Analysis of the Shufflenet with Different Contention Resolution Schemes*

Shueng-Han Gary Chan†
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
Email: shgchan@leland.stanford.edu
and
Hisashi Kobayashi
Department of Electrical Engineering
Engineering Quadrangle
Princeton University
Princeton, NJ 08544
Tel: (609) 258-1984
Fax: (609) 258-3745
E-mail: hisashi@ee.princeton.edu

Abstract

In our earlier study [1], we investigated the performance of a shufflenet with different control strategies using different buffer architectures and routing algorithms through simulation. In this paper, we present the analysis of the control strategies proposed in [1]. By modeling the system as a Markov process, the performance of the shufflenet can be derived through only one parameter – the probability of deflection, P_{def}, in the network. In the analysis of buffered shufflenet with two packet classes (i.e., "care" and "don't care" classes), the state space for the memory transition increases generally with the order of $O(n^2)$, where n is the buffer size in each node in the network. However, by making use of the particular structure of the memory, we can reduce the state space to $O(n)$, which greatly simplifies the analysis of a shufflenet. We show how the analysis can be applied to one of the control strategies, namely CS-2, which achieves the highest throughput among all the control strategies we proposed.

We also report the performance of a shufflenet which adopts a simple priority contention resolution rule in which the packet in its last hop to its destination is given the highest priority in output channel selection. From the simulation of a (2, 4) shufflenet with deflection routing, we observe that the priority scheme can improve throughput of the shufflenet with hot-potato routing by about 10%. Our analysis shows very good agreement with the simulation result.

1 Introduction

Optical fiber possesses a tremendous amount of bandwidth for communication [2]. In an optical network, multiple access schemes such as time-division multi-access (TDMA) and wavelength-division multi-access (WDMA) have been proposed to tap effectively the vast bandwidth of optical fiber. As an optical TDMA system must meet stringent synchronization and timing requirements [3, 4, 5], while the technology of optical transmitters and receivers are better developed, most of the proposed optical networks make use of WDMA to achieve high throughput [6, 7].

Because of the enormous usable bandwidth offered by optical fiber, any communication scheme that utilizes only a small fraction of the bandwidth can provide a substantial capacity. A multi-hop scheme therefore is proposed to overcome the device technology limit at the expense of the achievable bandwidth [7]. A shufflenet structure is one of the schemes to achieve a high throughput.

In any multi-connected mesh network such as shufflenet 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 correctly routing one of the packets and temporarily routing the rest of them into wrong channels. As a result, the "deflected" packets take more hops to reach their destina-

*This 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 for the Promotion of Science and Technology.

†The author was with Princeton University when this research was performed.

tions. In this way, deflection routing avoids potential packet loss due to buffer overflow. A priority contention resolution (CR) rule is a rule whereby packets are ranked or prioritized for channel selection when contentions occur. Different priority CR schemes have been proposed in deflection routing [8, 9] to meet certain performance requirements. It has been shown in Manhattan Street Network (MSN) that the delay and throughput of the network can be shaped through priority CR rules [10].

We first discuss the shufflenet system and review the models used in analysing the $(2, k)$ shufflenet. We have done simulations and analysis of the performance of the shufflenet using priority and non-priority CR schemes. The performance of different CR rules with different buffer architectures and routing algorithms has been simulated in [1]. In this paper, we present an analysis of the control strategies (CS) we proposed. By modeling the memory transition in a shufflenet as a Markov process, its steady state behavior can be obtained. From this, the probability of deflection for a "care" packet in the network, P_{def}, can be found by solving a "self-consistent" equation. The performance of the network is then obtained solely through this parameter [11].

In the analysis of memory transition in the deflection network in which there are two packet classes (i.e., "care" and "don't care" packets in the network), the state-space increases according to $O(n^2)$ in general, where n is the buffer size. However, by making use of the particular structure of the memory, the state space can be reduced to $O(n)$, which greatly simplifies the analysis of the network. We will show how this can be done in one of the control strategies, i.e., CS2, with arbitrary buffer size. The decrease in state space leads to simpler and complete analysis for the control strategy.

We finally discuss the performance of a $(2, k)$ shufflenet that uses the following simple priority CR rule: packets which are in their last hops to their destinations are assigned the higher priority in output channel assignment than the packets not in their last hops. In case of contentions among packets with equal priority, the packets are deflected with equal probability. We have conducted both simulation and analysis for such CR rule with a (2,4) shufflenet when no buffer is provided (i. e., hot-potato routing). The throughput of the network using this scheme can be improved by as much as 10% compared with the non-priority scheme. Our analysis shows very good agreement with the simulation result.

2 Shufflenet System

Each user in a shufflenet accesses the network through the Network Interface Unit (NIU). Each NIU has a number of lightwave receivers and transmitters [13]. A shufflenet is characterized by two numbers p and k. A (p, k) shufflenet consists of kp^k nodes arranged in k columns, and each column consists of p^k NIUs. Figure 1 shows a (2, 3) shufflenet. All the NIUs are interconnected as perfect shuffle, with the

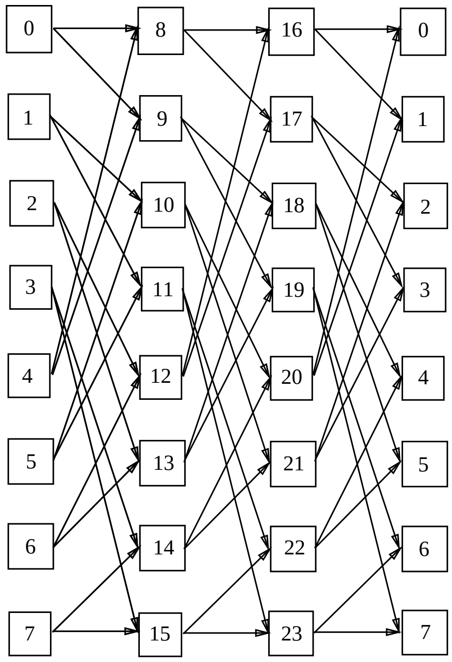

Figure 1: A (2, 3) shufflenet.

