Buffer Architectures and Routing Algorithms in the Performance of Shufflenet[*†]

Shueng-Han Gary Chan and Hisashi Kobayashi
Department of Electrical Engineering
Princeton University
Princeton, NJ 08544
U.S.A.

Abstract

Shufflenet achieves high throughput by allowing different users in the network to transmit information concurrently through different channels. Because optical memory can be expensive, a shufflenet with deflection routing has been proposed. We investigate four control strategies (CS) using different routing algorithms and buffer architectures. The performance of the control strategies is studied by simulating in 64-node (2,4) shufflenet with deflection routing. Tradeoffs between throughput, buffer cost and routing complexity in the network are observed and discussed. We conclude that CFDL ("Care" packet First, "Don't care" packet Last) is a very effective routing algorithm. The shufflenet performs very well with deflection routing, even when only a few buffers are provided at each node in the network.

1 Introduction

Shufflenet [1] or a shufflenet graph [2] refers to a highly regular topology that has been suggested for multi-hop lightwave networks. This graph is a re-circulating perfect shuffle interconnection pattern [3]. The perfect shuffle is widely used as a pattern to interconnect processors to form a multi-processing computer [4]. An efficient implementation of shufflenet is realized either in bus topology [3] or in ring topology [1], hence it is suggested to provide a migration path for Fiber Distributed Data Interface (FDDI) or Distributed Queuing Dual Bus (DQDB) [5]. With the shufflenet, the vast bandwidth of fiber is used to overcome current shortcomings of device technology.

A shufflenet is a slotted network. Packets can be transmitted in the network in a store-and-forward fashion, if sufficient buffer storage is provided. Each user in the shufflenet has a number of optical receivers and transmitters. A shufflenet is characterized by two parameters p and k. A (p, k) shufflenet consists of kp^k nodes arranged in k columns, and each column consists of p^k nodes. All the nodes are interconnected as a perfect shuffle, with the last column being "wrapped-around" to the first column like a completed cylinder [1, 3]. In this way, packets can be continuously circulated around the network until they reach their destinations.

In any multi-connected mesh network in which each node has p input channels and p output channels, deflection routing can be used. When more than one packet contend for the same output channel and there is no storage available, deflection routing resolves the conflict by routing all but one of the packets into wrong channels. As a result, the "deflected" packets have to take more hops to reach their destinations. In this way, deflection routing avoids packet loss due to buffer overflow by sacrificing some bandwidth of the fiber.

Optical buffers may be used in high speed networks [8, 9]. Optical buffers outperform electronic buffers by eliminating O/E (optics to electronics) and E/O (electronics to optics) conversion of packet data, thus eliminating the "electronic bottleneck". This will ultimately lead to an improvement of a few orders of magnitude in the throughput of the network compared with that of the current network.

As optical buffers and control are expensive, it is of significant interest to minimize the number of optical buffers and the complexity of control in an optical network while maintaining the performance of the network. Several queuing schemes and buffer architectures have therefore been proposed [7, 9]. In deflection routing, a routing and buffer scheme that can achieve a low deflection probability is highly sought, as the performance of the network degrades with an increase in the probability of deflection [10, 12].

We report simulation results for four control strategies based on different routing algorithms and buffer architectures. In the shufflenet we analyse, we assume $p = 2$, although our analysis and simulation method can be easily generalized to cases in which $p > 2$. The performance of the strategies will be discussed and compared. We have analysed the performance of the network using the Markov chain analysis [10, 12]. Our simulation results agree very well with the analysis. We conclude that a shufflenet with deflection rout-

*The work was supported in part by the New Jersey Commission on Science and Technology's Innovative Partnership Program, FY 1993 and in part by the Ogasawara Foundation of Science and Technology, Japan.

†*Proceedings of Singapore International Conference on Networks and International Conference on Information Engineering (SICON/ICIE'93)*, Vol.1, pp.34-38, Singapore, September 6-11, 1993.

ing performs very well even when only one buffer is allocated in each node of the network.

