

Research Article

Comparison of Memory Assignment Schemes for Switch Architectures with Shareable Parallel Memory Modules

Sanjeev Kumar¹ and Alvaro Munoz²

¹Department of Electrical /Computer Engineering, University of Texas-Pan American, Edinburg, TX 78541, USA

²Department of Electrical Engineering, University of Texas-Dallas, Richardson, TX 75080, USA

Correspondence should be addressed to Sanjeev Kumar, sjk@utpa.edu

Received 31 October 2009; Revised 15 April 2010; Accepted 21 April 2010

Academic Editor: Edward Au

Copyright © 2010 S. Kumar and A. Munoz. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Switching Architectures deploying shareable parallel memory modules are quite versatile in their ability to scale to higher capacity while retaining the advantage of sharing its entire memory resource among all input and output ports. The two main classes of such architectures, namely, the Shared Multibuffer-(SMB-) based switch and the Sliding-Window-(SW-) based packet switch, both deploy parallel memory modules that are physically separate but logically connected. In spite of their similarity in regards to using shareable parallel memory modules, they differ in switching control and scheduling of packets to parallel memory modules. SMB switch uses centralized control whereas the SW switch uses a decentralized control for switching operations. In this paper, we present a new memory assignment scheme for the Sliding-Window (SW) switch for assigning packets to parallel memory modules that maximizes the parallel storage of packets to multiple memory modules. We compare the performance of a sliding-window switch deploying this new memory assignment scheme with that of an SMB switch architecture under conditions of identical traffic type and memory resources deployed. The simulation results show that the new memory assignment scheme for the sliding window switch maximizes parallel storage of packets input in a given switch cycle, and it does not require speed-up of memory modules. Furthermore, it provides a superior performance compared to that of the SMB switch under the constraints of fixed memory-bandwidth and memory resources.

1. Introduction

Due to the nature of bursty traffic in Internet, the router/switch architectures allowing the sharing of memory resource among the output ports are well suited to provide the best packet-loss and throughput performance [1, 2] for a fixed size memory on a switching-chip. The slow speed of the memory elements limits the number of broadband lines that can share a common memory-resource, which in turn-limits the scalability of a shared-memory switch. Two well-known classes of switching architecture, namely, the Shared Multibuffer-(SMB-) based switch architecture [3, 4] and the Sliding-Window-(SW-) based packet switch architecture [5, 6] attempt to overcome this physical limitations of the memory-speed by deploying parallel memory modules that can be shared among all of input and output ports of these switches. Section 2 of this paper presents the switching scheme for the class of shared multibuffer

(SMB) switch architecture, which is used primarily for comparison purposes. In Section 3 of this paper, a new memory assignment scheme for the sliding-window switch architecture is presented, which is aimed at maximizing the parallel storage of packets to multiple memory modules in a given switch cycle. Section 4 presents the bursty traffic model that is used to evaluate the performance of the new switching scheme of the sliding-window switch and compare the performance with that of another well-known SMB switch. Performance results are discussed in Section 5, and Section 6 concludes the paper.

2. The Shared Multibuffer (SMB) Switch

The Shared-Multibuffer (SMB) switch architecture is discussed in detail in [3, 4]. Multiple memory modules are shared among all the input/output ports through cross-point

switches. The control for the SMB-based switching system is centralized and maintains a buffer address queue for each output port and the idle addresses pool to store the vacant addresses. For complete sharing of memory modules, the idle buffer and the address queue for each output port in SMB switch need to be as large as the total memory space. The centralized controller is responsible for coordinating all the switching functions for the SMB switching system, which in turn limits the scalability of this architecture. In SMB architecture [3, 4], an incoming packet is assigned to the least occupied memory module first. The least occupied memory module is given highest priority for an incoming packet to be written to. Based on the occupancy of the memory modules, it is possible to assign different memory modules to different packets input in a given switch cycle. This results in multiple packets to be stored in different memory modules in parallel, and hence requiring no increase in the memory bandwidth during the WRITE cycle. However, during the READ cycle, it is possible for multiple packets of the same memory module to be scheduled to go to different output ports during the same switch cycle. This requires multiple packets to be read out of the same memory modules in the same switch cycle, and hence this increases the memory-bandwidth requirement of the SMB switch during the READ cycle. Since memory speed-up is required only during the READ cycle, the memory-bandwidth requirement for a SMB switch can be defined as the number of memory READ cycles needed per switch cycle to output the scheduled packets from the memory modules.

