Packet Scheduling Algorithms and Performance of a Buffered Shufflenet with Deflection Routing

S.-H. Gary Chan, *Member, IEEE*, and Hisashi Kobayashi, *Fellow, IEEE*

***Abstract*—In a multihop network, packets go through a number of hops before they are absorbed at their destinations. In routing to its destination using minimum path, a packet at a node may have a preferential output link (the so-called "care" packet) or may not (the so-called "don't care" packet). Since each node in an optical multihop network may have limited buffer, when such buffer runs out, contention among packets for the same output link can be resolved by deflection. In this paper, we study packet scheduling algorithms and their performance in a buffered regular network with deflection routing. Using shufflenet as an example, we show that high performance (in terms of throughput and delay) can be achieved if "care" packets can be scheduled with higher priority than "don't care" packets.We then analyze the performance of a shufflenet with this priority scheduling given the buffer size per node. Traditionally, the deflection probability of a packet at a node is solved from a transcendental equation by numerical methods which quickly becomes very cumbersome when the buffer size is greater than one packet per node. By exploiting the special topological properties of the shufflenet, we are able to simplify the analysis greatly and obtain a simple closed-form approximation of the deflection probability. The expression allows us to extract analytically the performance trend of the shufflenet with respect to its buffer and network sizes. We show that a shufflenet indeed performs very well with only one buffer, and can achieve performance close to the store-and-forward case using a buffer size as small as four packets per node.**

***Index Terms*—Asymptotic performance, deflection routing, optical buffer, packet scheduling algorithm, shufflenet.**

I. Introduction

OPTICAL fiber provides a tremendous amount of bandwidth in excess of tens of terabits per second in its low-loss low-dispersion window at 1.2–1.6 μm. In wavelength-division-multiplexing (WDM) optical networks, such enormous bandwidth is divided into multiple wavelength channels whereby users may transmit and receive packets in parallel by tuning to the appropriate wavelengths.

Since wavelength-agile transmitters and receivers are still currently not available at low cost, each node in an optical network may be able to tune to only a limited number of wavelengths. By means of wavelength conversion, a packet can be forwarded to its destination through a series of intermediate channels. Such a "multihop" approach hence overcomes the current device limitations at the expense of some packet delay and network throughput. It should be noted that because of the enormous usable bandwidth in an optical fiber, communication schemes utilizing only a small fraction of such bandwidth can still achieve impressive throughput. For example, with a network making use of only 1% of the optical bandwidth, throughput in excess of hundreds of Gbits per seconds can be achieved.

Optical buffers may be used in high speed networks to avoid O/E (optics to electronics) and E/O (electronics to optics) conversion of data, the so-called "electronic bottleneck." Low-cost optical buffers are generally in the form of optical delay lines (ODL's) [1]–[3]. Since optical buffers are expensive, a node in an optical network generally has limited buffering. Packets are transmitted in the network in a store-and-forward manner, if there is enough buffer in the nodes. In the event of output contention, one of the packets will be routed correctly while the rest will be either buffered (if storage is available), or "deflected" or mis-routed temporarily to wrong channels. A deflected packet simply recirculates in the network and takes more hops to reach its destination. Therefore, in deflection routing, packets do not get lost due to a buffer overflow at the expense of some delay and bandwidth. A low deflection probability is generally of interest in such a system, as the performance of the network degrades with an increase in such probability. (The case with no buffering is called "hot-potato" routing.)

In this paper, we study packet scheduling algorithms and performance of a buffered regular network with deflection routing. The network we consider is a shufflenet, though the results and analytic methodologies can be extended to other networks of similar type. A shufflenet is a regular WDM multihop network proposed to interconnect multiple computers or processors together [4], [5]. A node in a shufflenet accesses the network through a number of lightwave receivers and transmitters. A shufflenet is characterized by two numbers p and k, where p is the number of wavelength channels that a node can receive or transmit while k is the number of columns in the network. A (p, k) shufflenet consists of $N = kp^k$ nodes arranged in k columns of p^k nodes each. We show in Fig. 1 the (2, 3) shufflenet (which has 24 nodes). The nodes are interconnected as a perfect shuffle, with the last column being cylinder; therefore, packets can continuously "re-enter" the network until they are absorbed at their destinations. The maximum distance in hops between any two nodes in the (p, k) shufflenet is $2k - 1$, independent of p.

Manuscript received July 19, 1999; revised January 5, 2000. This work was supported in part by a grant from the Ogasawara Foundation for the Promotion of Science and Engineering; by the Center for Ultafast Laser and Applications (CULA) of the New Jersey Commission on Science and Technology; and by the NCIPT (National Center for Integrated Photonic Technology) through the ARPA grant.

S.-H. G. Chan was with Princeton University, Princeton, NJ 08544 USA. He is now with the Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: gchan@cs.ust.hk).

H. Kobayashi is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: hisashi@ee.princeton.edu).

Publisher Item Identifier S 0733-8724(00)03040-1.

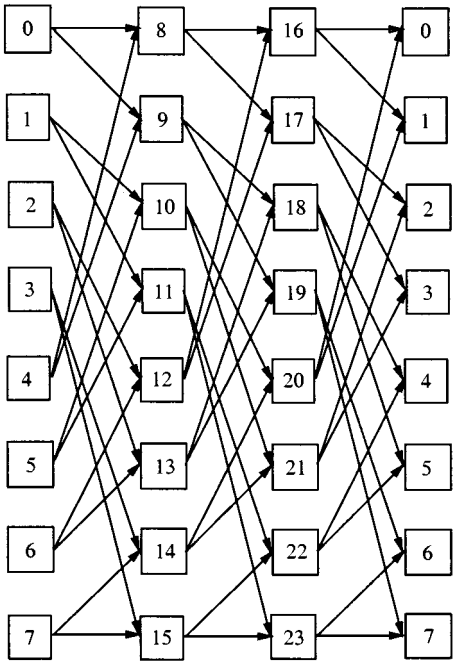

Fig. 1. An example of a $(p,\ k)$ shufflenet, with $p = 2$ and $k = 3$.

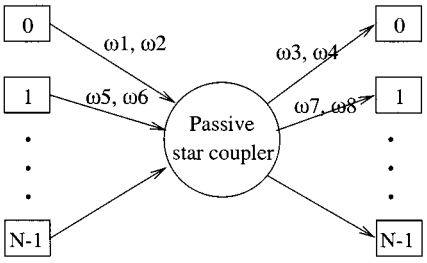

Fig. 2. A physical implementation of a $(2,\ k)$ shufflenet (with $N = k2^k$) using a passive star.

We show in Fig. 2 a physical implementation of a $(2,\ k)$ shufflenet (as opposed to the logical topology as shown in Fig. 1) using a central star coupler, in which each of the $N = k2^k$ nodes (labeled as $0,\ 1,\ \ldots,\ N-1$) is connected to the coupler by a fiber, and transmits and receives wavelengths labeled by ω's. If all the fibers are of the same length, the transmission and reception may be synchronized in the network (i.e., a time-slotted network). Other implementations using a bus or multiconnected ring topology have been discussed in [4], [6], [7]. A recent shufflenet experimental system has also been reported in [8].

In the $(p,\ k)$ shufflenet, each node can be identified by its address (c, r), where $c \in \{0,\ 1,\ \ldots,\ k-1\}$ (stands for the column) and $r \in \{0,\ 1,\ \ldots,\ p^k - 1\}$ (stands for the row). For a given packet at node $(c,\ r)$, let D be the number of columns between the source $(c,\ r)$ and the destination $(c^d,\ r^d)$. We clearly have

$$D = \begin{cases} (c - c^d) \bmod k, & \text{if } c^d \neq c \\ k, & \text{if } c^d = c \end{cases} \tag{1}$$

where D represents the lowest bound on the number of hops for the packet to go from $(c,\ r)$ to $(c^d,\ r^d)$. A node is said to be a "don't care" node to 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, a packet at its "don't care" node is called a "don't care" packet and will not suffer deflection in the node.) A property of a shufflenet is that a packet at a node is "don't care" if its destination is more than k hops away from the node when there is no deflection. In this case, it is not possible to route the packet in D hops; instead it takes $D + k$ hops. This is an important topological property of shufflenet which greatly reduces the state space when we analyze the network, as will be shown later in the paper. Conversely, all the nodes that are within k hops from a packet's destination are "care" nodes in which the packet has a preferential output channel/link in order to be routed to its destination with the minimum number of hops (the packet is hence called a "care" packet in this case). For the $(p,\ k)$ shufflenet, the distance of a packet from its current node to its destination in the case of a deflection is increased by k hops as compared to the distance without deflection. Therefore, except for the last hop of a packet, each deflection puts a packet from its "care" node to one of its "don't care" nodes (and hence turning it from a "care" packet to a "don't care" packet). In this paper, we will mainly focus on the $(2,\ k)$ shufflenet.