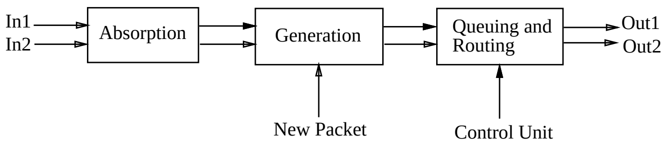

Figure 2: Steps a packet undergoes in the $(2, k)$ shufflenet.

last column being "wrapped-around" to the first column like a completed cylinder. In this way, packets can be continuously circulated around the network until they reach their destinations. In a (p, k) shufflenet, a total of kp^{k+1} channels have to be provided. The channels can be in the form of wavelengths or fibers [14]. We analyze the shufflenet with $p = 2$, but our analysis method can be generalized to $p > 2$.

In the (p, k) shufflenet, a packet at a node is said to be "don't care" with respect to its destination if the destination cannot be reached within k hops. A node is said to be a "don't care" node for a packet if the packet at the node can go from this node to its destination with the minimum number of hops by taking any link emanating from this node. Therefore, if a packet is at a "don't care" node, it will not be deflected. Similarly, a "care" packet in a (p, k) shufflenet is a packet that is within k hops from its destination. For a given packet, all the nodes that are within k hops from its destination are "care" nodes.

Figure 2 shows the procedure which each packet in the shufflenet undergoes in each node in each clock cycle [1, 16]. There are three steps involved in handling packets at the network node:

1. Absorption – The incoming packets are first checked for their addresses. Packets that are destined to the node are

absorbed (i.e., delivery of the packet to its destination node). We assume that absorption can take place on both links within a single clock cycle. No packet will therefore be left unabsorbed if it is already at its destination node.
2. Generation – We assume that each node in the network generates a packet with probability g in each time slot. If two packets are arriving at this stage, the newly generated packet will be discarded. Therefore we call g the normalized offered load. 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 that we propose [1] and are outlined in Section 4 and 5. Storage elements or memory buffers reside in this block. The control unit must keep track of the status of each packet in the memory and set the appropriate switches in the memory. The packet is then routed to an output channel which minimizes the number of hops to its destination [17].

3 System Analysis

3.1 Models

The "one node model" and "one packet model" are usually used in the analysis of multi-connected mesh network [19, 20, 21]. We will adopt the models in analyzing the shufflenet, making the following assumptions by referring to Figure 2:

- A node receives a packet on an incoming link independent to the fact whether there is another packet on the other link;
- A packet, after the generation stage, is equally likely to take any one of the two output links, independent of other packets in the node;
- On its way to its destination, a "care" packet in a node is deflected with probability P_{def}, and routed correctly to an output channel with probability $(1 - P_{def})$.

With the above assumptions, the network performance can be analyzed by focusing on only one node in the network, since all the nodes in the shufflenet are equivalent to each other.

3.2 General Analysis of Shufflenet

We pick a typical packet in a $(2, k)$ shufflenet and call it our "tagged" packet. Let P_{def} be the probability of deflection when the packet is in one of its "care" nodes and $S = \{0, 1, ..., 2k-1\}$ be a collection of possible distances between the current position of the packet and its destination. Here the distance is defined as the minimum number of hops that the packet must make to reach its destination without being deflected. We further let

$$D_i = \text{E[number of hops when the tagged packet is at distance } i \text{ from its destination} \mid \text{probability of deflection} = P_{def}], \quad \forall i \in S. \tag{1}$$

and

$$N_i^{dc} = \text{E[number of ``don't care'' nodes that the tagged packet at distance } i \text{ visits in its lifetime]}, \tag{2}$$

Then the expected delay is given by [11]

$$\mathrm{E}[D] = \frac{1}{kp^k - 1}\left[\sum_{j=1}^{k-1} p^j D_j + \sum_{j=0}^{k-1}(p^k - p^j)D_{k+j}\right], \tag{3}$$

and the expected number of "don't care" nodes that the packet hops through can be obtained as

$$\mathrm{E}[N^{dc}] = \frac{1}{kp^k - 1}\left[\sum_{j=1}^{k-1} p^j N_j^{dc} + \sum_{j=0}^{k-1}(p^k - p^j)N_{k+j}^{dc}\right] \tag{4}$$

The probability of "don't care" is defined by [11]

$$P_{dc} = \frac{\mathrm{E}[N^{dc}]}{\mathrm{E}[D]}. \tag{5}$$

The throughput of the $(2, k)$ shufflenet has been derived in [16]. Let α be the probability of absorption of a packet in the network. Then obviously,

$$\alpha = \frac{1}{\mathrm{E}[D]}. \tag{6}$$

Let λ be the normalized throughput (or carried load) of the network, which is defined as the probability that a node in the network absorbs a packet per unit time. Therefore the fraction $(g - \lambda)/g$, where g is the offered load of the network, represents the probability that a newly generated packet is discarded. Let u be the probability that any link in the network contains a packet. Then [16],

$$u = \frac{\sqrt{\alpha^2 + g^2(1-\alpha)^2} - \alpha}{g(1-\alpha)^2}, \tag{7}$$

and

$$\lambda = 2\frac{\sqrt{\alpha^2 + g^2(1-\alpha)^2} - \alpha}{g(1-\alpha)^2}\alpha. \tag{8}$$