2 Network Control Strategies

We propose and simulate four types of control strategies (CS) in a 64-node $(2,4)$ shufflenet with different buffer sizes. Figure 2 shows three steps, which are performed at each node of the network in each clock cycle of our simulation. The packets are routed using the shortest path routing algorithm [13]. The three steps performed in each time epoch are:

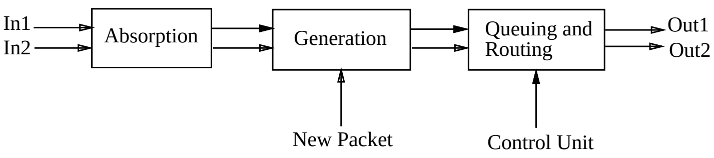

Figure 1: Simulation steps performed at each node in the $(2,4)$ shufflenet.

1. Absorption – The incoming packets are first checked for their addresses. Packets that are destined to the node are absorbed (i.e., delivery of a packet to its destination node). We assume that absorption can be done in parallel (i.e., on both links) in a single clock cycle. No packet will therefore be left unabsorbed if it is already at its destination node.

2. Generation – If there is one or no packet waiting to be queued or routed after the absorption stage, a new packet is generated at the node with probability g. We call g the offered load of the network. The destination for the newly generated packet is assumed to be uniformly distributed among all the other $(kp^k - 1)$ users in the network. The generated packet will be routed with other packets, if any, at the node within the same clock cycle.

3. Queuing and Routing – The packet(s), new or old, will then be queued or routed according to the control strategies we propose. The performance of the network is directly dependent on the choice of the strategy. The control unit in this step has to keep track of the status of the packets and set the appropriate switches in the memory within each clock cycle.

The architectures for the optical storage units (OSU) used in the network are shown in Figures 2, 3 and 4, which are called OSU-I, OSU-II and OSU-III, respectively and have been proposed and analysed for Manhattan Street Networks by Chlamtac and Fumagalli [9]. Buffer of sizes 0 (i. e., the case of hot-potato routing), 1, 2, 4, 8 and buffer of infinite size (i. e., the case of store-and-forward routing) are used in our simulation models. OSU-I and OSU-II both use 3×3 optical switches while OSU-III uses 2×2 switches. The optical delay line (ODL) is an optical fiber with the right length to delay the packet by one time slot.

In OSU-I, packets in the memory can be continuously circulated in the buffer. This may pose a problem in a practical

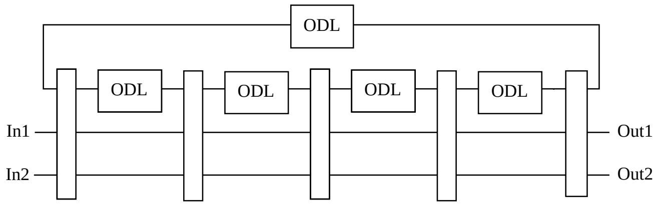

Figure 2: OSU-I with buffer size equal to 5.

system because the time that a packet can spend in the memory cannot be long due to the attenuation of signals in the fiber. Though semiconductor or Erbium-doped amplifiers could be used to extend the delay limit, this would undesirably increase the complexity of the system. Furthermore, internal noise generated by the amplifiers will ultimately limit the length of time a packet can stay in the optical memory.

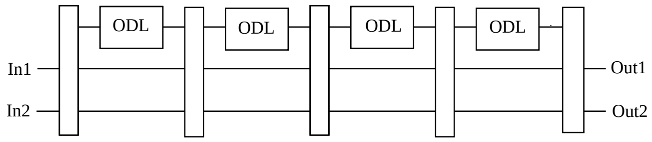

Figure 3: OSU-II with buffer size equal to 4.

OSU-III is the most feasible and least expensive memory available with the current technology. However, the access to the memory elements is not so easy compared with the other buffer architectures, especially when more than one ODL (i. e., multiple optical buffers) are considered. Both OSU-II and OSU-III relieve the packet circulation problem of OSU-I by limiting the time a packet can stay in the memory to the size of the memory.