In scheduling a packet in a deflection network, we have to considerwhether it is "care" and "don't care" in the node. We study here a nonpriority first-in-first-out scheme in which the packets are routed regardless of their classes, and a class-based priority scheme in which the "care" packets are routed at a higher priority than the "don't care" packets. Using simulation, we compare the performance of the schemes for a given buffer size per node. No matter how large the buffer size is, the performance of the nonpriority scheme is found to deteriorate to that of the hot-potato routing as the load increases, indicating that the buffers fill up very quickly in this scheme. On the other hand, though packets may not be served according to their arrival order, the priority scheme achieves substantially better throughput and delay than the nonpriority scheme. This result suggests the advantages in scheduling packets according to their classes in a deflection network.

In a shufflenet with deflection routing, it has been observed that using just one buffer can lead to a substantial performance improvement (as compared to the case of hot-potato routing) and achieve performance close to the infinite buffer case. However, there has not been a study to show explicitly how the throughput scales with respect to the buffer size and the network size. In this paper, we address this issue by first observing that the performance of a shufflenet under uniform traffic is known once the deflection probability of a packet in the network is obtained. Such probability can be obtained by solving numerically an implicit transcendal equation given a certain routing algorithm and buffer size [1], [9]. Most of the previous analyzes focus on the cases of zero buffer (hot-potato routing) and one buffer, mainly because the trancendental equation and the number of states that we need to keep track of become complex (and hence the numerical method becomes cumbersome) as the buffer size goes beyond one (except for the special case of store-and-forward routing which corresponds to the infinite buffer case). This makes it difficult to analyze the trend of the maximum throughput of a shufflenet (i.e., its "asymptotic" performance when the network load increases) with respect to the buffer size. By exploiting the topological properties of shufflenet, we are able to greatly simplify

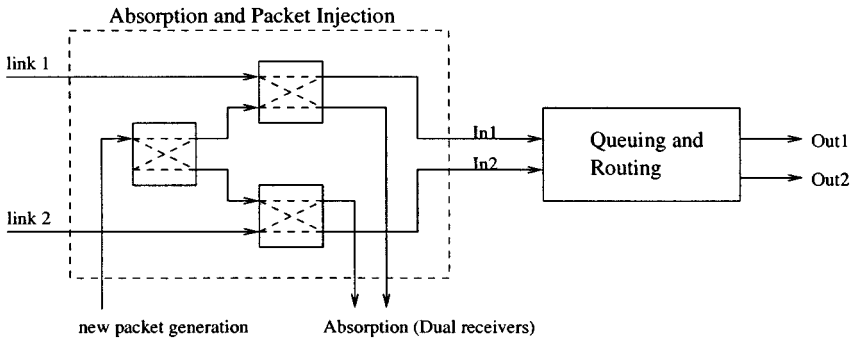

Fig. 3. Simulation steps performed at each node in the $(2, k)$ shufflenet.

the analysis of a shufflenet by reducing the state space of the corresponding Markov chain from $O(N)$ to $O(\log N)$. Based on this, we obtain an approximate formula for the deflection probability as a function of the buffer and network sizes for the class-based scheduling algorithm. We show that the deflection probability decreases rapidly with the buffer size B in a node and is asymptotically given by $O(B^{-1}2^{-B})$. Our approximation agrees very well with simulation results. Using the expression, we obtain the asymptotic throughput of the shufflenet and show that using only one buffer in a shufflenet node can indeed achieve high throughput, and the store-and-forward throughput is more or less achieved with buffer size as low as four. Such relative gain in performance for the one buffer case, however, decreases as the network size increases.

The paper is organized as follows. We first briefly review previous work in Section II. In Section III, we present the priority and nonpriority scheduling algorithms and their performance based on simulation. In Section IV we present the performance analysis of the shufflenet: We first discuss shufflenet analysis given the deflection probability, and then derive an approximate expression of the probability which allows us to evaluate the throughput of the buffered shufflenet given its buffer and network sizes. We conclude in Section V.

II. Previous Work

We briefly review previous work as follows. Analyzes of shufflenet have been recently reported by several authors [9]–[14]. We greatly reduce the state space by exploiting its special topological property. We also quantify for the first time analytically the effect of buffer size in shufflenet performance. In [14], split output queues have been considered. We differ from it in using shared queuing and class-based scheduling. The priority scheme we consider also achieves higher throughput. Most of the previous analyzes focus on hot-potato routing. While the one-buffer case is treated in [9], we consider the multiple buffer case here, which necessitates the consideration of "care" and "don't care" packet classes.

Some contention resolution schemes based on packet age or its distance from its destination have been reported in [15], [16]. It has been shown that if priority is given to an old packet or a packet with a shorter distance from its destination, slightly better throughput can be achieved. Architectural changes in the shufflenet have also been proposed in order to decrease the deflection probability: In [17], an alternate path is provided (via another channel) for the deflected packets so that the increase in path length would not be high, while a recirculating shufflenet with multiple cylinders is studied in [18] to decrease the deflection probability.

Another class of shufflenet called the "generalized shufflenet" has been proposed and studied in [19], [20] so that the number of nodes N is not restricted to kp^k. Our work on the "conventional" shufflenet would be useful in deriving the performance of this extended class of shufflenet. A bidirectional shufflenet, in which the channels are bidirectional for flow control and throughput improvement, has also been proposed (see [21], [22], and references therein). We will not address the performance of this network here. Another regular multiconnected network known as a Manhattan street network has been proposed in [23], [24]. An analysis of deflection routing in such a mesh network has been reported in [25].

Optimization of the shufflenet has been discussed in various aspects: In [26], [27], both static and dynamic nodal placements in a shufflenet with nonuniform traffic has been discussed. Implementation of a shufflenet for reconfigurability and scalability has been studied in [28], [29], while data multicast in a shufflenet has been treated in [30].

III. Scheduling Algorithms

A. Packet Scheduling Algorithms

We consider a $(2, k)$ shufflenet in which the time is slotted, with the packet transmission time being one time slot. The fibers are of equal length and the propagation delay of a packet in the fiber equals to a time slot. We show in Fig. 3 the procedure a packet undergoes in a node in each slot (or clock cycle), which is given as follows. The packets from the incoming links are first checked for their destination addresses. Packets that are destined to the current node are absorbed (i.e., delivery of the packets to their destination node). The absorption can take place on both links within a single time slot. We consider uniform load in which each node in the network generates a packet with probability g in each time slot independent of all the other nodes in the network. A newly generated packet is injected into the network only if at least one of the links is freeafter the absorption; otherwise it is discarded and cleared. The probability g is called the offered load of the network. The destination for a newly generated packet is uniformly distributed among all the other $N-1$ nodes in the network. The injected packet along with the other

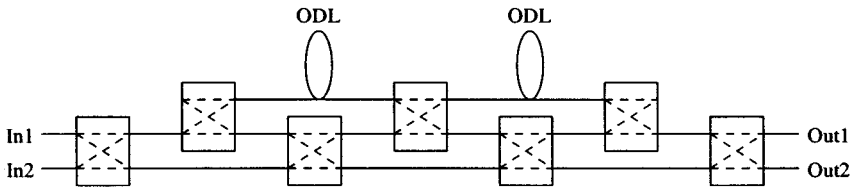

Fig. 4. Optical storage using delay lines with buffer size equal to 2.

packet, if any, are then queued or routed according to a scheduling algorithm. Packets are routed to an output channel using the shortest path algorithm.

