Asymptotic Performance of a Buffered Shufflenet with Deflection Routing[*][†]

Shueng-Han Gary Chan
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
and
Hisashi Kobayashi
Department of Electrical Engineering
Princeton University
Princeton, NJ 08544

Abstract

Throughput of a shufflenet with deflection routing under high load and low load is obtained as a function of the network and buffer sizes. We give general routing conditions which achieve high performance in a shufflenet. Using a routing algorithm similar to the algorithm we consider here, throughput of a shufflenet with only one buffer can be increased by more than 45% compared with the shufflenet without any buffer, the so-called hot-potato case. The increase is general for the shufflenet of size ranging from as few as 24 nodes to more than 10,000 nodes. The increase is more significant when the network becomes larger. We note that a large number of routing algorithms currently proposed to be used in the shufflenet satisfy the general routing conditions mentioned here. Using the routing algorithm we mention here, a shufflenet with only two buffers can achieve performance comparable to the store-and-forward case. In previous studies of the shufflenet, the derivation of the important parameter – the probability of deflection of a packet in the network – is usually complicated. We have obtained a simple approximation of this parameter, which greatly simplifies the analysis of a shufflenet of any size and with any number of buffers. This enables us to conclude that the performance of a shufflenet scales well with different network and buffer sizes if the routing algorithm is chosen properly. We finally verify our results with the simulations that have been done.

[*]This research has been supported, in part, by a grant from the Ogasawara Foundation for the Promotion of Science and Technology; by the Innovative Partnership Program of the New Jersey Commission on Science and Technology; and by the NCIPT (National Center for Integrated Photonic Technology) through its ARPA grant.
[†]To appear in *Proceedings of IEEE GLOBECOM'94*

1 Introduction

A shufflenet is a high speed multiconnected optical network. By making use of the vast bandwidth of fiber and allowing different users to simultaneously transmit information through the network, a shufflenet can achieve high throughput despite the current constraints in optical technology [1]-[3].

The performance of a shufflenet has been both simulated and analyzed in recent years [4]-[8]. In the simulation and analysis, a certain network size and a specific routing algorithm (e.g., FIFO or "Care Packets First, Don't care packets Last" – the so-called CFDL routing algorithm [9]) are usually assumed. In the case of finite buffering, a certain type of optical switches (e.g., 2×2 or 3×3 switches) and buffer architectures are usually assumed [5, 8, 9]. Therefore, every time a shufflenet undergoes changes in size due to splitting or expansion, or buffer structure changes due to the advance in technology, or a new routing algorithm is proposed, the same simulations and analysis have to be done all over again to obtain the network performance. This is inconvenient and time consuming. Furthermore, it is hard to draw a general conclusion on the trend of performance and network scalability for different network and buffer sizes.

In this paper, we consider a routing algorithm satisfying some very general routing conditions. The performance of a shufflenet as a function of network size and buffer capacity is obtained without any particular specification on the buffer structure. In previous studies [5]-[7], it has been observed that a (2,4) shufflenet with only two buffers can achieve performance comparable to the store-and-forward case. However, it was not clear whether this observation is still valid for any size of shufflenet. We present in this paper that, using a routing algorithm similar to the algorithm we consider here, any $(2, k)$ shufflenet can achieve such high performance for a wide range of parameter k, ranging from as few as $k = 3$ (i.e., 24 nodes) to $k = 10$ (i.e., more than 10,000 nodes in the network). Moreover, a shufflenet with only one buffer can achieve more than 45% increase in performance compared with the hot-potato case. The improvement in performance becomes even more significant as the network size increases.

We first briefly discuss the shufflenet and review the previous analysis, focusing mainly on the $(2, k)$ shufflenet. We will then consider a routing algorithm that achieves high throughput in shufflenet. We note that a number of routing algorithms proposed to be used in the shufflenet are very similar to the algorithm we consider here. In the analysis of a shufflenet with deflection routing, the probability of deflection for a packet in the network, P_{def}, is an important parameter to characterize the network performance [7]. We obtain a simple approximate expression for the parameter, which greatly simplifies the analysis and allows us to conclude that the performance of a shufflenet scales well with different network and buffer sizes using the algorithm we consider here. Once P_{def} is obtained, other network performance measures such as the hop distribution and the average delay can then be obtained without difficulties [7].