3. The Sliding-Window (SW) Switch

The class of sliding-window (SW) switch is discussed in detail in [5, 6]. The class of the sliding-window switch is characterized by parallel memory modules and decentralized control. The self-routing parameter assignment circuit computes the self-routing parameters (i, j, k) to be attached to the incoming packets. The parameter j in the self-routing tag designates the packet's location in i th memory module, and parameter k value determines a packets turn to go out. The parameter assignment circuit first determines the j and k parameters and then uses the j and k values to determine the value of the i th parameter, that is, the memory module (i) where an incoming packet is stored [5]. The packets input in the same switch cycle are assigned different values of i (i.e., the i th memory module) in an increasing order. However, in the switching scheme given in [5, 7], it is possible for two or more incoming packets of a switch cycle belonging to different OSVs (j) to be scheduled to be written to the same memory module. This requires the switching scheme in [5, 7] to speed-up the memory modules to enable multiple-write operations in a given switch cycle. This in turn requires an increase in the memory-bandwidth requirement of an SW-switch during the WRITE cycle. The read operation of the SW-switch is such that no more than one packet is output from one memory module [5, 6]. This results in only one read operation per memory module per switch cycle. Hence in the sliding-window switch architecture, the memory speed-up is needed only during the WRITE cycle

but not during the READ cycle. The memory-bandwidth requirement for a sliding-window switch is simply the number of memory WRITE cycles needed per switch cycle to store incoming packets to memory modules.

Due to physical limitations, the memory-speed can be increased only so far. It is important to design a memory assignment scheme that maximizes the parallel write of packets to different memory modules in a given switch cycle without requiring an increase in memory speed-up.

3.1. A New Memory Assignment Scheme for SW Switch. This new memory assignment scheme aims at maximizing the parallel write of packets to multiple memory modules without requiring a speed-up of memory modules. According to this scheme, we have an additional Array called Temp $[i]$ for $i = 1$ to m , where m is the number of memory modules deployed in the switch. The Temp-array is used to keep track of the assignment of memory modules in a given switch cycle. Each switch cycle, the Temp array is initialized to 0 to indicate availability of a memory module for packet assignment. Each switch cycle, only one packet is assigned to a given memory module. According to this scheme, a packet is assigned to a memory module i if and only if the i th slot is available in OSV- j as well as the Temp array. Availability of a slot in both the OSV- j and the Temp array forces the packets of a switch cycle to be assigned to different memory modules and hence maximizing the parallel write of packets to different memory modules. If an i th slot is available in OSV- j but not available in the Temp Array, then the packet is simply dropped. It can be noticed that the packet is dropped instead of allowing it to access a preassigned memory module. This means that in any switch cycle no more than one incoming packet is written to a given memory module. This corresponds to one WRITE operation per memory module per switch cycle. This also means that the memory modules can operate at the line speed and do not need a speed-up.

4. Performance Evaluation

4.1. Bursty Traffic model. To study performance of these switching systems, a bursty traffic is generated using a two-state ON-OFF model, that is, by alternating, a geometrically distributed period during which no arrivals occur (idle period), by a geometrically distributed period during which arrivals occur (active period) in a Bernoulli fashion (Figure 2) [1, 5].

If p and r characterize the duration of the active and idle period, respectively, then the probability that the active period lasts for a duration of i time slots is given by

$$P(i) = p(1 - p)^{i-1} \quad \text{for } i \geq 1, \quad (1)$$

and the corresponding *average burst length* is given by

$$E_B[i] = \sum_{i=1}^{\infty} i \cdot p(i) = \frac{1}{p}. \quad (2)$$

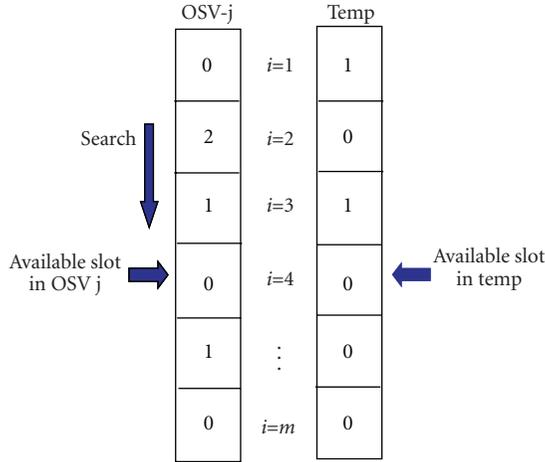


FIGURE 1: Assignment of a memory module for packets input in a given switch cycle.