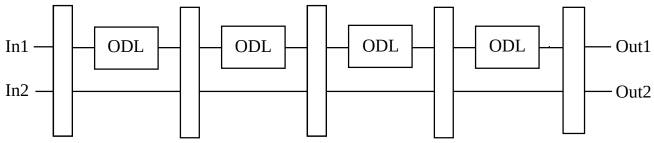

Figure 4: OSU-III with buffer size equal to 4.

We divide the packets into three classes: "don't care" packets, "care" packets destined to one of the output links, and "care" packets destined to the other output link[1] [10, 11]. A packet is said to be "hot" if the packet is at the right-most position in the memory so that the packet can no longer stay in the optical switch and has to be routed [6, 9]. Obviously there can be no hot packet in OSU-I buffer architectures. We now investigate the performance of shufflenet with four control strategies (CS), each of which is some form of the First-In-First-Out (FIFO) scheme:

- CS1 – Strictly FIFO among all the packets regardless of their classes, using storage architecture OSU-II

[1] A "care" packet in a (p,k) shufflenet is a packet which is within k hops from its destination, if no deflection occurs. Thus a "don't care" packet is more than k hops away from its destination assuming no deflection.

Incoming packets are randomly placed into the memory in each time epoch and routed in a strictly FIFO fashion. The packet that occupies the right-most memory has the highest priority and will be routed first. Therefore, the packet will be released and routed to its correct output link into the network. Then the next packet, if any, on its *immediate* left will be tested for collision. If this packet can be routed without collision, it is also released into the network. Deflection occurs when there are two incoming packets, the buffers are full and the two right-most packets in the memory contend for the same output link.

• CS2 – FIFO within each packet class, CFDL, using storage architecture OSU-I

In this strategy, "care" packets take priority in routing over "don't care" packets. This is called CFDL ("Care" packet First, "Don't care" packets Last) [6]. It is based on the observation that "care" packets are the only packets that are deflected. "Don't care" packets never contend for an output channel and hence can always be routed without deflection. Therefore, whenever a "care" packet can be routed instead of a "don't care" packet, it should be transmitted to avoid a possible deflection in the future. "Don't care" packets in the memory can be transmitted in this strategy only when there are no "care" packet in the memory or all the "care" packets are destined for the same output link. Deflection occurs only when two packets are coming in, the buffers are full and all the packets in the node, including the two incoming packets, are all "care" and they are all bound for the same output link.

• CS3 – FIFO within each packet class, CFDL, using storage architecture OSU-II

The strategy resembles CS2 except that a "don't care" packet can now leave the memory whenever it is hot. This shortens the lifetime that "don't care" packets can spend in optical memory.

• CS4 – FIFO within each packet class, CFDL, using optical buffer of type OSU-III

The most practical 2×2 optical switch is used. However, the memory elements cannot be accessed flexibly. Note that in each clock cycle, whenever there is no "hot" packet in the memory, only one packet can be transmitted. This adversely affects the throughput of the network. Shifting of the packets in the memory (hence the setting of the various switches) is more complicated in this strategy compared with the other strategies. The left-most buffer has to be occupied whenever there are two packets coming in after the generation stage.

3 Simulation Results and Discussions

• CS1

Figure 5 shows the simulation results for CS1 with different buffer sizes. There exists an optimal offered load, g^*, such that the throughput of the network is maximized. Under heavy traffic, deflections become excessive and the performance of the network deteriorates. Because of the indiscrimination among the three packet classes, buffers fill up very quickly even under moderately heavy traffic and performance deteriorates. We note that under very high load (i. e., as g approaches 1), buffer of any sizes does not improve the performance of the network, because all buffers become full. Instability is also observed in the case of infinite buffer size, as buffer occupancy grows without bound under heavy traffic. The packet delay, i. e., the time a packet takes to reach its destination, is observed to increase rapidly once the offered load exceeds g^* [8]. Therefore, though CS1 offers the simplest routing and queuing scheme and may be suitable for transmission of time-sensitive information, it is not a very good strategy for heavy load traffic.