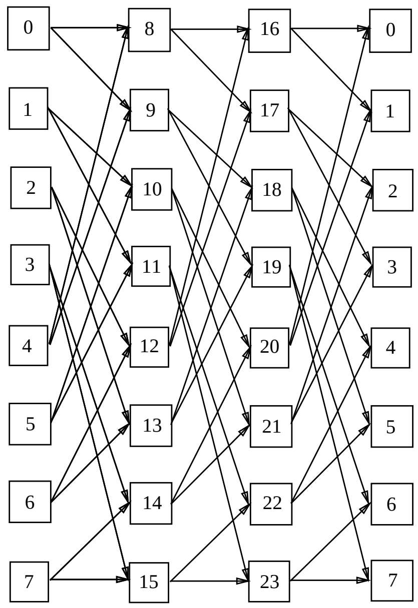

Figure 1: A (2,3) shufflenet

2 Shufflenet and its Analysis

A shufflenet is a multi-hop network [1]. Each user in the shufflenet accesses the network through the Network Interface Unit (NIU). Each NIU has a number of lightwave receivers and transmitters. A shufflenet is characterized by two numbers, p and k. A $(p,\ k)$ shufflenet consists of kp^k nodes arranged in k columns, with each column consisting of p^k NIUs. Figure 1 shows a (p, k) shufflenet with $p = 2$ and $k = 3$. In a shufflenet, all the NIUs are interconnected like a perfect shuffle, with the last column being "wrapped-around" to the first column to form a completed cylinder. In this way, packets can be continuously circulated around the network until they reach their destinations.

Packets are transmitted within the shufflenet in a store-and-forward fashion, as long as there is buffer available in the NIUs. A packet hops through the nodes until it reaches its destination, where the packet will be absorbed. Different packets destined to different destinations may suffer collisions with each other during the process of routing. In a shufflenet with deflection routing, one of the colliding packets will be routed correctly while the rest of them will be either stored if storage is available, or "deflected" temporarily to wrong channels. Therefore, with deflection routing, packets are never lost due to buffer overflow.

In a (p, k) shufflenet, a packet at a node is said to be "don't care" with respect to its destination if the destination is not reachable within k hops by the packet. A node is said to be "don't care" for a packet if the packet at this node can reach its destination with the minimum number of hops by taking any link emanating from this node. Therefore, if a packet is at its "don't care" node, it will never suffer deflection.

Shufflenet with deflection routing under uniform traffic condition and non-prioritized contention resolution rules has been analyzed in [7]. Let $E[D]$ be the average number of hops for a packet in the (p, k) shufflenet to reach its destination and α be the probability that a packet is absorbed in a node. Then, obviously,

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

and we find,

$$\begin{aligned} E[D] &= -\frac{kp(1-p^{k-1}(1-P_{def})^{k-1})}{(kp^k-1)(1-P_{def})^{k-1}(1-p+pP_{def})} \\ &\quad + \frac{kp^k}{kp^k-1}\left(\frac{k-1}{2}\right) + \frac{k}{(1-P_{def})^k}. \end{aligned} \tag{2}$$

Let $E[N^{dc}]$ be the expected number of "don't care" nodes that a packet visits before it reaches its destination, and P_{dc} be the probability that a packet visits one of its "don't care" nodes in each hop on the way to its destination. Then we have [7],

$$\begin{aligned} E[N^{dc}] &= E[D] - \frac{p}{P_{def}(kp^k-1)}\left[\frac{p^{k-1}-1}{p-1} - \right. \\ &\quad \left. \frac{1-p^{k-1}(1-P_{def})^{k-1}}{(1-P_{def})^{k-1}(1-p+pP_{def})}\right] - \\ &\quad \frac{1-(1-P_{def})^k}{(1-P_{def})^k P_{def}}, \end{aligned} \tag{3}$$

and

$$P_{dc} \triangleq \frac{E[N^{dc}]}{E[D]}, \tag{4}$$

$$= 1 - P_{cr}, \tag{5}$$

where P_{cr} is the probability that a packet visits one of its "care" nodes in each hop on its way to its destination.

In this paper, we will focus on $(2, k)$ shufflenet. Let u be the probability that a given time-slot in a link of the network is occupied by a packet. Denote g as the offered load, which is defined as the probability of generation of a new packet in a node in each clock cycle. u can then be expressed as [5]:

$$u = \frac{\sqrt{\alpha^2 + g^2(1-\alpha)^2} - \alpha}{g(1-\alpha)^2}. \quad (6)$$

The probability of deflection for a packet in a $(2, k)$ shufflenet can then be obtained by solving the following equation for P_{def}:

$$P_{def} = P_c \cdot \frac{1}{n_{bs}}, \quad (7)$$

where P_c is the probability that a packet encounters a "care" packet in the routing process in a node, and is given by [5],

$$P_c = [u(1-\alpha) + u\alpha g + (1-u)g](1 - P_{dc}), \quad (8)$$

and n_{bs} is the average "free" time before a "tagged" "care" packet in the network is deflected again since the last deflection of the packet, given that it is routed with another "care" packet in the routing process. In other words, $1/n_{bs}$ in Equation (7) is the probability of deflection of a "care" packet given that it encounters another "care" packet in the routing process. Therefore, the larger the value of n_{bs} is, the less likely a packet will be deflected in the network. In the analysis of shufflenet, n_{bs} is a parameter which depends on the buffer size, bs, and the access algorithm of a packet in the memory. Its value depends on the occupancy probability of the memory and contention resolution rules [5, 8]. In a shufflenet, we have the cases $n_0 = 4$ for hot-potato routing and $n_\infty = \infty$ for store-and-forward routing. As hot-potato routing attains the minimum throughput performance in a shufflenet, any other routing algorithms have n_{bs} no less than 4.

