNG-080101, Stanford University.

[22] Mischa Schwartz, *Broadband Integrated Networks*, Prentice Hall, 1996.

[23] N. Gogate and S.S.Panwar, "Assigning customers to two parallel servers with resequencing," *IEEE Communication Letters*, vol. 3, no. 4, pp. 119–122, April 1999.