Note that Equation (8) is a function in α (the absorption probability) and g (the offered load). As α is related to $E[D]$ according to Equation (6), the analysis of the shufflenet is reduced to the problem of finding the relationship between the offered load, g, and the average number of hops, $\mathrm{E}[D]$. The relationship depends on the routing algorithm and buffer

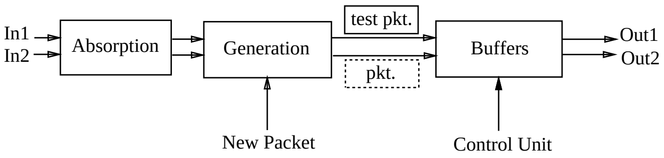

Figure 3: Diagram to obtain the probability of deflection, P_{def}, in the network.

architectures. Note that $\mathrm{E}[D]$ is in turn dependent on P_{def}. Therefore, the analysis of the shufflenet is reduced to finding the dependence between g and P_{def}. Once P_{def} is known for a given g, $E[D]$ can be obtained and so α can then be found. The throughput can then be calculated using Equation (8).

We find the deflection probability for a packet in the network by considering the deflection of the "test" packet in Figure 3, which shows the procedure that a packet goes through in each node in each clock cycle. The concept of "test" packet was introduced in [19] and discussed in [16]. Note that the nature of the "test" packet is no different from the "tagged" packet discussed earlier. A "test" packet is also a typical packet in the network. The switch of notation from the "tagged" packet to the "test" packet is mainly due to the fact that we are going to "test" our "tagged" packet for contention conditions. The testing will lead to an nonlinear equation in P_{def}, the solution of which is the probability of deflection of a packet in the network. We will show how this can be done in the following sections.

Referring to Figure 3, we define P_{pkt} as the probability that another packet is present in the other link of the node right before the buffer stage (i.e., the "Queuing and Routing" stage in Figure 2). From [16], P_{pkt} can be expressed as:

$$P_{pkt} = u(1-\alpha) + u\alpha g + (1-u)g. \tag{9}$$

Denote P_c as the probability that the packet on the other link of the "test" packet is also a "care" packet. Then,

$$P_c = [u(1-\alpha) + u\alpha g + (1-u)g]\,(1-P_{dc}), \tag{10}$$

where P_{dc} is given by Equation (5).

Let P_u be the probability that an input link of a node is occupied by a "care" packet. Then [16],

$$P_u = u(1-P_{dc}), \tag{11}$$

where u is given by Equation (7). Then the probability, $P_u^{'}$, that the input link is occupied by a "don't care" packet is

$$\begin{aligned} P_u^{'} &= uP_{dc} && (12)\\ &= u - P_u. && (13) \end{aligned}$$

4 Analysis of Control Strategies

In a (p, k) shufflenet, if packets are deflected with equal probability whenever output channel contention occurs, we have [11],

$$\begin{aligned} \mathrm{E}[D] \;=\; & -\frac{kp(1-p^{k-1}(1-P_{def})^{k-1})}{(kp^k-1)(1-P_{def})^{k-1}(1-p+pP_{def})} \\ & + \frac{kp^k}{kp^k-1}\left(\frac{k-1}{2}\right) + \frac{k}{(1-P_{def})^k}, \qquad (14) \end{aligned}$$

and

$$\begin{aligned} \mathrm{E}[N^{dc}] \;=\; & \mathrm{E}[D] - \frac{p}{P_{def}(kp^k-1)}\left[\frac{p^{k-1}-1}{p-1} - \right. \\ & \left. \frac{1-p^{k-1}(1-P_{def})^{k-1}}{(1-P_{def})^{k-1}(1-p+pP_{def})}\right] - \\ & \frac{1-(1-P_{def})^k}{(1-P_{def})^k P_{def}}. \qquad (15) \end{aligned}$$

For a buffered shufflenet, the state of the buffer can be modeled as a Markov chain. As there are 2 packet classes in the defection network – "care" and "don't care" packet, the state of the memory element in general can be labelled with 4 states: '0' (element empty), 'c1' (element occupied by a "care" packet destined for output link 1), 'c2' (element occupied by a "care" packet destined for output link 2) and 'd' (element occupied by a "don't care" packet in the memory). By symmetry, the states 'c1' and 'c2' are equivalent and we have $P_{c1} = P_{c2}$, where P_{c1} and P_{c2} are the probability that the memory element is occupied by a "care" packet destined for output channel 1 and 2 respectively. We can therefore merge the two states 'c1' and 'c2' into a single state, 'cm' (a "care" packet in the "memory"). As a result, if n is the buffer size, the total number of states in the analysis of memory transition is $(n+1)(n+2)/2$. In the following session, we provide "one buffer analysis" for all the control strategies we proposed in [1]. However, our analysis is general enough to be extended to arbitrary buffer size. By making use of the particular memory structure of OSU-I proposed by Chlamtac and Fumagalli [15], we are able to reduce the state space to only $(n+1)$ states. This greatly reduces the complexity of the analysis from $O(n^2)$ to $O(n)$, and leads to a complete and simple solution to one of the control strategies, namely CS2.