It should be noted that from Equation (8) and under high load (i.e., $g \simeq 1$), the probability that a "care" packet encounters another "care" packet in the routing process, P_c, and the probability that a packet visits one of its "care" nodes in its way to the destination, P_{cr}, are related by

$$P_c \simeq P_{cr}. \quad (9)$$

This approximation will be used in the asymptotic analysis of shufflenet.

We see from Equation (8) that P_c depends on both u and P_{dc}, which in turns depend on $E[N^{dc}]$ and $E[D]$. As both $E[D]$ and $E[N^{dc}]$ are functions of P_{def} through Equations (2) and (3), Equation (7) is a nonlinear equation in P_{def}, the probability of deflection of a packet in a $(2, k)$ shufflenet.

The solution of Equation (7) gives P_{def}. With the knowledge of P_{def}, the normalized throughput, λ, defined as the number of packet generated (or absorbed) in a node in each clock cycle in the steady state, can be obtained as [5]

$$\begin{aligned} \lambda &= 2\alpha u & (10) \\ &= 2\alpha \frac{\sqrt{\alpha^2 + g^2(1-\alpha)^2} - \alpha}{g(1-\alpha)^2}. & (11) \end{aligned}$$

We note from Equation (11) that under high load (i.e., $g \to 1$) and $\alpha \ll 1$,

$$\lambda \approx 2\alpha. \quad (12)$$

3 Buffered Shufflenet under Low Load

For a $(2, k)$ shufflenet under low load, i.e., $g \to 0$, the probability of finding a packet in a time-slot is very small, yielding $u \simeq 0$. Equation (8) therefore gives

$$P_c \simeq 0. \quad (13)$$

As a result, from Equation (7), $P_{def} \simeq 0$. Therefore, using $E[D]$ in Equation (2) and $p = 2$, we have

$$\begin{aligned} \lim_{P_{def} \to 0} E[D] &= \frac{k\,(2 - 3 \cdot 2^k + 3 \cdot 2^k\,k)}{2\,(-1 + 2^k\,k)}, & (14) \\ &\approx \tfrac{3(k-1)}{2}, \quad \text{for } k \geq 3. & (15) \end{aligned}$$

With $g^2((1-\alpha)/\alpha)^2/2 \ll 1$ (which gives $g \ll 1/(k-1)$), expansion of λ in Equation (11) around $g \simeq 0$ gives

$$\lambda \approx g. \quad (16)$$

The result simply says that under low load (i.e., $g \ll 1/(k-1)$), the throughput of a shufflenet is directly proportional to the offered load, i.e., the packet generation rate. The initial increase in throughput with traffic load, g, is independent of the buffer size and routing algorithm. This verifies the linearity previously observed in the simulations of a (2,4) shufflenet with different buffer sizes and routing algorithms reported previously [5, 6].

4 Store-and-Forward Shufflenet under High Load

For the $(2, k)$ shufflenet using store-and-forward routing algorithm, we have $P_{def} \equiv 0$. Figure 2 shows the normalized throughput using Equation (11) under the high load condition ($g \simeq 1$) and $\alpha = 1/E[D]$, where $E[D]$ is given in Equation (14).

For a large shufflenet, Equation (15) gives $\alpha \approx 2/(3(k-1))$. With $\alpha \ll 1$, the normalized throughput, λ, of Equation (11) can be expressed as

$$\begin{aligned} \lambda &\approx 2\alpha & (17) \\ &\approx \frac{4}{3(k-1)} & (18) \\ &\sim \frac{4}{3\log_2 N}, & (19) \end{aligned}$$

where N is the network size. The inverse relationship in the store-and-forward throughput of a shufflenet against the network size is clearly observed in Figure 2.

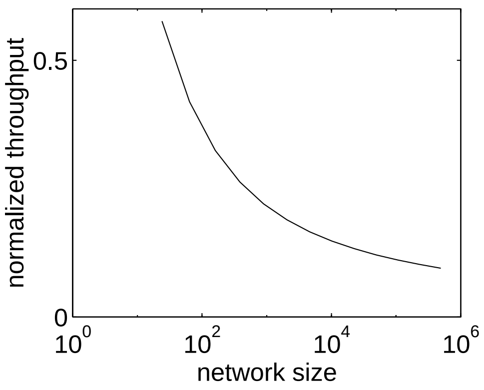

Figure 2: Normalized store-and-forward throughput of a shufflenet verses the network size, N. Throughput decreases as $\sim 1/\log_2 N$, for large N.