In the event of output contention, one of the packets is put into a buffer (if available). We show in Fig. 4 a nonblocking optical storage architecture with buffer size $B = 2$. An optical delay line (ODL) delays the packet by one time slot. Note that packets can be randomly accessed in such node, and a packet will stay in the node at most B time slots. We will assume this storage architecture in this paper.

As mentioned before, there are two types of packets in a node, "don't care" packets and "care" packets. Routing decisions are done at the beginning of a time slot and packets are routed within the same time slot. We consider the following scheduling algorithm to decide which packets should be routed in each slot.

- First-in–first-out (FIFO):This is a nonpriority scheme in which the packets are routed strictly in the first-in-first-out manner. At the beginning of each slot, the oldest packet in the buffer (if any) would be scheduled first and hence it would not suffer deflection. If it is a "don't care" packet, the next oldest packet, if any, would also be routed without deflection. However, if it is a "care" packet, and the next oldest packet is also a "care" packet but with a different preferential output link, both packets will be routed in the same slot. If only one packet is in the buffer and two packets come in at the beginning of a slot, one of the incoming packets will be chosen at random for routing. Incoming packets not routed in the current slot are put into the buffer in random order. Clearly, a deflection occurs when there are two incoming packets (at `In1` and `In2`), the buffers are full and the two packets considered for routing contend for the same output link.
- Class-based scheduling: This is a priority scheme in which the "care" packet(s) are routed first, followed by the "don't care" packet(s). This scheme is expected to improve the performance as it reduces the number of "care" packets in a node by routing without deflection as many "care" packets as possible in a time slot. A "don't care" packet is transmitted only when it has stayed in the buffer for B time slots, or when there are no "care" packets in the node, or when all the "care" packets in the node contend for the same output link. A deflection occurs only when two packets come in, the buffers are full, all the packets in the node (including the two incoming ones) are "care" packets, and they contend for the same output link.

B. Simulation Results

We have simulated the aforementioned scheduling policies for the $(2, k)$ shufflenet under uniform load (i.e., all the nodes in the network have the same traffic characteristics). The performance measures of interest are the following.

- Normalized throughput (i.e., throughput per node) λ—It is defined as the average number of packets absorbed per node per time slot (λ is hence also the expected number of new packets generated per time slot). Note that λ is strictly less than the offered load g, as a newly generated packet is not always injected into the network due to two unabsorbed by-passing packets at a node [note that $(g - \lambda)/g$ is the fraction of newly generated packets that are discarded or cleared].
- The hop distribution and the average number of hops $E[H]$—We define the number of hops as the number of nodes a packet visits (including the ending or destination node) before it is absorbed in the destination. The hop distribution and the average number of hops are indicators of the delay performance of the network. In deflection routing, the delay increases with the deflection probability.

We show in Fig. 5 the throughput of the (2, 4) shufflenet with FIFO versus the offered load g, given buffer size B ($B = 0, 1, 2, 4$, and 8). When $B \geq 1$, the throughput first increases and then decreases. There exists an offered load such that the throughput of the network is maximized. We note that with high g, no matter how large the buffer size may be, the throughput reduces to the hot-potato case. This is because of the indiscriminating nature in the FIFO, which fills up the buffers very quickly. We also observe instability in the case of infinite buffer size under heavy load, as the number of stored packets increases without bound.

We show in Fig. 6 the normalized throughput λ for the (2, 4) shufflenet versus the offered load g for the class-based scheduling, given B. Clearly, λ increases with g. With $B = 4$, the throughput is very close to the store-and-forward case, suggesting that deflections have been mostly eliminated with this buffer size. Therefore, in a shufflenet with deflection routing, the amount of buffering does not need to be large in order to achieve a high performance. The throughput is substantially improved with merely one buffer, as has been observed by other investigators as well [9], [14].

We show in Fig. 7 the expected number of hops $E[H]$ versus g for the hot-potato, FIFO (1-buffer case) and class-based scheduling (1-buffer case). As g increases, $E[H]$ increases and the hop distribution spreads due to deflection (the hop distribution given $E[H]$ will be discussed in Section IV). For FIFO, $E[H]$ increases quickly toward the hot-potato case as g increases, while for the class-based scheduling, $E[H]$ remains at a low level. Our results suggest that packet scheduling based on packet classes leads to a good performance.

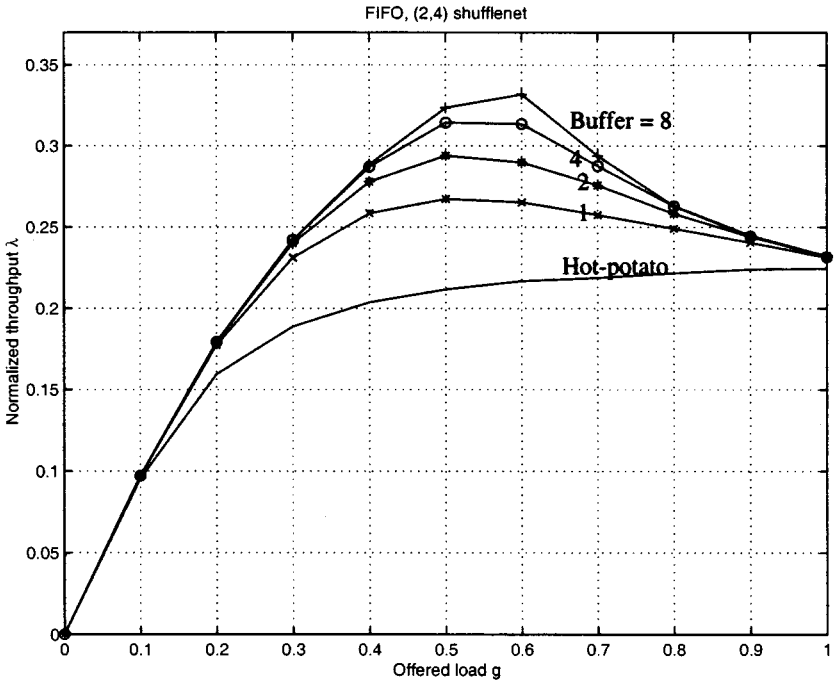

Fig. 5. λ versus g for the FIFO scheduling policy in the (2, 4) network given B.

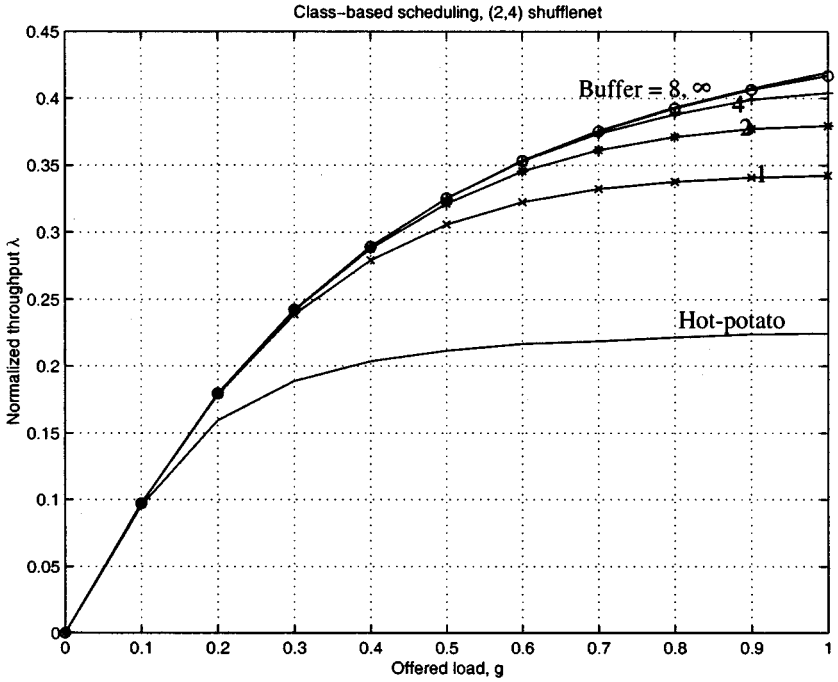

Fig. 6. λ versus g for the class-based scheduling policy given B in the (2, 4) shufflenet.