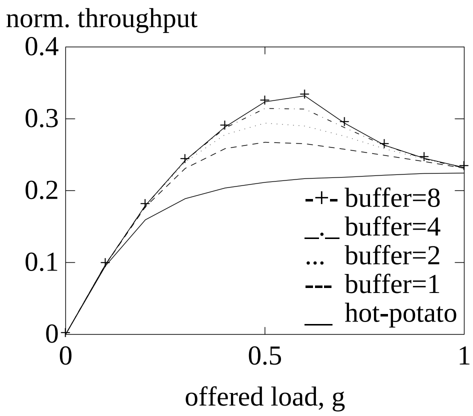

Figure 5: Normalized throughput vs. offered load, g, in the network for CS1 using OSU-II with hot-potato routing and various buffer sizes of 1, 2, 4 and 8.

We explain the optimal offered load, g^*, by noting that throughput depends on two opposing factors in the network: packet generation rate, g, and deflection probability. The initial increase in throughput with the offered load is due to the increased number of generated packets. However, as the offered load is increased, buffers become full and the deflection problem becomes serious. As packets are put into the buffers randomly and accessed in a strictly FIFO fashion, lots of packets are doomed to be deflected once the buffers become full; consequently the performance deteriorates. Under heavy load, packets get deflected more often and just circulate in the network without being absorbed. As g approaches 1, the performance of the network almost reduces to the case of hot potato routing (i. e., the case with no buffer), no matter how large the buffer size may be. One way to decrease deflection, and hence enhance the network performance, is to alternate "care" packets destined to different output channels and/or to alternate "care" and "don't care" packets when they are placed into the buffers. However, this inevitably increases the complexity of the queuing discipline.

• CS2

Figure 6 shows the performance of the shufflenet with CS2. Control strategy CS2 achieves the maximum throughput among all the control strategies with the same buffer size. The average delay is also minimal among all the strategies due to the small deflection probability. Control and access of the memory elements are simple in the scheme, though the 3×3 switches used in the memory buffer can be expensive in reality. This network has another undesirable feature: it may take a long time for "don't care" packets to get out of the memory under heavy traffic. This is due to the fact that in order to minimize deflections, "don't care" packets are routed with the lowest priority. Without any time-out control in the memory, "don't care" packets will stay in the memories far longer than the "care" packets. Because in any time epoch, a "don't care" packet at a given node is likely to have already taken more hops than a typical "care" packets to go to its destination and it may visit more "don't care" nodes in the next few hops, it will take a long time for the "don't care" packets to arrive at their destinations with this control strategy.

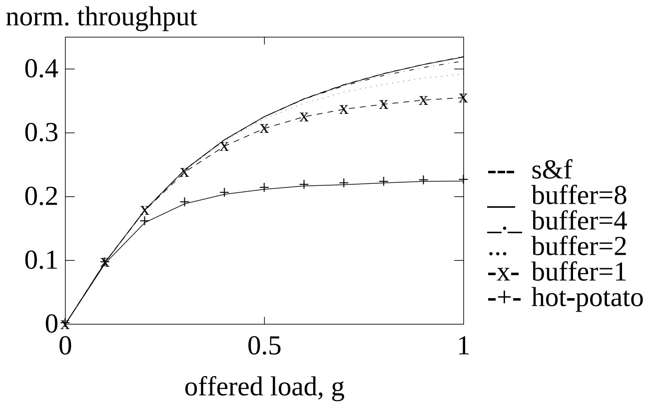

Figure 6: Simulation results for CS2 using OSU-I for the cases of hot-potato routing, buffer sizes of 1, 2, 4 and 8, and store-and-forward routing (i. e., the infinite buffer case).

- CS3

The performance of CS3 is very similar to that of CS2 as shown in Figure 7. However, because of the upper bound to the duration a "don't care" packet can stay in the memory, the performance is slightly worse than CS2. We therefore see a trade-off between the length of time a packet can stay in the memory and the performance of the network.

- CS4

Figure 8 shows the performance of the network with CS4. This control strategy also imposes an upper bound to the length of time a packet can stay in the memory and adopts the simplest type of optical switches. However, heavy burdens are placed on the control unit for queuing and routing. The shifting of packets in the memory is not so flexible as in CS2 and CS3. The performance of the network in terms of throughput and average delay is observed to be somewhat worse than CS3.