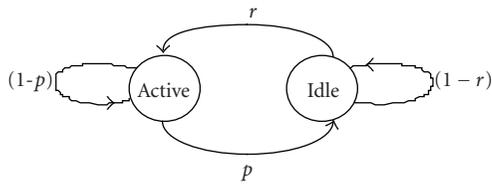


FIGURE 2: A two-state ON-OFF model.

Similarly, the probability that an idle period lasts for j time slots is

$$R(j) = r(1-r)^j \quad \text{for } j \geq 0, \quad (3)$$

and the corresponding mean idle period is given by

$$E_I[j] = \sum_{j=0}^{\infty} j \cdot R(j) = \frac{(1-r)}{r}. \quad (4)$$

Hence for a given p and r , the offered load L is given by

$$L = \frac{E_B[i]}{(E_I[i] + E_B[j])} = \frac{r}{r+p-rp}. \quad (5)$$

4.2. Simulation Setup. The measures of interest considered in the simulation studies are the offered load for a bursty traffic of a given average burst length (ABL), memory-bandwidth requirement, switch throughput, packet-loss ratio (PLR), and memory utilization of the switch. For memory bandwidth, the measurement was done for the average-case and the worst-case memory-bandwidth requirement. We used two traffic scenarios: balanced traffic where the incoming bursts of packets were uniformly distributed to the output ports and an unbalanced traffic scenario where half of the output ports had greater probability of receiving bursts of packets. The simulation experiment started out with empty memory modules. Depending on the offered load, a maximum of 64×10^5 packets were generated for evaluation of performance parameters of SMB and SW switches. A bursty traffic with an average burst length (ABL) of 4 packets was used for comparative evaluation.

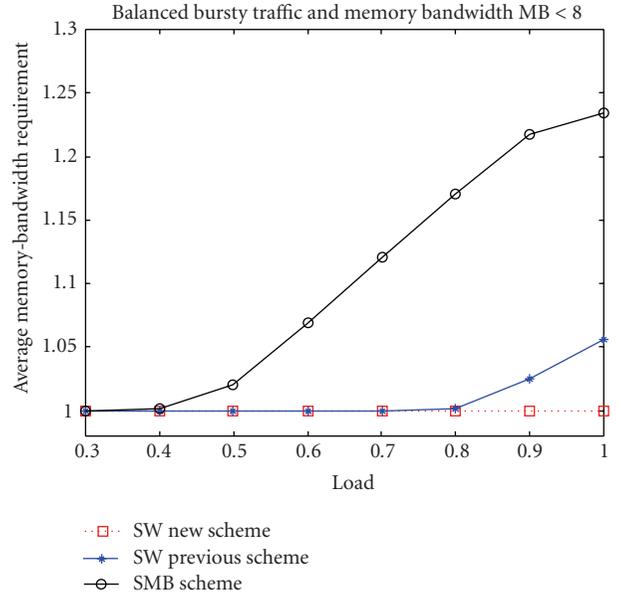


FIGURE 3: Average memory-bandwidth requirements of SW and SMB switches.

4.3. Switch configuration for SMB and SW Switch. The switch size of 32×32 was considered for both the SMB switch and the SW switch for comparative evaluation (in only one simulation study the size of the switch is increased to 64×64). The total shared-memory deployed in both switches equals 1024 packets, the minimum number of memory modules required according to the requirements given in [5] equals $m = 2N = 64$, and the packet-size of a memory module equals $\sigma = 16$ packets. A maximum queue length of 256 packets is allowed in both switches. For efficient sharing of common memory space among the output ports of these switches, the dynamic threshold scheme is used as given in [8] with $\alpha = 1$. Two types of results are presented to compare the SMB and SW switches: in one case, unlimited memory-bandwidth requirements, and in the second case, the most restrictive memory-bandwidth requirement $MB = 1$; that is, at most one READ/WRITE operation per switch cycle is allowed in any memory module.