IV. Performance of a Buffered Shufflenet

A. Shufflenet Analysis

In this section, we present the analysis of a shufflenet under the uniform load. One important parameter in the analysis is the deflection probability of a packet in its "care" node, P_{def}, which critically determines how well the shufflenet performs. In fact, there is a unique relationship between P_{def} and the shufflenet performance in terms of throughput and the number of hops.

We first observe that the shufflenet is a symmetric network. When the load is uniform, all the nodes are equivalent; thus we can focus on one node and, by analyzing its performance, we can obtain the global performance of the network. Using this "one node model," we further make the following independence assumptions:

- a node may have a packet on an incoming link independent of whether there is another packet on the other link;
- a packet is equally likely to take any one of the two output links, independent of other packets in the node;

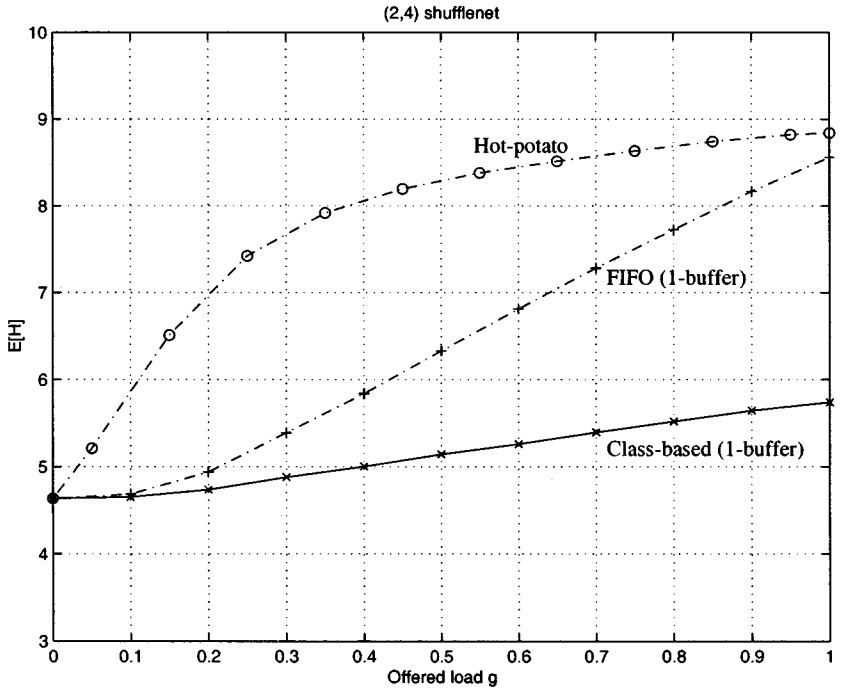

Fig. 7. $E[H]$ versus g in the (2, 4) shufflenet.

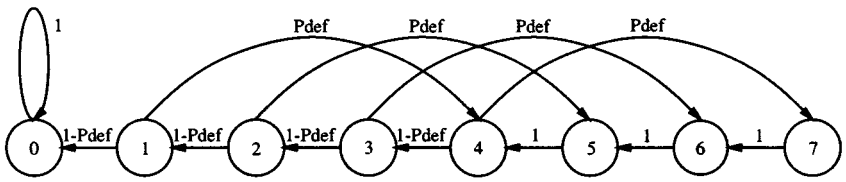

Fig. 8. State transition diagram for the $(2, 4)$ shufflenet.

- on its way to its destination, a "care" packet in a node is deflected with probability $P_{\rm def}$, and routed correctly to its output channel with probability $(1 - P_{\rm def})$, independent of the distance from its destination node (so long as it is less than or equal to k).

In the following, we first derive the hop distribution and its average, and then the probability of don't care, $P_{\rm dc}$, defined as the probability that a packet visits one of its "don't care" nodes in a given hop. Note that except for its last hop, a packet in the shufflenet is always deflected to its "don't care" node. Let $N^{\rm dc}$ be the random variable which represents the number of "don't care" nodes that a packet visits on its way to the destination, and let H be the number of hops that the packet takes. Then we have

$$P_{\rm dc} \triangleq \frac{E[N^{\rm dc}]}{E[H]}. \tag{2}$$

We now present the analytical relationships among $P_{\rm def}$, $P_{\rm dc}$, the hop distribution and the expected number of hops, $E[H]$. Recall that in any (p, k) shufflenet, the "care" nodes for a packet are the nodes within diameter k hops from its destination. All the other nodes are "don't care" nodes where the packet will not suffer deflection. Using this property, we need to deal with a Markov chain with only $2k$ states, instead of N states as usually used in studying mesh networks.

Let us select a packet arbitrarily and observe the trajectory of the "tagged" packet. Let our state space $S = \{0, 1, \ldots, 2k-1\}$ be a collection of possible distances between the current position of the tagged packet and its destination, where the distance is defined as the minimum number of hops that the packet must make to travel to its destination in the absence of deflection. Let $H_i = E$ [number of hops when the tagged packet is at distance i from its destination | probability of deflection $= P_{\rm def}$], $\forall\, i \in S$. We model the network as an absorbing Markov chain with state space S, and state 0 is the absorbing state. [We show in Fig. 8 the state transition diagram for the $(2, 4)$ shufflenet.] As each deflection increases the packet's hops by k, we have $H_i = P_{\rm def}H_{i-1+k} + (1 - P_{\rm def})H_{i-1} + 1$, for $1 \le i \le k$. When the packet is at a distance greater than k hops from its destination, it is at its "don't care" node and therefore will not suffer a deflection (i.e., $P_{\rm def} = 0$). Hence, $H_i = H_k + (i - k)$, for $k+1 \le i \le 2k-1$. Since state 0 is the destination of the tagged packet, we have $H_0 = 0$. Therefore

$$H_j = \begin{cases} j + \dfrac{k}{(1 - P_{\rm def})^k}[1 - (1 - P_{\rm def})^j], & 1 \le j \le k \\ H_k + (j - k), & k + 1 \le j \le 2k - 1. \end{cases} \tag{3}$$

Note that the above equation does not depend on the parameter p of the (p, k) shufflenet.

Note that for $1 \le j \le k-1$, there are p^j nodes at j hops away from a given node, and for $0 \le j \le k-1$, there are $(p^k - p^j)$ nodes at $k+j$ hops away. The expected number of hops, $E[H]$, for any packet in the network is therefore given by

$$E[H] = \frac{1}{kp^k - 1}\left[\sum_{j=1}^{k-1} p^j D_j + \sum_{j=0}^{k-1}(p^k - p^j) D_{k+j}\right]. \quad (4)$$

To find the probability distribution of the number of hops, we form a $(2k) \times (2k)$ transition matrix, $\boldsymbol{T}$, where the entry $t_{i,j}$ is the one step transition probability given by $t_{i,j} \triangleq P$ [the tagged packet at distance j from its destination hops to distance i in the next time slot], $\forall i, j \in S$. Then we have

$$\begin{aligned} t_{0,0} &= 1, \quad \text{(absorbing state)} \\ t_{i+k-1,i} &= P_{\text{def}}, \quad \text{for } 1 \le i \le k \\ t_{i-1,i} &= \begin{cases} 1 - P_{\text{def}}, & \text{for } 1 \le i \le k \\ 1, & \text{for } k+1 \le i \le 2k-1 \end{cases} \\ t_{i,j} &= 0, \quad \text{elsewhere.} \end{aligned} \quad (5)$$

Clearly $t_{i,j}$'s satisfy $\sum_{i \in S} t_{i,j} = 1, \forall j \in S$.