4.1 Control Strategy 1 (CS1)

The optical storage unit used in CS1 is shown in Figure 4 below. It uses 3×3 optical switches and is called OSU-II. In CS1, 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. Therefore, 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.

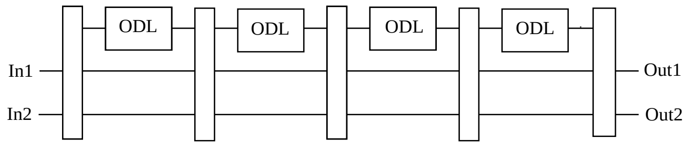

Figure 4: OSU-II with buffer size equal to 4.

We find the probability of deflection, P_{def}, for each offered load, g, by considering the deflection of our "test" packet in Figure 3. As the "test" is no different from any other packet in the network, the deflection probability found is also the network-wide deflection probability and, therefore, can be used to obtain other performance measures [11]. In the one buffer analysis of CS1 [1], deflection of the "test" packet in Figure 3 occurs when all of the following conditions are satisfied:

- There is a "care" packet in the memory. This occurs with probability P_{cm}, which is the steady state probability that the memory stores a "care" packet.
- A packet is present in the other link of the node. This occurs with probability P_{pkt} in accordance with Equation (9).
- In the routing procedure, the "test" packet is randomly chosen to be routed with the packet in the memory in the clock cycle. This occurs with probability 1/2. In this way, the packet in the other link is put into the memory buffer in the next clock cycle. Note that if the "test" packet is not chosen, it will never be deflected according to the control strategy.
- There is a collision in output channel assignment between the "test" packet and the packet residing in the memory. In this case, since the "care" packet in the memory has higher priority in output channel assignment according to the control strategy, the "test" packet is deflected. This occurs with probability 1/2.

Let the probability of deflection of our "test" packet in the network be P_{def}. According to the model assumed in Section 3.1, the four conditions mentioned above are independent. As mentioned before, P_{def} is also the probability of deflection of any "care" packet in the network. The deflection probability, P_{def}, can therefore be obtained by solving the self-constraint equation for our "test" packet:

$$P_{def} = P_{cm} P_{pkt} \frac{1}{4}. \tag{16}$$

Note that P_{pkt} given in Equation (16) depends on u and α (see Equation (9). Furthermore, u and α are related to each other through Equation (7), while α is related to $E[D]$ given in Equation (6). Finally $E[D]$ is given in Equation (14). As a result, because $E[D]$ depends only on P_{def}, Equation (16) is a nonlinear equation for P_{def}. In order to solve it for P_{def}, we have to find P_{cm}, the steady state probability that the memory is occupied by a "care" packet. We will express P_{cm} in terms of the familiar parameter u so that P_{cm} can also be expressed in terms of P_{def} through Equations (7) and (14). In this way, Equation (16) can be solved for P_{def} for each value of g.

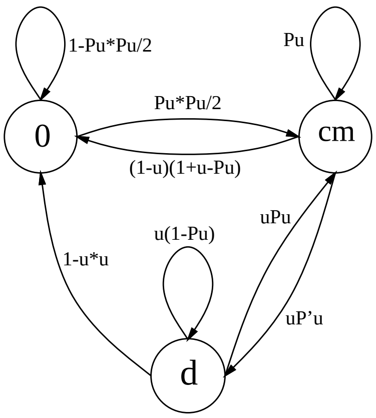

Figure 5: State transition diagram for one buffer case of CS1.

We model the transition of the memory state as a discrete time Markov chain using parameters u, P_u and $P_u^{'}$, which are given in Equations (7), (11), (13) respectively, and derive P_{cm} from the state transition diagram. The state transition diagram is shown in Figure 5. The transitions are explained as follows:

By considering the buffer stage in Figure 3, there is a transition into the 'cm'-state in a time epoch when one of the following conditions is met:

- The memory state is empty ('0'-state) and there are two "care" packets coming in and both contend for the same output link. One of them is then put into the memory. This occurs with probability $P_u^2/2$, where P_u is the the probability that there is a "care" packet in the input link of the buffer stage; or
- The memory is in the 'd'-state ("don't care" state) and
  - There are two "care" packets coming into buffer stage. This occurs with probability P_u^2; or
  - there is a "don't care" packets and a "care" packet coming in, which occurs with probability $2P_u^{'}P_u$, where $P_u^{'}$ is the probability that a "don't care" packet is found in the input link to the buffer stage. Furthermore, by the random selections of the packets in the control strategy, the "don't care" packet happens to be routed before the "care" packet. This occurs with probability 1/2.

Therefore, the total transition probability from 'd' state to 'cm' state is

$$P_{d\to cm} = P_u^2 + 2P_u^{'} P_u \frac{1}{2} \tag{17}$$

$$= uP_u, \tag{18}$$

where Equation (13) is used to get Equation (18).

A transition out of the 'cm'-state takes two different paths:

- The memory state moves from the 'cm'-state to the '0'-state. This occurs when one of the following conditions is satisfied.
  - There is only one incoming packet into the buffer stage and it is a "care" packet (occurs with probability $2P_u(1-u)$). The incoming "care" packet does not collide with the "care" packet in the memory (occurs with probability 1/2), which is definitely to be routed correctly in the clock cycle according to the FIFO algorithm. Therefore, the total probability for the above conditions is $(2P_u(1-u)) \times \frac{1}{2}$; or
  - There is only one packet coming in. The packet is "don't care". This occurs with probability $(2P_u^{'}(1-u))$; or
  - No packet is present in both link, which occurs with probability $(1-u)^2$.