5. Performance Results and Discussion

5.1. Balanced Bursty Traffic and Unlimited Memory Bandwidth. First, we measure the average memory-bandwidth requirement, worst-case memory-bandwidth requirement, throughput, packet loss ratio (PLR) for the same-size SMB-based switch, and SW-based switch under balanced bursty traffic conditions. It is observed in Figure 3 that the average memory-bandwidth requirement for the SMB switch is much higher than that of the SW switch with previous and new memory assignment scheme. Furthermore, unlike SMB switch, the memory-bandwidth requirement of the SW switch with new memory assignment scheme is always one (optimal). In Table 1, we measure the worst-case memory-bandwidth requirement in terms of the memory cycles

TABLE 1: Worst-case scenario for number of memory-cycles required per switch cycle for packets' READ or WRITE memory operations in SMB and SW switches, respectively.

Number of Memory Cycles	Load = 80%			Load = 100%		
	SW New Scheme	SW Previous Scheme	SMB Scheme	SW New Scheme	SW Previous Scheme	SMB Scheme
1 Cycle	100%	99.7046%	71.9705%	100%	89.4748%	64.3640%
2 Cycles	0%	0.2954%	24.6341%	0%	10.5016%	28.9760%
3 Cycles	0%	0%	3.1929%	0%	0.0237%	5.8846%
4 Cycles	0%	0%	0.1957%	0%	0%	0.7160%
5 Cycles	0%	0%	0.0066%	0%	0%	0.0561%
6 Cycles	0%	0%	0.0001%	0%	0%	0.0032%
7 Cycles	0%	0%	0%	0%	0%	0.0001%

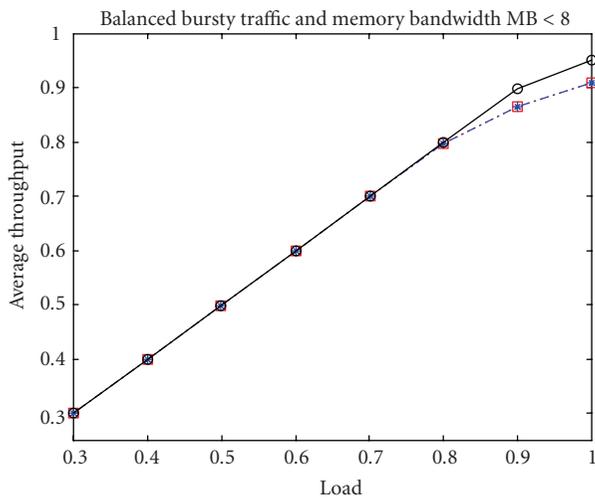


FIGURE 4: Average throughput with speed-up of memory modules.

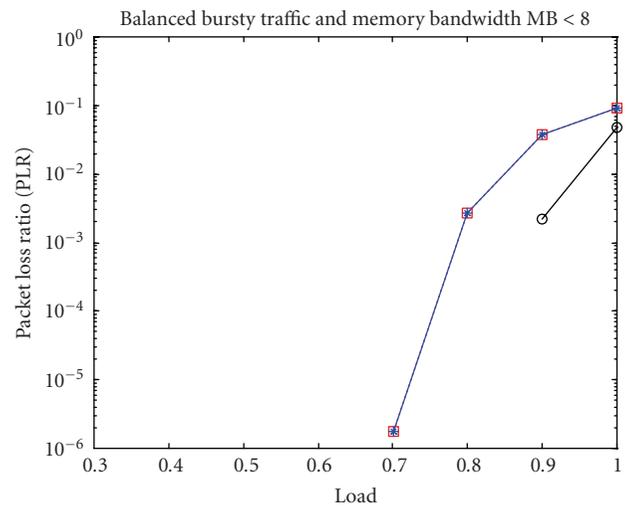


FIGURE 5: Packet-loss ratio with speed-up of memory modules.