Let $\boldsymbol{P}_0$ and $\boldsymbol{P}_n$ be column vectors. The ith element of $\boldsymbol{P}_0$ is the initial probability that a packet is generated at distance i and the ith element of $\boldsymbol{P}_n$ is the probability that the packet visits state i in its nth hop, where $i \in S$. Then, $\boldsymbol{P}_n = \boldsymbol{T}\boldsymbol{P}_{n-1} = \boldsymbol{T}^n \boldsymbol{P}_0 = \boldsymbol{S}\boldsymbol{\Lambda}^n \boldsymbol{S}^{-1} \boldsymbol{P}_0$, where $\boldsymbol{\Lambda}$ is the diagonal matrix formed by the eigenvalues of $\boldsymbol{T}$, and $\boldsymbol{S}$ is the corresponding matrix formed by the eigenvectors of $\boldsymbol{T}$. As the destination of a packet just generated is randomly distributed among all the other $(N-1)$ users in the network, the initial probability distribution for the distance of a packet can be written as

$$\boldsymbol{P}_0 = \frac{1}{kp^k - 1}\begin{pmatrix} 0 \\ p \\ \cdot \\ \cdot \\ p^{k-1} \\ p^k - 1 \\ p^k - p \\ \cdot \\ \cdot \\ p^k - p^{k-1} \end{pmatrix}. \quad (6)$$

Note that since the underlying Markov chain has one absorbing state (i.e., state 0) and all the other $(2k-1)$ states are transient, it is known that only one eigenvalue of the transition matrix $\boldsymbol{T}$ is unity, and the magnitudes of all the other eigenvalues are strictly less than 1. Therefore in the limit

$$\lim_{n \to \infty} \boldsymbol{\Lambda} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ . & . & . & . \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

and from the theory of Markov chain, we can show that $\lim_{n \to \infty} \boldsymbol{P}_n = \boldsymbol{e}_0$, where $\boldsymbol{e}_0$ is a column vector whose entries are all zero except the first element (which corresponds to state 0), which is unity. Therefore the tagged packet reaches its destination with probability one.

Let Q_n be the probability that a packet reaches its destination at the nth hop. $\{Q_n, n = 1, 2, 3, \dots\}$ is then the probability distribution of the number of hops taken by a packet and is given by $\sum_{i \le \min}(n, 2k-1) P$. [Packet generated at distance i reaches its destination exactly at the nth hop.] Since the state $0 \in S$ is the absorbing state, Q_n is given by the first element of the vector $(\boldsymbol{P}_n - \boldsymbol{P}_{n-1})$. Thus, we have

$$Q_n = \boldsymbol{e}_0^T \cdot (\boldsymbol{P}_n - \boldsymbol{P}_{n-1}), \quad \text{for} \quad n = 1, 2, 3, \dots \quad (7)$$

where $\boldsymbol{e}_0^T$ is the transpose of the vector $\boldsymbol{e}_0$.

We next obtain P_{dc}. Let $N_i^{\text{dc}} = E$ [number of "don't care" nodes that the tagged packet at distance i visits in its lifetime]. We obviously have $N_i^{\text{dc}} = P_{\text{def}} N_{i+k-1}^{\text{dc}} + (1 - P_{\text{def}}) N_{i-1}^{\text{dc}}$, for $1 \le i \le k$, and $N_i^{\text{dc}} = N_k^{\text{dc}} + (i-k)$, for $k+1 \le i \le 2k-1$. Given that $N_0^{\text{dc}} = 0$, we have

$$N_j^{\text{dc}} = \begin{cases} D_j - \dfrac{1 - (1 - P_{\text{def}})^j}{P_{\text{def}}(1 - P_{\text{def}})^k}, & 1 \le j \le k \\ N_k^{\text{dc}} + (j-k), & k+1 \le j \le 2k-1. \end{cases} \quad (8)$$

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

$$E[N^{\text{dc}}] = \frac{1}{kp^k - 1}\left[\sum_{j=1}^{k-1} p^j N_j^{\text{dc}} + \sum_{j=0}^{k-1}(p^k - p^j) N_{k+j}^{\text{dc}}\right]. \quad (9)$$

The probability that the node which the tagged packet visits is a "don't care" node is then given by (2) with the use of (4) and (9).

As an illustrative example, let us consider the (2, 4) shufflenet. From (5), the transition matrix $\boldsymbol{T}$

$$\begin{pmatrix} 1 & 1-P_{\text{def}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1-P_{\text{def}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-P_{\text{def}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-P_{\text{def}} & 0 & 0 & 0 \\ 0 & P_{\text{def}} & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & P_{\text{def}} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & P_{\text{def}} & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & P_{\text{def}} & 0 & 0 & 0 \end{pmatrix} \quad (10)$$

and from (6)

$$\boldsymbol{P}_0 = \tfrac{1}{63}\begin{pmatrix} 0 \\ 2 \\ 4 \\ 8 \\ 15 \\ 14 \\ 12 \\ 8 \end{pmatrix}. \quad (11)$$

We present in Fig. 9 $E[H]$ versus P_{def} [according to (4)] in solid line for the (2, 4) shufflenet, along with the points obtained from simulating the shufflenet (using the scheduling algorithms discussed in the previous section with different buffer sizes), from which we see that the analysis and simulation agree to each other, validating the independence assumptions made. As P_{def} increases, $E[H]$ also increases. The deflection probability

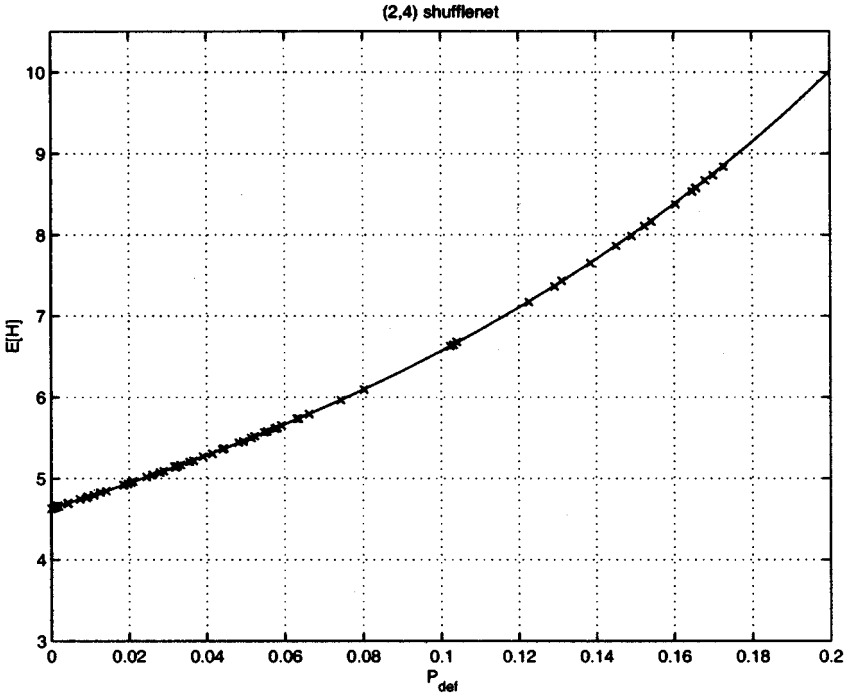

Fig. 9. Simulation and analytic results of $E[H]$ versus $P_{\rm def}$ for the $(2, 4)$ shufflenet.

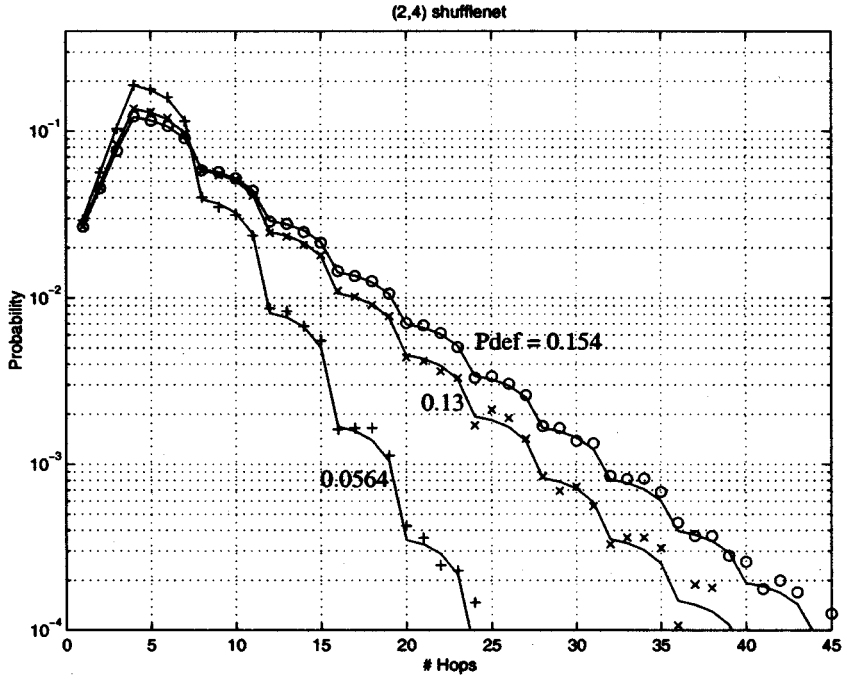

Fig. 10. Probability distribution for the number of hops for the $(2, 4)$ shufflenet given $P_{\rm def}$.

is generally low (less than 0.15), and most of them are less than 0.05, which corresponds to the class-priority scheme with buffer size 1 or greater. Note that $\lim_{P_{\rm def}\to 0} E[H] = 4.6349$, and $\lim_{P_{\rm def}\to 0} P_{\rm dc} = 0.2123$.