  Therefore, summing all the probabilities of the three conditions together, the total probability to go from 'cm'-state to '0'-state is

$$P_{cm\to 0} \quad = \quad (1-u)(1+u-P_u). \tag{19}$$

- The memory state moves from the 'cm'-state to the 'd'-state if
  - there is a "don't care" packet and a "care" packet coming in and the "care" packet is selected to be routed in the clock cycle instead of the "don't care" packet. This occurs with probability $(2P_uP_u^{'}) \times \frac{1}{2}$, or
  - there are two "don't care" packets coming into the buffer stage, which occurs with probability $P_u^{'2}$.

  Therefore the total transition probability from 'cm' state to 'd' state is

$$\begin{aligned} P_{cm\to d} &= P_uP_u^{'} + P_u^{'2} & (20)\\ &= uP_u^{'}. & (21)\end{aligned}$$

We further note that the 'd'-state can transit to '0'-state only if at most one of the links is occupied (i.e., not both links are occupied). Hence,

$$P_{d\to 0} = 1 - u^2. \tag{22}$$

With the condition

$$P_d + P_0 + P_{cm} = 1, \tag{23}$$

P_{cm} can be obtained [18] in terms of u and P_u, which are related to P_{def} as mentioned before. Equation (16) can then be solved for P_{def} for a given offered load, g. Throughput can then be obtained by using Equations (8) and (14).

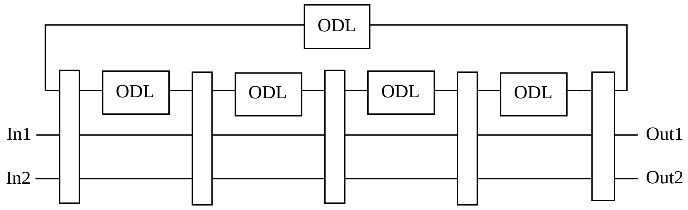

Figure 6: OSU-I with buffer size equal to 5.

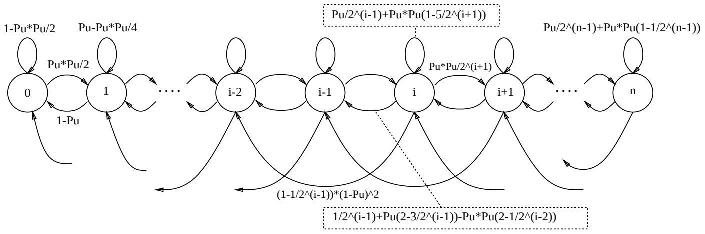

Figure 7: State transition diagram for CS2 with buffer size n.

4.2 Control Strategy 2 (CS2)

CS2 uses optical storage unit OSU-I shown in Figure 6 [1, 15]. Note that packets can be continuously circulated in the buffer in this memory construction.

In this strategy, "care" packets take priority in routing over "don't care" packets. This is called CFDL ("Care" packets First, "Don't care" packets Last)[12]. 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.

A complete analysis of the memory behavior with OSU-I can be done with only $(n+1)$ states, by making use of th particular structure of the memory. Note that in this strategy, the packets stored in the memory can circulate indefinitely in the memory and "don't care" packets are routed with the lowest priority [1]. Therefore, "don't care" packets in the memory are basically no different from an empty slot. In the analysis, we can therefore cascade the empty states (i.e., the '0'-state) and the "don't care" states (i.e., the 'd'-state) together to form a single state. The state number in the transition diagram can then be the total number of "care" packets in the memory. In other words, $(n-i)$ is the total number of "don't care" packets and empty slots in the memory, where n is the buffer size. This reduces the state space from the original $(n+1)(n+2)/2$ states to only $(n+1)$ states. The memory transition diagram for this particular type of memory for arbitrary buffer size n is shown in Figure 7.

The state transition diagram can be obtained as follows by referring to Figure 3 again: Consider state i. There will be transition to state $(i+1)$ if there are two "care" packets coming into the buffer stage and they both contend for the same output link with all the "care" packets in the memory. This occurs with probability $P_u^2 \cdot (2/2^{i+2})$.

The memory in state i will remain there if one of the following conditions are satisfied:

- there are two "care" packets coming into the buffer stage in Figure 3, and they, together with all the i "care" packets in the memory, are not destined for the same output link. In this way, two "care" packets can be routed. This occurs with probability $P_u^2(1-2/2^{i+2})$; or
- there is a "care" packet coming in. All the "care" packets in the memory and the incoming packet are destined for the same output link. In this way, only one "care" packet can be routed. This occurs with probability $(2P_u(1-P_u)) \cdot \frac{2}{2^{i+1}}$.

Hence, the total probability of staying in the i state in the next clock cycle is the sum of the above two probabilities:

$$P_{i \to i} = \frac{P_u}{2^{i-1}} + P_u^2(1 - \frac{5}{2^{i+1}}), \qquad (24)$$

for $1 \leq i \leq n-1$.

There will be a transition from state i to state $(i-2)$ only if there are no "care" packet coming into the node and all the i "care" packets in the memory do not contend for the same output channel. Therefore,

$$P_{i \to (i-2)} = (1-P_u)^2(1-1/2^{i-1}),$$

for $2 \leq i \leq n$.