required to write/read packets to/from the memory in a given switch cycle. The first column of Table 1 shows the number of memory-cycles used for READ/WRITE operations for packets belonging to a given switch cycle. The second column of the Table 1 shows the percentage of the switch cycles that needed a given number of memory-cycles (as indicated by a given row of the table) for READ/WRITE operations. In the worst-case scenario in Table 1, the SMB switch requires a maximum of 7 memory-cycles to be a nonblocking switch, whereas the SW switch with previous assignment scheme requires a maximum of 3 memory-cycles. The optimal SW switch with new memory assignment scheme requires only 1 memory-cycle for READ/WRITE operations for packets of a given switch cycle.

The SMB scheme (Figures 4 and 5) with unlimited memory-bandwidth resources achieves a better throughput performance and lower packet-loss ratio than the SW switch with previous and new assignment schemes. However, in a router/switch implementation the number of WRITE/READ

operations in memory modules is fixed. Thus, the SMB switch is not a scalable architecture when the size or speed of the system is increased. The difference among the two SW memory assignment schemes with unlimited memory bandwidth is negligible. In Figures 4 and 5, the performance of the SW switch with previous memory assignment scheme is slightly better (< 1%) compared to the SW switch with new memory assignment scheme.

5.2. Balanced Bursty Traffic and Memory Bandwidth MB = 1.

As the size of the switch increases, the memory-bandwidth requirement will also increase, which in turn imposes a physical limitation on the switch's scalability. Since the memory bandwidth cannot be increased beyond a certain point, the switches will have to operate with a fixed memory speed. Since the memory-bandwidth is usually fixed due to physical constraints, the performance of the SW switch is compared with that of SMB switch under conditions of a fixed memory bandwidth and balanced bursty traffic. For

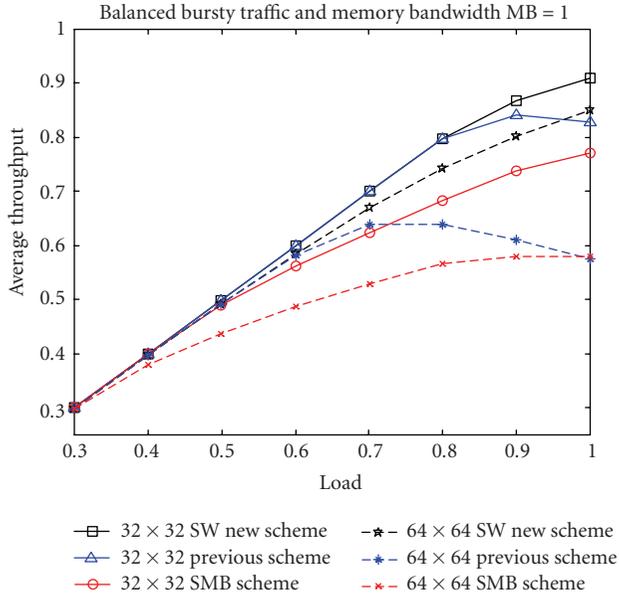


FIGURE 6: Average throughput with memory bandwidth ($MB = 1$) of SW and SMB switches.

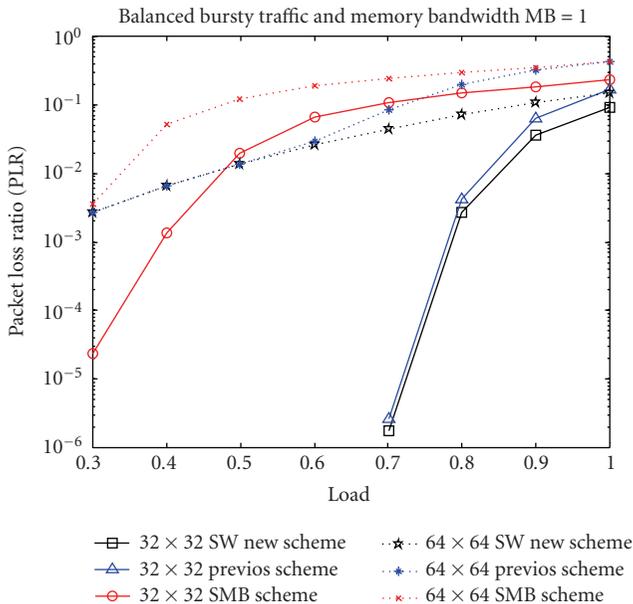


FIGURE 7: Packet-loss ratio with memory bandwidth ($MB = 1$) of SW and SMB switches.