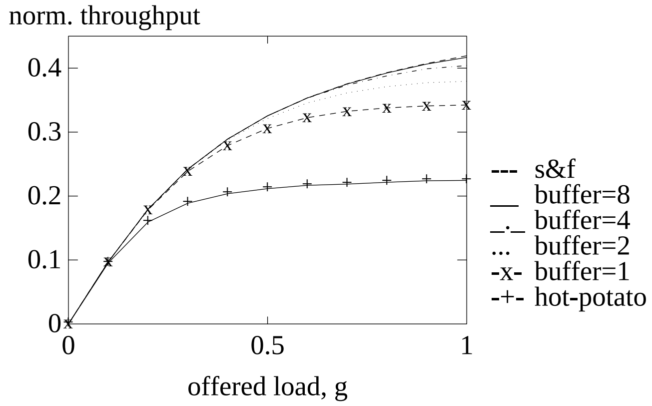

Figure 7: Simulation results for CS3 with OSU-II with buffer sizes of 1, 2, 4 and 8, along with hot-potato and store-and-forward routing cases.

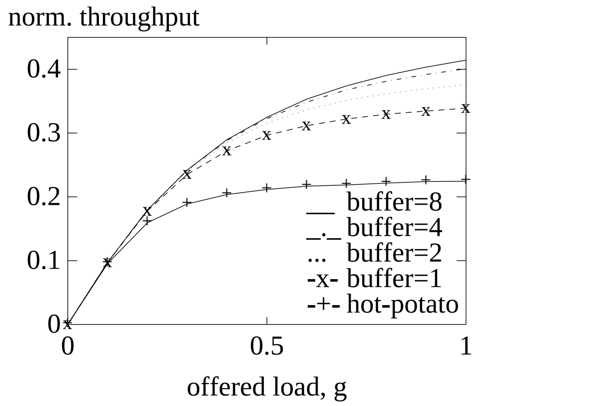

Figure 8: Simulation results for CS4 using OSU-III with buffer sizes of 1, 2, 4 and 8, along with hot-potato and store-and-forward routing cases.

From the result above, we see that there are clear trade-offs between the routing complexity (control complexity), the switch cost, the buffer size and the performance of shufflenet. From the case of CS4, we see that the routing complexity can be traded against the buffer cost to achieve a high level of performance. The network performance is also critically dependent on the the routing algorithm. This is seen from the performance of CS1. Even using high-performance 3×3 optical switches and with sufficient buffer size, the performance of CS1 is considerably worse than the other control strategies. The CFDL control strategy is concluded to be a very effective routing algorithm.

From our simulation results in Figures 6, 7 and 8, we note that buffer size of eight can achieve performance almost comparable with that of the network with infinite buffer size, i. e., the conventional store-and-forward routing. Therefore, in shufflenet with deflection routing, the cost of providing more buffering beyond eight may not be justifiable. We see from the figures that deflection is basically resolved with buffer size as low as four. Even with only one buffer, more that 80% of the throughput compared with store-and-forward case can be attained in the network. Shufflenet with deflection routing therefore performs very well in high-speed communication with memory capacity constraint.

In addition to the simulations discussed above, we have also conducted analytical studies for all the control strategies proposed. By modeling the memory transition in shufflenet as a Markov process, its steady state behavior can be obtained. From this, the probability of deflection of the packets in the network, P_{def}, can be found by solving a transcendental equation. The performance of the network can then be obtained solely through this parameter [10]. Figure 9 shows the analytic results for one-buffer cases of all the control strategies, along with hot-potato and store-and-