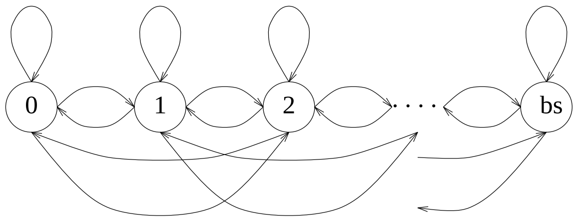

Figure 3: Memory transition diagram in each node for a $(2, k)$ shufflenet with buffer size bs.

5 Shufflenet with Finite Buffer Size under High Load

5.1 Routing Conditions

Throughput of a shufflenet whose size ranges from as few as 24 nodes (corresponding to $k = 3$) to as many as 10,000 nodes (corresponding to $k = 10$) will be analyzed under very general routing algorithm and buffer structures. In the analysis of the buffered shufflenet, the occupation probability of the memory is usually found [5, 8] from the Markov chain model. Let bs be the buffer size in a $(2, k)$ shufflenet and the state be the number of "care" packets in the memory, then the memory transition for each node in a $(2, k)$ shufflenet can be drawn as in Figure 3.

The transition diagram is obtained as follows:

- In any cycle, at most two packets can be routed in each node. Hence, the buffer state can decrement by at most two. This occurs when two "care" packets are routed out of the memory into the network and no new "care" packet is put into the memory in the cycle;
- In any cycle, at most two "care" packets can be put into the memory because there are only two incoming links at a node in the $(2, k)$ shufflenet. Therefore, the state number increases by two, when the two incoming "care" packets are put into the memory, and no "care" packet is injected into the network.

Let π_i, $0 \le i \le bs$, be the steady state probability that the memory is at state i in Figure 3. Clearly,

$$\sum_{i=0}^{bs} \pi_i = 1. \qquad (20)$$

It should be noted that $P_{i,j}$, the transition probability from state i to state j, depends on the routing algorithm and the buffer structures [5, 6, 8].

We consider a high performance routing algorithm satisfying the following routing conditions:

"Care packets First, Don't care packet Last," or simply CFDL, has been shown to be an effective routing method to achieve high throughput performance [9, 6]. In CFDL, "care" packets in the memory always have higher priority in routing than "don't care" packets in the node, provided that no deflection is incurred. Using this routing method, at least one "care" packet is sent out of the buffer in each cycle. This gives $P_{i,i+2} = 0$, for $i \ge 0$.

As the number of "care" packets in the memory increases, there are more choices of the "care" packets to be routed in each clock cycle. Therefore, it becomes more likely for the "care" packets to be routed in each clock cycle. In other words, it would be less likely for the number of "care" packets in the memory to grow in the higher state. We consider in our algorithm that the steady state memory occupancy probability satisfies

$$\pi_i \ge \pi_{i+1}. \qquad (21)$$

The probability that the memory is full of "care" packets, π_{bs}, can then be expressed as,

$$0 \le \pi_{bs} \le \frac{1}{bs+1}. \qquad (22)$$

Consider the access of packets in the memory. To maximize the number of packets routed in each cycle without deflection, all the "care" packet in the memory should be screened before the output channel assignment is done. In our high-performance routing algorithm, we consider the case that every packet will be compared for output channel assignment. Note that strictly FIFO algorithm, which routes the packets without any distinction between "care" and "don't care" packet classes, does not satisfy the above condition [6].

Equation (7) is a self-consistent equation in P_{def} which can be written as follows. Consider a typical packet in the network as in Figure 4, and let it be our "tagged" packet. Clearly, the "tagged" packet will be deflected only if it is a "care" packet in the node. Therefore, deflection of the "tagged" packet in a node occurs when the following conditions are satisfied:

- There is another "care" packet on the other link (which occurs with probability P_c given in Equation (8));

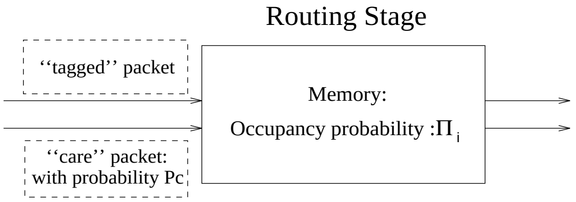

Figure 4: Figure to obtain the equation in P_{def} in the analysis of shufflenet.

bs	0(hot-potato)	1(one-buffer)	2	3	4
n_{bs}	4	16	48	128	320

Table 1: Values of n_{bs} as a function of buffer sizes, bs, for the routing algorithm being considered in the paper.

- The memory is full of "care" packets (which occurs with probability π_{bs});
- All the $(bs+2)$ "care" packets in the node contend for the same output channel and the "tagged" packet loses with a coin flip. This occurs with probability $1/2^{bs+2}$.