comparison, we limit the switches' memory-bandwidth to one (i.e., the memory speed equals line speed). We then compare the throughput (Figure 6) and the packet-loss ratio (Figure 7) for the SW switch and the SMB switch under condition of a fixed memory bandwidth ($MB = 1$) (i.e., without any memory speed-up). The simulation employs two switch sizes (32×32 and 64×64) and a constant total buffer space of 1024 locations.

It is observed (Figure 6) that the throughput of the SW switch with any memory assignment schemes achieves a

much higher throughput compared to that of the SMB switch under conditions of identical switch size, memory size, traffic type, and the memory bandwidth. In these simulations, the memory bandwidth (MB) equals 1, that means the memory modules operate at the same speed as the line speed. Similarly, the packet-loss ratio (PLR) of the SW switch is much reduced (Figure 7) compared to that of the same size SMB-switch. However, the SMB switch and the SW switch with previous scheme have similar throughput performance (Figure 6) for a 64×64 switch size at full load (100%). The declining performance of the 64×64 SW switch with previous scheme at load $> 70\%$ (Figure 6) is because the number of memory modules ($m = 64$) is not at least the double of the switch size ($N = 64$) as specified in [5]. As a result, a lot of packets are dropped as the load increases since it becomes more difficult to distribute packets to different memory modules.

It can be noticed that the three memory assignment schemes deliberately drop packets of a switch cycle if they require more than one memory cycle to be stored/read to/from a given memory module. Nevertheless, the SW switch with new memory assignment scheme experiences far less packet-loss ratio (Figure 7) compared to that of the SMB switch and the SW switch with previous assignment scheme [5], especially at high loads.

5.3. Unbalanced Bursty Traffic and Unlimited Memory Bandwidth. In this section, we measure the performance of the SMB and SW switches with unbalanced bursty traffic and unlimited memory-bandwidth requirements. The unbalance-factor in the bursty traffic is varied to measure average throughput, packet-loss ratio, memory utilization, and average memory bandwidth under 70% and 100% switch loads. In the simulation experiment, half of the output ports of the switch have a greater chance of receiving burst of packets. As expected, the performance of both switch architectures decreases as the unbalanced factor increases in Figures 8 and 9. The SMB switch with unlimited memory-bandwidth requirements has a higher throughput (Figure 8) and lower packet loss ratio (Figure 9) compared to the SW switch. However, the difference in the performance of the SMB switch and the SW switch is small under unlimited memory bandwidth and unbalanced bursty traffic conditions. Also, the performance of the SW switch with previous memory assignment scheme is slightly better ($< 1\%$) compared to the SW switch with new memory assignment scheme. It is observed in Figure 10 that the memory utilization in the SMB switch is much higher than that of the SW switch. The superior memory utilization (Figure 10) in the SMB switch is the reason why it can achieve a better throughput performance and lower packet loss than the SW switch. However, this also causes in the SMB switch to have a much higher memory-bandwidth requirements compared to the SW switch. High memory-bandwidth requirement of SMB switch, which increases with the switch size, also makes SMB an impractical architecture to be deployed for large switch sizes. As the unbalanced factor varies in Figure 11, there is a small variation in the average