Furthermore, when the buffer is full of "care" packets (i. e., when it is in the n-state), it will remain in the state in the next clock cycle if

- there are two "care" packets coming in, which occurs with probability $P_u^2/2$;
- there is a "care" packet coming in and all the "care" packets contend for the same output link, so that only one "care" packet can be routed. This occurs with probability $(2P_u(1-P_u)) \cdot \frac{2}{2^{n+1}}$.

Hence, summing the probabilities of the two conditions together,

$$P_{n \to n} = \frac{P_u}{2^{n-1}} + P_u^2(1 - \frac{1}{2^{n+1}}).$$

Let π_i be the steady-state probability for the memory to be in state i. Referring to Figure 7, with

$$\sum_{i=0}^{n} \pi_i = 1,$$

π_n can be solved.

We are now ready to consider the deflection of the "test" packet. The "test" packet in Figure 3 will be deflected when all of the following conditions are satisfied:

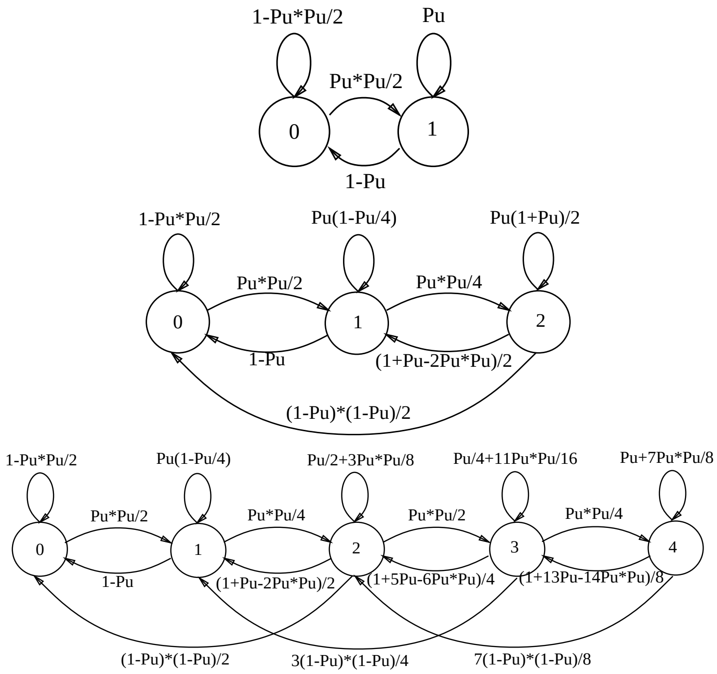

Figure 8: State transition diagrams for CS2 with buffer sizes 1, 2 and 4.

- the buffer is full of "care" packets (occurs with probability π_n);
- there is a "care" packet on the other link (occurs with probability P_c given in Equation (10));
- all the "care" packets are destined for the same output link as the "test" packet's (occurs with probability $1/2^{n+1}$); and
- the "test" packet is deflected with a coin flip (occurs with probability $1/2$).

Therefore, the self-consistent equation for the probability of deflection of our "test" packet is given by

$$P_{def} = \pi_n P_c \frac{1}{2^{n+1}} \frac{1}{2}. \qquad (25)$$

Figure 8 shows the transition diagram for buffer sizes, n, equal to 1, 2 and 4. These enable us to find π_n, the steady state probability that the memory is full of "care" packets.

After π_n is found, P_{def} can be solved from Equation (25), as P_c depends on u and P_{dc} which in turn depends on P_{def} (see Equation (10)). Throughput can then be obtained using Equation (8). The complete analysis with this strategy along with the simulation results is shown in Figure 9. We see a very close agreement of our analysis with the simulation results.

4.3 Control Strategy 3 (CS3)

CS3 uses storage architecture OSU-II as in CS1. The strategy resembles CS2 except that a "don't care" packet can now

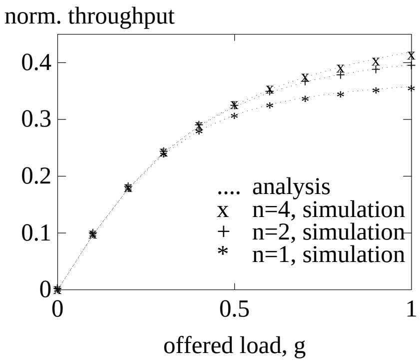

Figure 9: Analytic and simulation results for CS2 with buffer sizes 1, 2 and 4.

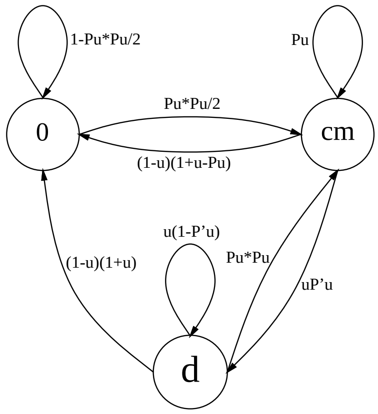

Figure 10: State transition diagram for one buffer case with CS3.

leave the memory whenever it is at the rightmost position of the memory.

As the memory structure used in this strategy is the same as that used in CS1, memory transition probability is very similar to that of CS1. However, because CFDL is used in this strategy, $P_{d \to cm}$ is slightly different from what we obtained in CS1. There is a transition from 'd'-state to 'cm'-state whenever there are two incoming "care" packets. Therefore,

$$P_{d \to cm} = P_u^2.$$

The state transition diagram is shown in Figure 10. With $P_{cm} + P_0 + P_d = 1$, P_{cm} can be solved [18].