Therefore, a self-consistent equation for the deflection probability of the "tagged" packet in its "care" node, P_{def}, can be expressed as

$$P_{def} = P_c \pi_{bs} \frac{1}{2^{bs+2}}. \qquad (23)$$

Comparing Equations (7), (22) and (23), we have

$$n_{bs} \geq (bs+1)2^{bs+2}. \qquad (24)$$

We will use the relationship $n_{bs} = (bs+1)2^{bs+2}$ for the analysis of the routing algorithm. Table 1 shows the values of n_{bs} for different buffer sizes. It should be noted that although particular values of n_{bs} are considered, our result is general enough to apply to any real value of n_{bs}.

In reality, depending on the buffer structures and routing algorithms, CFDL and examination of every packet may not be fully implemented before a routing decision has to be made [6]. This is the case if 2×2 optical switches, non-circulating buffer architecture or strictly FIFO queuing discipline are used. However, so long as a routing algorithm is implemented similar to the routing algorithm we consider here, i.e., making an effort to implement CFDL and examine all the packets in the memory before a routing decision is made. The values of n_{bs} in Table 1 approximate the performance of a shufflenet under different buffer structures and routing schemes [6]. This shows that a large number of routing algorithms resemble the algorithm and would have similar performance we consider here.

5.2 Probability of Deflection

We obtain the first-order approximation of P_{def} by solving Equation (7) and verify its accuracy later.

Under high traffic load (i.e., $g \simeq 1$), Equations (5) and (8) give

$$\lim_{g \to 1} P_c = 1 - P_{dc} \quad \text{(Equation (8))} \qquad (25)$$

$$= P_{cr} \quad \text{(Equation (5))}. \qquad (26)$$

Let $P^{(1)}_{def,bs}$ be the first-order approximation of the actual value of P_{def} with buffer size equal to bs. The approximate value of P_{def} can be obtained by assuming that P_{def} is small. We will justify the accuracy of the approximation later. From Equation (7), using Equation (26), we therefore have

$$P_{cr} = P_c = n_{bs} P^{(1)}_{def,bs}. \qquad (27)$$

Furthermore, from Equation (5) and given that P_{def} is small, we can write

$$P_{cr} \approx A + B P^{(1)}_{def,bs}, \qquad (28)$$

where

$$A = \frac{2\left(2 - 2^{k+1} + k + 2^k k^2\right)}{k\left(2 - 3 \cdot 2^k + 3 \cdot 2^k k\right)} \qquad (29)$$

$$\approx \frac{2k}{3(k-1)}, \quad \text{for } k \geq 3; \qquad (30)$$

$$B = \frac{b_1}{b_2} \qquad (31)$$

$$\approx \frac{-24 + 13k - k^3}{9(k-1)^2}, \quad \text{for } k \geq 3 \qquad (32)$$

$$\sim -k/9, \qquad (33)$$

and

$$b_1 = -8 \cdot 2^k + 8 \cdot 2^{2k} - 6k + 25 \cdot 2^k k - 24 \cdot 2^{2k} k - 2k^2 - 2 \cdot 2^k k^2 + 13 \cdot 2^{2k} k^2 - 3 \cdot 2^k k^3 - 2^{2k} k^4; \qquad (34)$$

$$b_2 = k\left(2 - 3 \cdot 2^k + 3 \cdot 2^k k\right)^2. \qquad (35)$$

Equations (27) and (28) give the following approximate expression for the deflection probability:

$$P^{(1)}_{def,bs} = \frac{A}{n_{bs} - B} \qquad (36)$$

$$\sim \frac{2/3}{n_{bs} + k/9}. \qquad (37)$$

From Equations (2), (3) and (5), a necessary condition for the expansion of Equation (28) to be accurate is $kP^{(1)}_{def,bs} \ll 1$. Using Equation (37), we therefore need $n_{bs} \gg 5k/9$. As we are considering a $(2,k)$ shufflenet with $k \leq 10$, we need $n_{bs} \gg 4$. With the use of Table 1, $P^{(1)}_{def,bs}$ given in Equation (36) is sufficiently accurate for $bs \geq 2$. The accuracy of $P^{(1)}_{def,bs}$ for $bs = 0$ (i.e., the hot-potato case which has $n_0 = 4$) and $bs = 1$ (i.e., the one-buffer case with $n_1 = 16$ in the algorithm) needs further justification. However, as shown in the Appendix, $P^{(1)}_{def,bs}$ given by Equation (36) is also a sufficiently good approximation for these cases. $P^{(1)}_{def,bs}$ given by Equations (29), (31) and (36) is therefore a good approximate solution of Equation (7) for any buffer size bs in any $(2,k)$ shufflenet under high load.

k	N	$P^{(1)}_{def,0}$(Eq.(36))	λ_0	λ_0/λ_∞
3	24	0.1901	0.3476	0.6034
4	64	0.1777	0.227	0.5414
5	160	0.1681	0.1588	0.4897
6	384	0.1604	0.1166	0.4436
7	896	0.1539	0.0886	0.4022
8	2048	0.1483	0.0692	0.3653
9	4608	0.1434	0.0552	0.3324
10	10240	0.1389	0.0448	0.3033