In Fig. 10, we show the corresponding hop distribution [i.e., Q_n, $n = 1, 2, 3, \ldots$ of (7)] given $P_{\rm def}$. We also show in discrete points the simulation results for hot-potato routing with $g = 0.1$ (corresponding to $P_{\rm def} = 0.0564$), $g = 0.3$ (corresponding to $P_{\rm def} = 0.13$) and $g = 0.5$ (corresponding to $P_{\rm def} = 0.15$). The "ripples" of four are expected, due to the fact that the shufflenet has $k = 4$ columns and all packets within $k = 4$ hops from their destinations are "care" packets and hence are potentially deflected. We verify the exponential tail of the hop distributions as reported by others [12].

B. Asymptotic Throughput of a Buffered Shufflenet

In this section, we present the asymptotic throughput of a buffered shufflenet. Recall that "asymptotic throughput" means the maximum throuhgput when $g = 1$. Let $\alpha = 1/E[H]$ be the probability of a packet being absorbed in a given hop. The

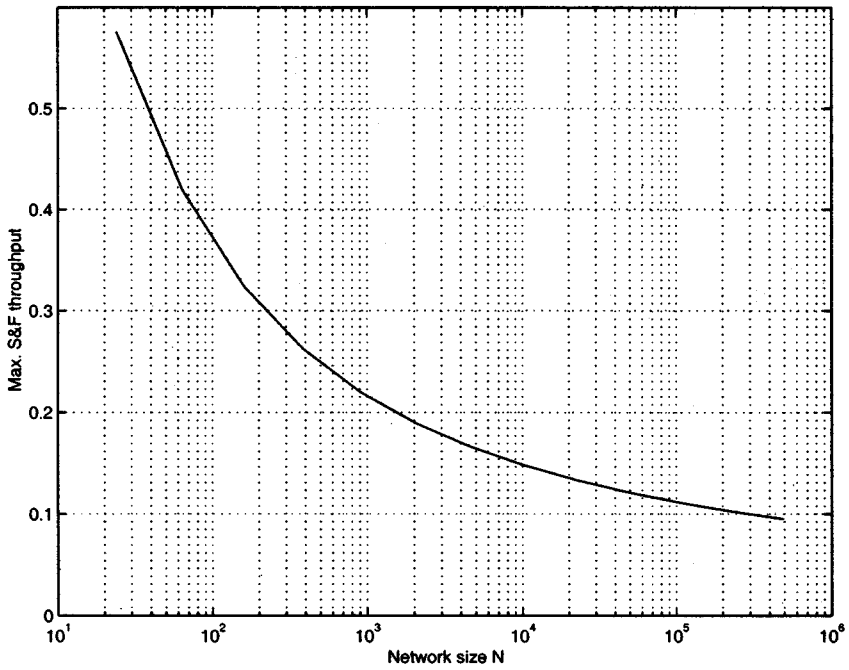

Fig. 11. Maximum store-and-forward throughput of a shufflenet verses N.

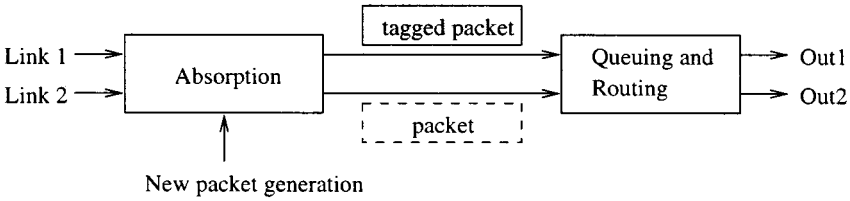

Fig. 12. Diagram to obtain the probability of deflection, $P_{\rm def}$, in the network.

asymptotic throughput of a $(2, k)$ shufflenet, $\hat{\lambda}$, has been given by [9]

$$\hat{\lambda} = 2\frac{\sqrt{\alpha^2 + (1-\alpha)^2} - \alpha}{(1-\alpha)^2}\,\alpha. \tag{12}$$

Note that as α is a function of $P_{\rm def}$, so is $\hat{\lambda}$. We see from the above that the average number of hops, the hop distribution, and the throughput of the network can all be analytically obtained once $P_{\rm def}$ is known.

Note that $P_{\rm def} = 0$ in the store-and-forward case. We show in Fig. 11 the asymptotic throughput $\hat{\lambda}$ using (12) versus network size N. The maximum throughput of the shufflenet decreases as N increases, but it does not decrease linearly (ranging from around 0.7 for 24-nodes to around 0.1 for as many as 10^5 nodes). Note that, from (4), $\lim_{P_{\rm def}\to 0} E[H] = k(2 - 3\cdot 2^k + 3\cdot 2^k k)/(2(-1+2^k k)) \approx 3(k-1)/2$; and hence $\alpha \approx 2/(3(k-1))$. When $\alpha \ll 1$ (say, $k = 3$ or higher), $\hat{\lambda} \approx 2\alpha$. Therefore, for the store-and-forward case

$$\hat{\lambda} \approx \frac{4}{3(k-1)}. \tag{13}$$

For finite buffers, we obtain the deflection probability for a packet in the network by following the trajectory of an arbitrary "tagged" packet and consider its deflection as shown in Fig. 12. Note that at a given slot, there is almost always a packet on the other input link besides the tagged packet (because $g \simeq 1$). The probability that the packet is "care," and hence would contend with the "tagged" packet in its "care" node, is given by $1 - P_{\rm dc}$ (obviously there would be no deflection if the packet is at its "don't care" node). Therefore, the deflection probability of the "tagged" packet, $P_{\rm def}$, is given by

$$P_{\rm def} = (1 - P_{\rm dc}) \cdot \frac{1}{n_B} \tag{14}$$

where $1/n_B$ is the probability that the "tagged" packet is deflected given that there is a "care" packet on the other link. Therefore, the larger the n_B is, the less likely a packet will be deflected in the network. The value n_B depends not only on B, but also on the scheduling algorithm (for example, $n_0 = 4$ for hot-potato routing and $n_\infty = \infty$ for store-and-forward routing [9], [31]). Note that since $P_{\rm dc}$ is a function of $P_{\rm def}$, (14) is a nonlinear equation in $P_{\rm def}$ which can be solved by numerical methods. In the following, we obtain an approximate value of $P_{\rm def}$, from which we obtain $\hat{\lambda}$.

We consider a class-based scheduling algorithm, in which the "care" packets are routed first before the "don't care" packets. Using this routing algorithm, at least one "care" packet is sent out of the buffer in each cycle. We model the transition of the buffer as a Markov chain, with the state being the number of "care" packets in the buffer. By observing that in any time slot, two "care" packets may be routed and at most one "car" packet may be added into the buffer, we therefore obtain the buffer

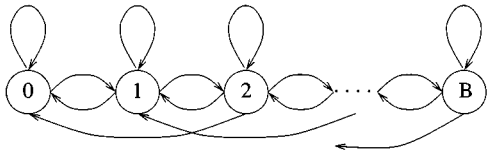

Fig. 13. State transition diagram for a shufflenet with buffer size B using the class-based scheduling.