Consider the buffer stage in Figure 3. Deflection of the "test" packet occurs when

- the buffer contains a "care" packet (occurs with probability P_{cm}),
- there is a "care" packet in the other link coming into the buffer stage (occurs with probability P_c),
- the two "care" packets are destined for the same output channels as the "test" packet's (occurs with probability 1/4) and
- the "test" packet deflects with a coin flip (occurs with probability 1/2).

Therefore, the nonlinear equation for P_{def} is

$$P_{def} = \frac{1}{8} P_{cm} P_c. \tag{26}$$

4.4 Control Strategy 4 (CS4), Hot-Potato and Store-and-Forward Routing

These cases have been considered in [16, 11, 18]. The analyses follow similar procedure above by solving a self-consistent equation in P_{def}.

Once we obtain P_{def}, network parameters such as normalized throughput, average delay, probability of don't care and hop distribution can all be obtained [11]. The analysis of the performance of shufflenet is then complete.

Figure 11 shows our analytic results on throughput together with the simulation results reported earlier [1, 5]. We see a close agreement between analysis and simulation.

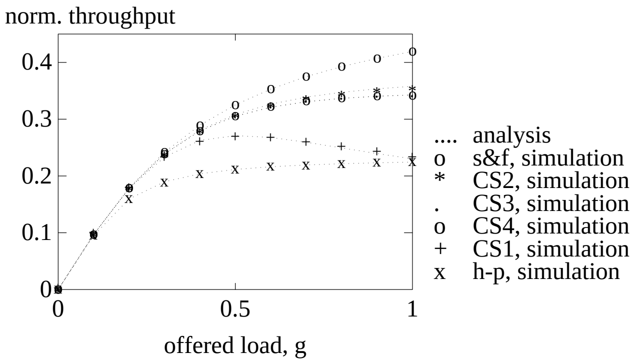

Figure 11: Analytic and simulation results for hot-potato routing, store-and-forward routing and one-buffer cases for all the control strategies proposed.

5 Priority CR Scheme

The priority CR rule we consider here give packets at their last hops to their destinations the highest priority in output channel assignment. All the other packets are given equal but lower priority. In case of contention of packets with equal priority, one of the packets are deflected with a coin flip. This CR rule is a slight variation of the static distance preference scheme discussed in [8]. From the simulation result on MSN (Manhattan Street Network), the static distance preference scheme improves the throughput of the network. It is observed that our simplified CR rule as applied to shufflenet also improves the throughput of the network.

The analysis of the distance-based priority CR rule in the shufflenet is more complicated than the distance-

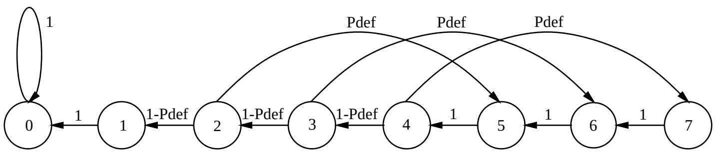

Figure 12: State transition diagram for a typical packet in the network for the priority CR rule used in the $(2,4)$ shufflenet.

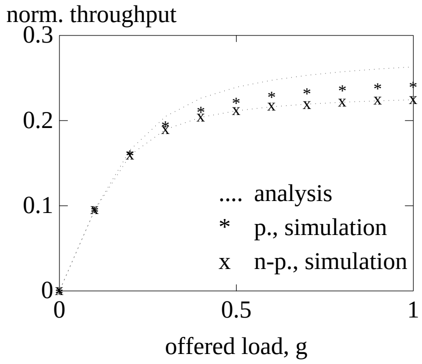

Figure 13: Analytic and simulation results for the priority (p.) and non-priority (n-p.) CR rule in the (2, 4) shufflenet with hot-potato routing.

independent priority scheme, because the deflection probability of a packet in the network depends on how far the packet is from its destination. As the probability that two packets destined for the same destination meeting at their last hop is very small, in order to obtain a reasonable estimate of the performance using the priority CR rule, we neglect the small probability of deflection in the last hop. As a result, packets are never deflected in their last hop in our model. As we will see, the analysis agrees well with the simulation for a $(2,4)$ shufflenet, indicating that the probability of deflection in the last hop can be justifiably neglected.

Figure 12 shows the state transition diagram for a typical packet in our simplified model in the (2, 4) shufflenet. The state is the distance of the packet from its destination. With $D_1 = 1$ and $N_1^{dc} = 0$, we obtain

$$D_j = \begin{cases} j + \frac{k}{(1-P_{def})^{k-1}}[1-(1-P_{def})^{j-1}], & 1 \le j \le k, \\ D_k + (j-k), & \text{for } k+1 \le j \le 2k-1, \end{cases} \quad (27)$$

and

$$N_j^{dc} = \begin{cases} D_j - 1 - \frac{1-(1-P_{def})^{j-1}}{P_{def}(1-P_{def})^{k-1}}, & \text{for } 1 \le j \le k, \\ N_k^{dc} + (j-k), & \text{for } k+1 \le j \le 2k-1, \end{cases} \quad (28)$$

where D_i and N_i^{dc} are defined in Section 3.2.

Using Equations (3) and (4), the probability of "don't care", P_{dc}, can then be obtained with Equation (5).