Table 2: Normalized throughput of hot-potato routing ($n_0 = 4$) in a $(2, k)$ shufflenet with different network sizes, N, for the routing algorithm we consider here. λ_∞ is the store-and-forward throughput of the shufflenet.

k	N	$P^{(1)}_{def,1}$(Eq.(36))	λ_1	λ_1/λ_∞
3	24	0.05	0.5418	0.9406
4	64	0.0479	0.3639	0.8679
5	160	0.0464	0.2685	0.8278
6	384	0.0453	0.2100	0.7992
7	896	0.0443	0.1708	0.7753
8	2048	0.0435	0.1427	0.7537
9	4608	0.0428	0.1218	0.7335
10	10240	0.0422	0.1055	0.7143

Table 3: Normalized throughput of a shufflenet with one buffer ($n_1 \simeq 16$) for the type of routing algorithm we consider. The throughput of a shufflenet with only one buffer is more than 70% of the store-and-forward case.

5.3 Hot-Potato and One-Buffer Cases

Let $E_{bs}[D]$ be the expected number of hops (expected delay) for a packet in a shufflenet with buffer size bs using the routing algorithm under high load. It is shown in the Appendix that, for hot-potato routing,

$$E_0[D] \simeq \frac{k}{(1 - P^{(1)}_{def,0})^k}, \tag{38}$$

and for the one-buffer case,

$$E_1[D] \simeq \frac{\gamma_k k}{(1 - P^{(1)}_{def,1})^k}, \tag{39}$$

where γ_k is given by

$$\gamma_k = 1 + \frac{k-3}{2k} e^{-k/24}. \tag{40}$$

Let λ_{bs} be the normalized throughput of a $(2, k)$ shufflenet with buffer size bs using the approximate value of $P^{(1)}_{def,bs}$ given in Equation (36). The normalized throughput is given by Equations (1) and (11) with $g \simeq 1$. Tables 2 and 3 show the calculated throughput for the routing algorithm (i.e., using $n_{bs} = (bs + 1)2^{bs+2}$). λ_∞ is the normalized throughput for store-and-forward routing, the highest throughput performance achievable in a shufflenet (i.e., $P_{def} = 0$). The fractions λ_0/λ_∞ and λ_1/λ_∞ therefore indicate how well a shufflenet performs with respect to the highest achievable throughput with buffer sizes 0 and 1 respectively. A simulation study for the (2,4) shufflenet shows that $\lambda_0 = 0.2245$, and λ_1 ranges from 0.34 to 0.36, depending on the particular routing algorithms [6]. The close agreement between simulation and our approximation confirms that most of the routing algorithms proposed for use in the shufflenet resemble the high-performance routing algorithm we consider here.

From Tables 2 and 3, we see that hot-potato routing achieves good performance only for small to median size shufflenet. As network size increases, provision of only one buffer in a shufflenet can increase the throughput greatly. In fact, the improvement in throughput with only one buffer over the hot-potato case can be very impressive. This is generally true for the wide range of network sizes considered. Actually, using Equations (12), (38) and (39), we can express the ratio between the throughput of the hot-potato and one-buffered cases as

$$\frac{\lambda_0}{\lambda_1} \approx \frac{\gamma_k (1 - P^{(1)}_{def,0})^k}{(1 - P^{(1)}_{def,1})^k} \tag{41}$$

$$\approx \gamma_k e^{-k/8} \tag{42}$$

Hence, with a routing algorithm under similar routing conditions we consider here, a shufflenet with only one-buffer can enjoy an increase in throughput of more than 45% compared with that of the hot-potato case. The improvement is even more substantial with the increase in network size. The throughput of a shufflenet with only one buffer is more than 70% of the store-and-forward case.

5.4 More Than One Buffer Cases

The condition $kP^{(1)}_{def,bs} \ll 1$ is satisfied when more than one buffer is used. Therefore, we can calculate the normalized throughput by first-order expansion of Equation (11) in P_{def} to give,

$$\lambda_{bs} \approx \lambda_\infty - \beta P^{(1)}_{def,bs}, \tag{43}$$

where β is a constant which can be estimated as follows. A first-order expansion of $E_{bs}[D]$ in terms of deflection probability gives