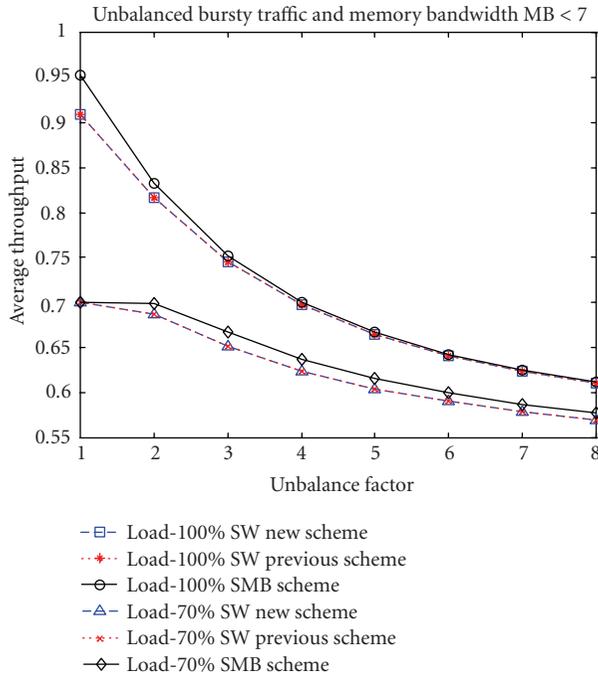


FIGURE 8: Average throughput with speed-up of memory modules.

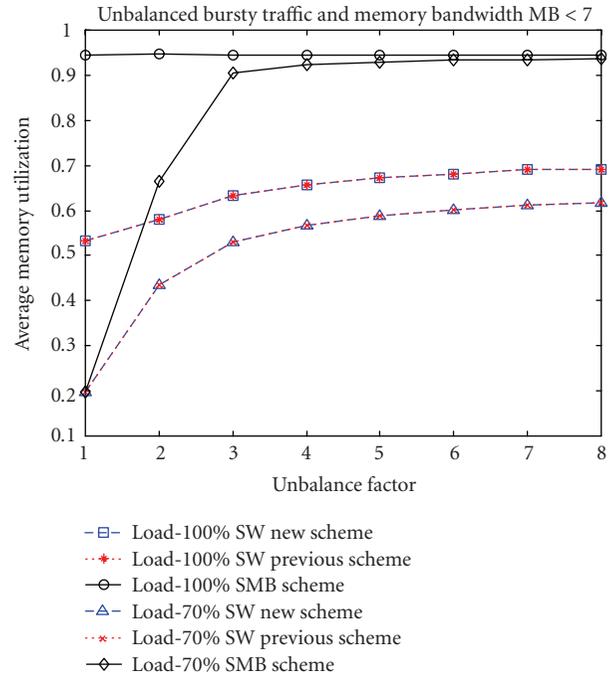


FIGURE 10: Average memory utilization with speed-up of memory modules.

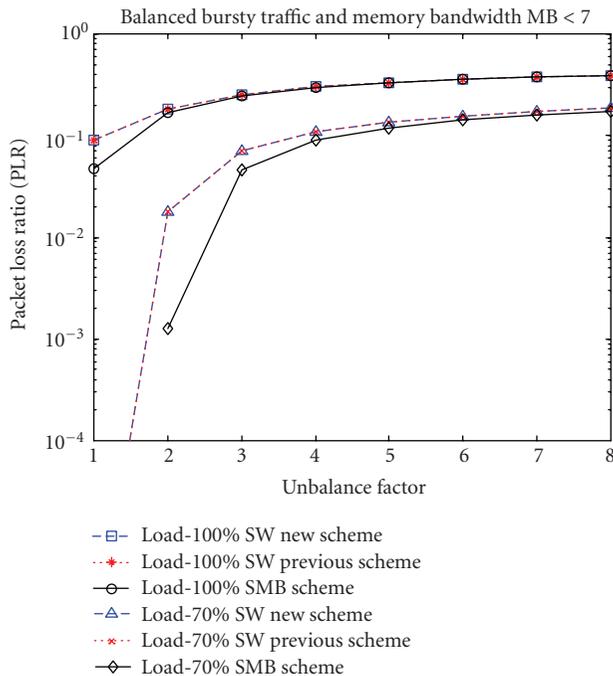


FIGURE 9: Packet-loss ratio with speed-up of memory modules.