As it has been shown that priority CR rule has hardly any effect on buffered network [8], we apply the priority CR scheme to a $(2,k)$ shufflenet in the hot-potato case. Consider a "test" packet in the network as in Figure 3. The probability that there will be a "care" packet in the other link is given by P_c as in Equation (10). The "test" packet will be deflected if it is more than one hops away from its destination and the following conditions are satisfied:

- The "care" packet in the other link is one hop away from its destination (occurs with probability $2/(\sum_{i=1}^{k} 2^i - 1)$) and the "test" packet contends for the same output link with the packet (occurs with probability 1/2). In this case, the "test packet" is deflected definitely. Therefore, the condition occurs with probability

$$\frac{2}{\sum_{i=1}^{k} 2^i - 1} \cdot \frac{1}{2} = \frac{1}{2^{k+1}-3};$$

- The other packet is more than one hop from its destination (occurs with probability $1 - 2/(\sum_{i=1}^{k} 2^i - 1)$), but it contends with the "test" packet for the output channel (occurs with probability 1/2). The test packet loses with a coin flip (with probability 1/2). Therefore the total probability is

$$(1 - \frac{2}{\sum_{i=1}^{k} 2^i - 1}) \cdot \frac{1}{4} = \frac{2^{k+1}-5}{4(2^{k+1}-3)}.$$

Hence, the self-consistent equation for P_{def} is given by:

$$P_{def} = P_c \left[\frac{2^{k+1}-1}{4(2^{k+1}-3)} \right]. \quad (29)$$

After P_{def} is found, D_i and N_i^{dc} can then be found from Equations (27) and (28). Using Equations (3) and (4), P_{dc} can then be obtained with Equation (5). Figure 13 shows the analytical results (after solving Equation (29) with Equations (7) and (8)) with the simulation results for the (2,4) shufflenet. Our approximation is substantial. The over estimation in the throughput is mainly due to the neglect of the packets in their last hops to their destinations. There is about 10% increase in throughput in hot-potato routing compared with the case when non-priority scheme is used.

6 Conclusion

We have presented a Markov chain model for the analysis of a shufflenet. The analysis of shufflenet is complete, once the probability of deflection, P_{def}, is obtained. Performance measures of interest in a shufflenet (e.g., hop distribution, average delay and throughput) are obtained solely through this parameter [11].

We have shown the analysis of one buffer cases for all the control strategies we proposed in [1]. Our analysis can be easily generalized to arbitrary buffer size. The state space for the memory transition in a shufflenet with two packet classes (i.e., "care" and "don't care" packet classes) generally increases with buffer size, n, according to $O(n^2)$; however, by making use of the particular memory structure of OSU-I [1, 12], we can reduce the state space tremendously to $O(n)$.

As a result, the analysis is substantially simplified. Therefore, a complete analysis using optical storage unit OSU-II is possible with less effort and time. Our analysis shows excellent agreement with simulation reported earlier [1].

We have also provided an analysis of a priority CR scheme. The priority scheme is applied to hot-potato routing in shufflenet. By minimizing the deflection probability of a packet in its last hop to the destination, the throughput of the network is increased. The increase in throughput in a (2, 4) shufflenet can be as high as 10% in the hot-potato case. However, the tail of the hops distribution is expected to be broadband in this case. We remark that although the complexity of the network does not increase greatly with this priority scheme, the deterioration in hop distribution has to be considered when the scheme is to be used. We have also done analysis for the priority scheme. Our analysis shows close agreement (within 9%) with the simulation result.

References

[1] S-H. Chan and H. Kobayashi, "Buffer Architectures and Routing Algorithms in the Performance of Shufflenet", *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.

[2] P. Henry, "High-Capacity Lightwave Local Area Networks," *IEEE Communications Magazine*, pp. 20-26, Oct. 1989.

[3] P. Prucnal, "Optically-processed self-routing, synchronization and contention resolution for 1D and 2D photonic switching architectures," *Proceedings in IEEE Topical Meeting on Multiaccess Network*, Aug., 1992.

[4] P. Prucnal and P. Perrier, "Optically-Processed Routing for Fast Packet Switching," *IEEE LCS Magazine*, vol. 1, no. 2, pp. 54-67, May 1990.

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

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

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

[8] T. Robertazzi and A. Lazar, "Deflection Strategies for the Manhattan Street Network," *ICC'91*, pp. 1652-1658, 1991.

[9] A. Choudhury and V. Li, "Effect of Contention Resolution Rules on the Performance of Deflection Routing," in *Proc. IEEE GLOBECOM '91 Conf.*, pp. 1706-1711, 1991.

[10] Z. Zhang and A. Acampora, "Performance Analysis of Multi-hop Lightwave Network with Hot Potato Routing and Distance-Age-Priority", *IEEE INFOCOM'91*, pp.1012-1021.

[11] S-H. Chan and H. Kobayashi, "Performance Analysis of ShuffleNet with Deflection Routing," IEEE GLOBECOM'93, Vol. 2, pp. 854-859.

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

[13] M. Hluchyi and M. Karol, "Shufflenet: An Application of Generalized Perfect Shuffles to Multihop Lightwave Networks", *Journal of Lightwave Technology*, Vol.9, No.10, Oct. 1991, pp.1386-1397.

[14] 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.

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

[16] 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.*

[17] 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.

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

[19] A. Greenberg and J. Goodman, "Sharp Approximate Models of Deflection Routing in Mesh Networks," *Proc. of International Sem. on Teletraffic Analysis and Computer Performance Evaluation*, The Neverland, June 1986.

[20] A. Krishna and B. Hajek,"Performance of Shuffle-Like Switching Networks with Deflection," in *Proc. IEEE INFOCOM'90*, pp. 473-480, 1990.

[21] A. Choudhury and V. Li, "Performance Analysis of Deflection Routing in the Manhattan Street Network," *IEEE ICC'91*, pp. 1659-1665, 1991.