$$E_{bs}[D] \approx a + bP^{(1)}_{def,bs}, \tag{44}$$

where

$$a = \frac{k\,(2 - 3\cdot 2^k + 3\cdot 2^k\,k)}{2\,(-1 + 2^k\,k)} \tag{45}$$

$$\approx \tfrac{3(k-1)}{2}, \quad \text{for } k \geq 3; \tag{46}$$

$$b = \frac{k\,(2 - 2\cdot 2^k + k + 2^k\,k^2)}{-1 + 2^k\,k} \tag{47}$$

$$\approx \quad k^2, \text{ for } k \geq 3. \tag{48}$$

Therefore, under high load (i.e., $g \simeq 1$),

$$\lambda_{bs} \approx 2\alpha \quad \text{(from Equation (12))} \tag{49}$$

$$\approx 2/(a + bP^{(1)}_{def,bs}) \tag{50}$$

$$\approx \lambda_\infty - \frac{2b}{a^2}P^{(1)}_{def,bs}. \tag{51}$$

Comparing Equations (43) and (51) and using Equations (46) and (48), we have,

$$\beta \approx \frac{8}{9}\left(\frac{k}{k-1}\right)^2. \tag{52}$$

As $n_{bs} \gg k/9$ with the network size of interest (i.e., $k \leq 10$), Equation (37) gives $P^{(1)}_{def,bs} \approx (2/3)/n_{bs}$. The throughput can therefore be expressed as (with $bs \geq 2$):

$$\lambda_{bs} \approx \lambda_\infty - \frac{16k^2}{27(k-1)^2 n_{bs}}. \tag{53}$$

From Equation (18), we have $\lambda_\infty \simeq 4/(3(k-1))$. Therefore, for $bs \geq 2$,

$$\frac{\lambda_{bs}}{\lambda_\infty} \approx 1 - \frac{4k^2}{9(k-1)n_{bs}} \tag{54}$$

$$\geq 0.897. \tag{55}$$

The last inequality is obtained by observing that $k^2/((k-1)n_{bs})$ is an increasing function in k and a decreasing function in n_{bs} (and hence the lowest bound is achieved by substituting $k = 10$ and $n_2 = 48$ using the algorithm being considered here). Previous simulations for the (2, 4) shufflenet [6] also yielded the values $\lambda_{bs}/\lambda_\infty$ of 0.9-0.94 for $bs \geq 2$, which verifies the result here. Therefore, using a routing algorithm similar to the one we consider in this section, a shufflenet with only two buffers can achieve throughput comparable with the store-and-forward case. This impressive performance in throughput holds for any shufflenet with sizes ranging from 24-nodes to more than 10,000 nodes.

6 Conclusion

The performance of a $(2, k)$ shufflenet under high load and low load has been obtained. We have obtained the throughput of a shufflenet as a function of network size and buffer capacity. Network sizes ranging from as few as 24 nodes to as many as 10,240 nodes have been investigated. We have considered general routing conditions under which high throughput performance in a shufflenet can be achieved. Using a routing algorithm similar to the algorithm we consider here, throughput of a shufflenet with only one buffer can be increased by more than 45% over the hot-potato case. The improvement in throughput is generally observed for the wide range of network sizes we examined. Furthermore, the improvement is more significant with the increase in network size. We also quantified that a shufflenet with only two buffers can achieve more than 90% of the throughput compared with the store-and-forward routing algorithm. It can therefore be concluded that in a shufflenet with deflection routing, the cost of providing more than two buffers may not be justified. The analysis is further verified with our previous simulation results.

We have also presented an approximation method which greatly simplifies the analysis of a shufflenet. A simple closed form expression for the important parameter, P_{def} (the probability of deflection of a "care" packet in one of its "care" nodes in the network), is obtained. With this parameter, various network performance measures such as throughput, average delay and hop distribution can then be derived [7]. With the simple approximation of P_{def}, performance of the shufflenet can therefore be obtained as a function of the network size and buffer capacity. Throughput of a shufflenet scales well with both network and buffer sizes if the routing algorithm is chosen according to the general routing conditions we presented in Section 5.1.

7 APPENDIX

A Accuracy of $P^{(1)}_{def,bs}$ for the Hot-Potato and One-Buffer Cases

Our expression of $P^{(1)}_{def,bs}$ given in Equation (36) is not only accurate for $bs \geq 2$, but also reasonably good for the hot-potato (i.e., $n_0 = 4$) and 1-buffer (i.e., $n_1 = 16$) cases. This can be shown as follows.

Let P_{def} be the exact deflection probability obtained by solving Equation (7). Let ϵ be the error of $P^{(1)}_{def,bs}$ such that

$$P_{def} = P^{(1)}_{def,bs} + \epsilon, \tag{56}$$

where $P^{(1)}_{def,bs}$ is given in Equation (36). Then, using

$$P_{cr}(P_{def}) = P_{cr}(P^{(1)}_{def,bs}) + \epsilon\frac{dP_{cr}(\xi)}{dP_{def}}, \tag{57}$$

where ξ is a value between $P^{(1)}_{def,bs}$ and P_{def}, and Equation (7),

$$P_c = n_{bs}(P^{(1)}_{def,bs} + \epsilon), \tag{58}$$

we have,

$$P_{cr}(P^{(1)}_{def,bs}) + \epsilon\frac{dP_{cr}(\xi)}{dP_{def}} = n_{bs}(P^{(1)}_{def,bs} + \epsilon), \tag{59}$$

where Equation (26) has been used.