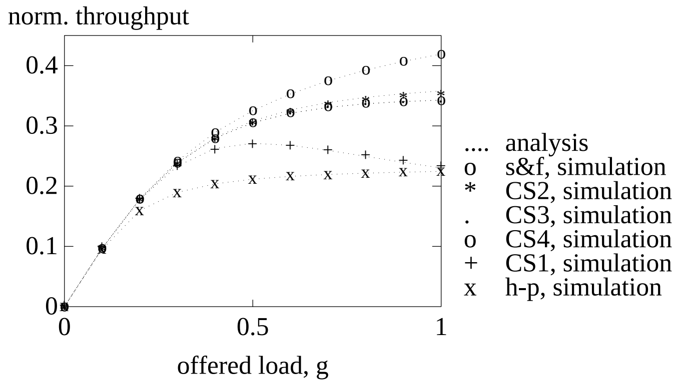

Figure 9: Analytic and simulation results for hot-potato routing, store-and-forward routing and one-buffer cases for all the control strategies proposed.

forwards cases [10]. A Markov chain analysis of the model discussed in the present paper will be reported elsewhere in the near future.

4 Conclusion

We have made a comparative study of four network control strategies with different routing algorithms and buffer architectures. There is a very strong trade-off between the control complexity and the buffer cost in the performance of a shufflenet. CFDL is found to be a much better strategy compared with the strictly FIFO algorithm.

We presented the performance analysis of the shufflenet with deflection routing in terms of its normalized throughput. In a separate study, we have also shown that other important performance measures, such as hops distribution and average delay, are related to the normalized throughput through a single probability measure – the probability of deflection of the packets in the network, P_{def} [10, 12]. Consequently, these important network performance measures can be deduced easily from the network throughput. We conclude that shufflenet performs very well with deflection routing, using only a few buffers at each node of the network.

Acknowledgement
We thank Mr. Alberto Bononi for valuable discussions and comments he provided in the course of this study.

References

[1] A. Acampora and M. Karol, "An Overview of Lightwave Packet Networks," *IEEE Network*, pp. 29-41, Jan. 1989.

[2] P. E. Green, Jr., *Fiber Optic Networks*, Prentice Hall, 1993.

[3] A. Acampora, "A Multichannel Multihop Local Lightwave networks," in *Proc. IEEE GLOBECOM '87 Conf.*, pp. 1459-1467, Nov. 1987.

[4] H. S. Stone, "Parallel Processing with Perfect Shuffles," *IEEE Trans. on Computer*, vol. 20, no. 2, pp. 153-161, 1971.

[5] M. Karol and R. Gitlin, "High-Performance Optical Local and Metropolitan Area Networks: Enhancement of FDDI and IEEE 802.6 DQDB," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 8, pp. 1439-1448, Oct. 1990.

[6] I. Chlamtac and A. Fumagalli, "Quadro: A Solution to Packet Switching in Optical Transmission Networks," to appear in *Computer Networks and ISDN Systems* Invited Paper.

[7] M. Hluchyj and M. Karol, "Queuing in High-Performance Packet Switching," *IEEE JSAC*, vol. 6, no. 9, pp. 140-150, 1988.

[8] S-H. Chan, "All-Optical High-Speed High-Capacity Network: ShuffleNet," ELE 497 – Senior Independent Work, Department of Electrical Engineering, Princeton University, Fall, 1992.

[9] I. Chlamtac and A. Fumagalli, "An All-Optical Switch Architecture for Manhattan Networks," to appear in *IEEE J. Selected Areas in Communication.*

[10] S-H. Chan and H. Kobayashi, "Performance Analysis of ShuffleNet with Deflection Routing," to appear in IEEE GLOBECOM'93.

[11] F. Forghieri, A. Bononi and P. Prucnal, "Analysis and Comparison of Hot-Potato and Single-Buffer Deflection Routing in Very High Bit Rate Optical Mesh Networks," to appear in *IEEE Transactions on Communication.*

[12] S-H. Chan, "ShuffleNet with Deflection Routing: Performance and Analysis," ELE 498 – Senior Independent Work, Department of Electrical Engineering, Princeton University, Spring, 1993.

[13] M. Karol and S. Shaikh, "A Simple Adaptive Routing Scheme for ShuffleNet Multi-hop Lightwave Networks," *IEEE GLOBECOM'88 Conf. Rec.,* pp. 1640-1647, Nov. 1988.