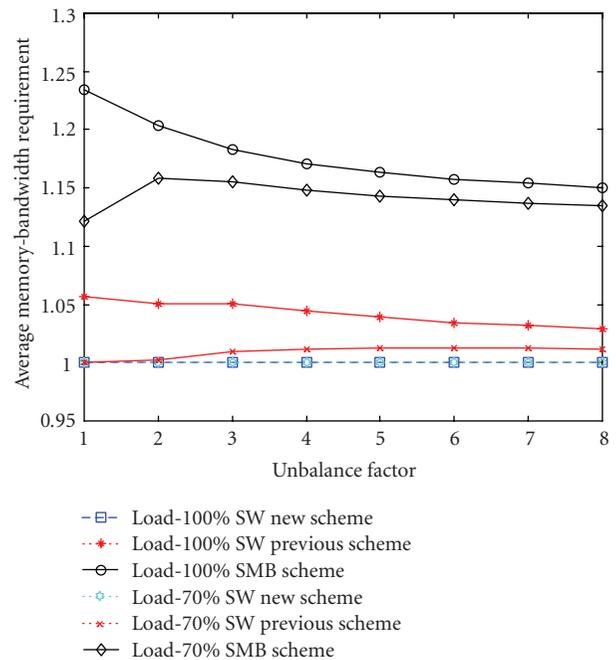


FIGURE 11: Average memory-bandwidth requirements of SW and SMB switches.

memory bandwidth in the SMB and SW switches, respectively. Nevertheless, always the average memory-bandwidth requirements (Figure 11) in the SW switch are much lower compared to the SMB switch. Furthermore, the memory-bandwidth requirement of the SW switch with new memory assignment scheme is always equal to 1 (best possible), which is independent of the switch size and traffic type.

6. Conclusion

In this paper, we present a new memory assignment scheme for the class of the sliding-window switch that aims at maximizing the switch throughput without requiring a speed-up of the memory modules. The speed of the memory modules is kept same as the line speed. That means that no more than

one WRITE/READ operations are performed per memory module per switch cycle. The new memory assignment scheme for the sliding-window switch architecture is used to compare its performance against another high-performance switch architecture that also deploys parallel and shareable memory modules, namely, the SMB switch [3, 4]. Also, we compare them with previous work of the sliding-window switch [5, 7]. We measure the average-case and worst-case memory-bandwidth requirements, throughput, packet-loss, and memory utilization performance of the SMB switch and SW switch (with new and previous memory assignment scheme) under conditions of identical switch size, memory size, memory speed, and traffic type. It is observed that under conditions of identical memory resources and traffic type, the class of sliding-window switch with the new memory assignment scheme has much reduced memory-bandwidth requirement compared to that of its SMB-based switch counterpart and previous SW work. Furthermore, for a fixed value of memory-bandwidth, the class of the sliding-window switch with new memory assignment scheme achieves higher throughput and smaller packet-loss when compared with those of a same size SMB-based switch [3, 4] and the previous SW memory-assignment scheme [5, 7].

References

- [1] H. J. Chao, C. H. Lam, and E. Oki, *Broadband Packet Switching Technologies: A Practical Guide to ATM Switches and IP Routers*, Wiley-Interscience, New York, NY, USA, 1st edition, 2001.
- [2] M. I. Irland, "Buffer management in a packet switch," *IEEE Transactions on Communications*, vol. 26, no. 3, pp. 328–337, 1978.
- [3] K. Oshima, H. Yamanaka, H. Saito, et al., "A new ATM switch architecture based on STS-type shared buffering and its implementation," in *Proceedings of the 14th International Switching Symposium (ISS '92)*, vol. 1, pp. 359–363, October 1992.
- [4] H. Yamanaka, H. Saito, H. Kondoh, et al., "Scalable shared-buffering ATM switch with a versatile searchable queue," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 5, pp. 773–783, 1997.
- [5] S. Kumar, "The sliding-window packet switch: a new class of packet switch architecture with plural memory modules and decentralized control," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 4, pp. 656–673, 2003.
- [6] A. Munoz and C. D. Cantrell, "Memory-configuration and memory-bandwidth in the Sliding-Window (SW) switch architecture," in *Proceedings of the 52nd Midwest Symposium on Circuits and Systems (MWSCAS '09)*, pp. 288–291, August 2009.
- [7] S. Kumar and A. Munoz, "Performance comparison of switch architectures with shareable parallel memory modules," *IEEE Communications Letters*, vol. 9, no. 11, pp. 1015–1017, 2005.
- [8] A. K. Choudhury and E. L. Hahne, "Dynamic queue length thresholds for shared-memory packet switches," *IEEE/ACM Transactions on Networking*, vol. 6, no. 2, pp. 130–140, 1998.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