We therefore have,

$$\left|\frac{\epsilon}{P^{(1)}_{def,bs}}\right| \tag{60}$$

$$\simeq \left| \frac{P_{cr}(P^{(1)}_{def,bs}) - n_{bs} P^{(1)}_{def,bs}}{P^{(1)}_{def,bs}(n_{bs} - dP_{cr}(\xi)/dP_{def})} \right| \tag{61}$$

$$\approx \left| \frac{P_{cr}(P^{(1)}_{def,bs}) - n_{bs} P^{(1)}_{def,bs}}{P^{(1)}_{def,bs}(n_{bs} - dP_{cr}(0)/dP_{def})} \right| \tag{62}$$

$$= \left| \frac{B}{n_{bs} - dP_{cr}(0)/dP_{def}} \right| \quad \text{(Eqs. (27) and (28))} \tag{63}$$

$$= \left| \frac{1}{1 - n_{bs}/B} \right| \quad \text{(from Equation (28))}, \tag{64}$$

where B is given in Equation (32). As B is a decreasing function in k, it achieves its minimum at -1.23 with $k = 10$. We therefore have,

$$\left| \frac{\epsilon}{P^{(1)}_{def,bs}} \right| \lesssim \frac{1}{1 + 0.8 n_{bs}} \tag{65}$$

For hot-potato routing, $n_0 = 4$ and for the one-buffer case, $n_1 \simeq 16$. Therefore, our first order approximation of P_{def} is reasonably good.

B $E_{bs}[D]$ for $bs = 0$ or $bs = 1$

From Equation (37), we have

$$(1 - P^{(1)}_{def,bs})^k \approx (1 - \frac{2/3}{n_{bs} + k/9})^k \tag{66}$$

$$\approx e^{-\frac{2k/3}{n_{bs}+k/9}} \tag{67}$$

$$\approx \begin{cases} e^{-k/6} & n_0 = 4, \text{ hot-potato case,} \\ e^{-k/24} & n_1 \simeq 16, \text{ 1-buffer case,} \end{cases} \tag{68}$$

where we have used $(1 + 1/x)^x \approx e^x$ to get Equation (67).

Using Equation (2), the ratio of the third term to the first two terms in $E[D]$ gives, with $k \geq 3$,

$$\frac{\frac{k}{(1-P_{def})^k}}{\frac{kp^k}{kp^k - 1}\left[\frac{k-1}{2} - \frac{1 - p^{k-1}(1-P_{def})^{k-1}}{p^{k-1}(1-P_{def})^{k-1}(1-p+pP_{def})}\right]}$$

$$\approx \frac{\frac{k}{(1-P^{(1)}_{def,bs})^k}}{\frac{k-1}{2} - 1} \quad \text{(for } p = 2\text{)} \tag{69}$$

$$= \frac{2k}{k-3}(1 - P^{(1)}_{def,bs})^{-k} \tag{70}$$

$$\approx \begin{cases} \gg 1 & \text{hot-potato case,} \\ \frac{2k}{k-3} e^{k/24} & \text{1-buffer case,} \end{cases} \tag{71}$$

where we have used the approximation $(1 - 1/(2 - 2P_{def})^{k-1})/(1 - 2P_{def}) \approx 1$. Equation (68) is used to get Equation (71).

Therefore,

$$E_{bs}[D] \simeq \begin{cases} \frac{k}{(1-P^{(1)}_{def,0})^k} & bs = 0, \text{ hot-potato case,} \\ \gamma_k \frac{k}{(1-P^{(1)}_{def,1})^k} & bs = 1, \text{ 1-buffered case,} \end{cases} \tag{72}$$

where

$$\gamma_k = 1 + \frac{k-3}{2k} e^{-k/24}. \tag{73}$$

References

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

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

[3] 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.

[4] A. Acampora and S. Shah, "Multihop Lightwave Networks: A Comparison of Store-and-Forward and Hot-Potato Routing," *IEEE Transactions on Communications*, pp. 1082-1090, Vol. 40, No. 6, June 1992.

[5] A. Bononi, F. Forghieri and P. Prucnal, "Analysis of One-Buffer Deflection Routing in Ultra-Fast Optical Mesh Network," in *Proceedings of IEEE INFOCOM'93*, Vol. 1, pp. 303-311.

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

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

[8] S-H. Chan and H. Kobayashi, "Analysis of the Shufflenet with Different Contention Resolution Schemes," in *Proceedings of the 28th Annual Conference on Information Sciences and Systems (CISS)*, pp. 261-270, Princeton, March, 1994.

[9] I. Chlamtac and A. Fumagalli, "An All-Optical Switch Architecture for Manhattan Networks," *IEEE J. Selected Areas in Communication*, Vol. 11, No. 4, pp. 550-559, May 1993.