TABLE I
n_B FOR THE CLASS-BASED SCHEDULING ALGORITHM

B	0	1	2	3	4
n_B	4	16	48	128	320

state transition diagram as shown in Fig. 13. Let π_i ($0 \le i \le B$) be the steady state probability that the buffer is in state i, with $\sum_{i=0}^{B} \pi_i = 1$. It should be noted that exact value of the transition probability from buffer state i to state j, depends on the routing algorithm (such transition probability for $j = i+2$ using the class-based scheduling is clearly zero).

As the number of "care" packets in the buffer increases, there are more choices of "care" packets to be routed in a time slot; hence, an upward state transition is less probable. Therefore, the steady state buffer occupancy probability likely satisfies $\pi_i \ge \pi_{i+1}$. The steady-state probability, π_B, that the buffer is full of "care" packets can then be expressed as

$$0 \le \pi_B \le \frac{1}{B+1}. \tag{15}$$

Using the class-based routing algorithm, our "tagged" is deflected when there is another "care" packet on the other link [occurs with probability $(1 - P_{\rm dc})$], the buffer is full of "care" packets (occurs with probability π_B), all the current $B+2$ "care" packets contend for the same output channel, and the "tagged" packet loses with a coin flip (occurs with probability $1/2^{B+2}$). Therefore, a self-consistent equation for the deflection probability of the "tagged" packet at its "care" node, $P_{\rm def}$, can be expressed as

$$P_{\rm def} = P_c \pi_B \frac{1}{2^{B+2}}. \tag{16}$$

Comparing (14)–(16), we have $n_B \ge (B+1)2^{B+2}$. From simulation, we found that

$$n_B = (B+1)2^{B+2} \tag{17}$$

is a good approximation which we will assume for the rest of the following discussion. We show in Table I the values of n_B for different buffer sizes. Note that n_B increases very rapidly with B. This is the main reason why the deflection probability decreases very rapidly as buffer size increases.

We now obtain a first-order approximation of $P_{\rm def}$ given buffer size B. Let $P^{(1)}_{\rm def,\,B}$ be such approximation, which is obtained by observing that $P_{\rm def}$ is small (Fig. 9, 2, 4 shufflenet). Using (4) and (9), we expand (2) around $P_{\rm def} = 0$ to obtain

$$1 - P_{\rm dc} \approx A + C P^{(1)}_{\rm def,\,B} \tag{18}$$

where $A = 2(2 - 2^{k+1} + k + 2^k k^2)/(k(2 - 3\cdot 2^k + 3\cdot 2^k k)) \approx 2k/(3(k-1))$, and $C = c_1/c_2 \approx (-24 + 13k - k^3)/(9(k-1)^2) \sim -k/9$, where $c_1 = -8\cdot 2^k + 8\cdot 2^{2k} - 6k + 25\cdot 2^k k - 24\cdot 2^{2k} k - 2k^2 - 2\cdot 2^k k^2 + 13\cdot 2^{2k} k^2 - 3\cdot 2^k k^3 - 2^{2k} k^4$, and $c_2 = k(2 - 3\cdot 2^k + 3\cdot 2^k k)^2$. Equations (14) and (18) yield

$$P^{(1)}_{\rm def,\,B} = \frac{A}{n_B - C} \tag{19}$$

$$\sim \frac{2/3}{n_B + k/9}. \tag{20}$$

A necessary condition for the expansion of (18) to be accurate is $kP^{(1)}_{\rm def,\,B} \ll 1$. Using (20), we therefore need $n_B \gg 5k/9$. With $k \le 10$ (corresponding to more than 10^4 nodes), we need $n_B \gg 4$, which is clearly satisfied when $B \ge 1$. Note that as B increases, $P_{\rm def} \sim O(B^{-1}2^{-B})$.

Let $\hat{\lambda}_B$ be the normalized throughput of the $(2, 4)$ shufflenet given its buffer size B by using the expression of $P^{(1)}_{\rm def,\,B}$ in (19) and substituting it into (12). We show in Table II the approximate asymptotic throughput for different values of k with $B = 1$ (i.e., with $n_1 = 16$). The simulation values for $P_{\rm def}$ and $\hat{\lambda}_1$ when $k = 4$ and 6 are also shown in brackets, showing close agreement with our approximation. We see that the throughput of the shufflenet with only one buffer achieves more than 70% of the throughput corresponding to the store-and-forward case, as shown in the rightmost column of Table II.

TABLE II
APPROXIMATE MAXIMUM THROUGHPUT OF THE $(2, k)$ SHUFFLENET WITH ONE BUFFER ($n_1 = 16$) FOR DIFFERENT VALUES OF k. SIMULATION VALUES ARE SHOWN IN BRACKETS

k	N	$P^{(1)}_{def,1}$(Eq.(19))	$\hat{\lambda}_1$	$\hat{\lambda}_1/\hat{\lambda}_\infty$
3	24	0.05	0.5151	0.8942
4	64	0.0479 (0.042)	0.3608 (0.34)	0.8606
5	160	0.0464	0.27	0.8325
6	384	0.0453 (0.041)	0.2121 (0.20)	0.8072
7	896	0.0443	0.1725	0.7836
8	2048	0.0435	0.1442	0.7613
9	4608	0.0428	0.1228	0.74
10	10240	0.0422	0.1063	0.7195

We finally obtain an approximate expression for $\hat{\lambda}_B/\hat{\lambda}_\infty$ in terms of k and $B \ge 1$. Let $E_B[H]$ be the average number of hops of a packet given buffer size B. Expanding $E_B[H]$ around $P_{\rm def} = 0$ yields $E_B[H] \approx a + bP^{(1)}_{\rm def,\,B}$, where $a = k(2 - 3\cdot 2^k + 3\cdot 2^k k)/(2(-1 + 2^k k)) \approx 3(k-1)/2$, and $b = k(2 - 2\cdot 2^k + k + 2^k k^2)/(-1 + 2^k k) \approx k^2$. Recalling that $\hat{\lambda}_B \simeq 2\alpha$, we have $\hat{\lambda}_B \approx 2/(a + bP^{(1)}_{\rm def,\,B}) \approx \hat{\lambda}_\infty - (2b/a^2)P^{(1)}_{\rm def,\,B}$. From (20), we have $P^{(1)}_{\rm def,\,B} \approx (2/3)/n_B$ (because $n_B \gg k/9$). Therefore

$$\hat{\lambda}_B \approx \hat{\lambda}_\infty - \frac{16k^2}{27(k-1)^2 n_B}. \tag{21}$$

Using $\hat{\lambda}_\infty \simeq 4/(3(k-1))$ (13), we have

$$\frac{\hat{\lambda}_B}{\hat{\lambda}_\infty} \approx 1 - \frac{4k^2}{9(k-1)n_B}. \tag{22}$$

We plot in Fig. 14 the above expression of $\hat{\lambda}_B/\hat{\lambda}_\infty$ versus k for $B = 1$, 2, and 4. We see that $\hat{\lambda}_B/\hat{\lambda}_\infty$ decreases with k. Using the class-based scheduling policy, a shufflenet with only

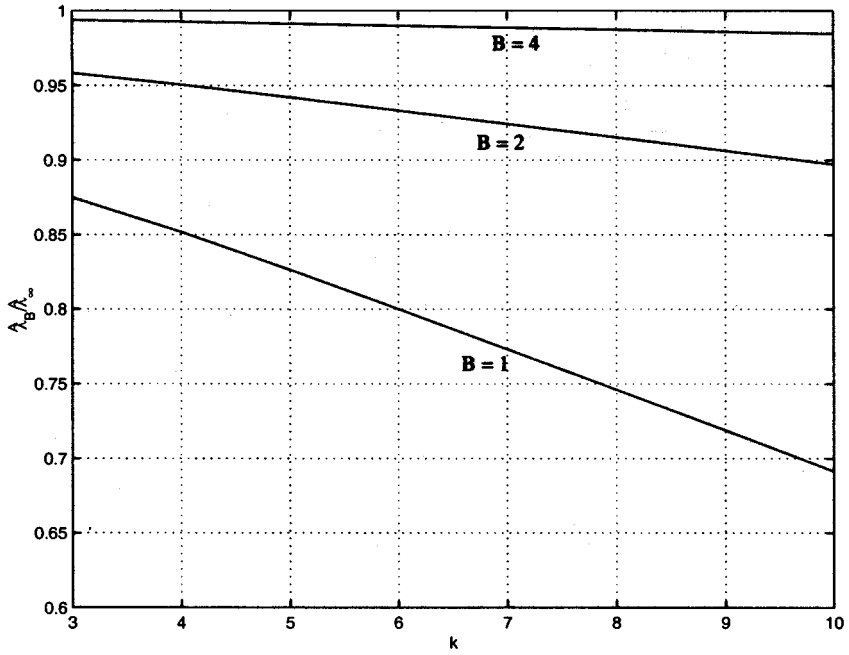

Fig. 14. $\hat{\lambda}_B/\hat{\lambda}_\infty$ versus k in the $(2, k)$ shufflenet.

one buffer can achieve performance close to store-and-forward performance, and with four buffers it achieves throughput comparable with the store-and-forward case, i.e., $\hat{\lambda}_B/\hat{\lambda}_\infty \approx 1$.

V. Conclusion

In a multihop network, packets go through multiple hops before they are absorbed. In order to reach its destination with the minimum number of hops, a packet at a node may have a preferential output channel to use (the so-called "care" packet) or does not have such preference (the so-called "don't care" packet). Deflection routing can be used whenever packets contend for the same output at a node which runs out of buffer. Since available optical buffers may be limited, a good scheduling algorithm is important in the network performance. In this paper, we have studied packet scheduling algorithms and their performance in a buffered regular network using deflection routing. Using shufflenet as our example, we have shown that class-based scheduling, in which "care" packets are scheduled at a higher priority than "don't care" packets, can achieve substantial performance improvement (in terms of throughput and delay) compared with its nonpriority counterpart. Our results suggest that scheduling packets strictly in a first-come-first-served manner regardless of whether they are "care" or "don't care" is not efficient.

In a shufflenet with deflection routing, there has not been enough study to show explicitly how the performance may scale as the buffer size per node and the network size increase. We have analyzed how the performance scales using the class-based scheduling. Using the symmetric property of the shufflenet, the state space in analyzing a shufflenet can be greatly reduced and the trajectory of an arbitrarily chosen packet in the network can be modeled as a discrete time Markov chain. With this model, important network performance measures (such as throughput and delay) can be analytically derived, once the deflection probability of a packet in the network is known. The performance analysis of shufflenet is hence reduced to finding such probability with respect to the offered load and the scheduling algorithm. Previous studies generally obtained the deflection probability by solving numerically a transcendental equation which becomes complicated as the buffer size increases beyond one. We have obtained a simple closed-form approximation of the deflection probability. The expression, validated with our simulations, greatly simplifies the analysis of the shufflenet and allows us to extract the performance trend of a shufflenet with respect to the buffer and network sizes. The deflection probability decreases very quickly with the buffer size B in each node (as $O(B^{-1}2^{-B})$), accounting for the substantial performance improvement in the shufflenet as the buffer size increases. A shufflenet with one buffer per node can indeed achieve impressive throughput and with the buffer size as low as four packets per node, throughput close to the store-and-forward casecan be achieved.

Acknowledgment

The authors wish to thank Dr. A. Bononi for introducing the problem to us and for providing valuable discussions and comments.

References

[1] B. Y. Yu, I. Glesk, and P. R. Prucnal, "Analysis of a dual-receiver node with high fault tolerance for ultrafast OTDM packet-switched shuffle networks," *J. Lightwave Technol.*, vol. 16, pp. 736–744, May 1998.

[2] I. Chlamtac, A. Fumagalli, and C.-J. Suh, "Switching multi-buffer delay lines for contention resolution in all-optical deflection networks," in *Proc. 1996 GLOBECOM*, London, U.K., Nov. 1996, pp. 1624–1628.

[3] I. Chlamtac and A. Fumagalli, "Quadro-star: A high performance optical WDM star network," *IEEE Trans. Commun.*, vol. 42, pp. 2582–2591, Aug. 1994.

[4] A. Acampora, "A multichannel multihop local lightwave networks," in *Proc. 1987 IEEE GLOBECOM*, Nov. 1987, pp. 1459–1467.

[5] M. Karol and R. Gitlin, "High-performance optical local and metropolitan area networks: Enhancement of FDDI and IEEE 802.6 DQDB," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 1439–1448, Oct. 1990.
[6] A. Acampora and M. Karol, "An overview of lightwave packet networks," *IEEE Network*, vol. 3, pp. 29–41, Jan. 1989.
[7] R. D. Gitlin and T. B. London, "Broadband Gigabit research and the LuckyNet testbed," *J. High Speed Networks*, vol. 1, no. 1, pp. 1–47, 1992.
[8] J. J. Yoo, J. E. Leight, C. Kim, G. Giaretta, W. Yuen, A. E. Willner, and C. J. Chang-Hasnain, "Experimental demonstration of a multihop shuffle network using WDM multiple-plane optical interconnection with VCSEL and MQW/DBR detector arrays," *IEEE Photon. Technol. Lett.*, vol. 10, pp. 1507–1509, Oct. 1998.
[9] 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 network," *IEEE Trans. Commun.*, vol. 43, pp. 88–98, Jan. 1995.
[10] A. Bononi, G. A. Castañòn, and O. K. Tonguz, "Analysis of hot-potato optical networks with wavelength conversion," *J. Lightwave Technol.*, vol. 17, pp. 525–534, April 1999.
[11] A. Bononi and P. R. Prucnal, "Analytical evaluation of improved access techniques in deflection routing networks," *IEEE/ACM Trans. Networking*, vol. 4, pp. 726–730, Oct. 1996.
[12] A. Acampora and S. Shah, "Multihop lightwave networks: A comparison of store-and-forward and hot-potato routing," *IEEE Trans. Commun.*, vol. 40, pp. 1082–1090, June 1992.
[13] S.-H. G. Chan and H. Kobayashi, "Performance analysis of shufflenet with deflection routing," in *Proc. 1993 IEEE GLOBECOM*, TX, Dec. 1993, pp. 854–859.
[14] A. Choudhury and V. Li, "An approximate analysis of the performance of deflection routing in regular networks," *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 1302–1316, October 1993.
[15] S.-H. G. Chan and H. Kobayashi, "Buffer architectures and routing algorithms in the performance of shufflenet," in *Proc. 1993 IEEE SICON/ICIE*, Singapore, Sept. 1993, pp. 34–38.
[16] Z. Zhang and A. Acampora, "Performance analysis of multihop lightwave networks with hot potato routing and distance-age priorities," in *Proc.IEEE INFOCOM*, FL, April 1991, pp. 1012–1021.
[17] L. Wang and K.-W. Hung, "Contention resolution in the loop-augmented shufflenet multihop lightwave network," in *Proc. 1994 IEEE GLOBECOM*, CA, Dec. 1994, pp. 186–190.
[18] S. P. Monacos and A. A. Sawchuk, "A scalable recirculating shuffle network with deflection routing," in *Proc.3rd Int. Conf. Massively Parallel Processing Using Optical Interconnections*, HI, Oct. 1996, pp. 122–129.
[19] S.-W. Seo, P. R. Prucnal, and H. Kobayashi, "Generalized multihop shuffle networks," *IEEE Trans. Commun.*, vol. 44, pp. 1205–1211, Sept. 1996.
[20] C.-L. Ng, S.-W. Seo, and H. Kobayashi, "Performance analysis of generalized multihop shuffle networks," in *Proc. INFOCOM'97*, Kobe, Japan, April 1997, pp. 824–829.
[21] P. Palnati, E. Leonardi, and M. Gerla, "Bidirectional shufflenet: A multihop topology for backpressure flow control," in *Proc. 4th Int. Conf. Comput. Commun. Networks*, NV, Sept. 1995, pp. 74–81.
[22] M. Gerla, E. Leonardi, F. Neri, and P. Palnati, "Minimum distance routing in the bidirectional shufflenet," in *Proc. IEEE INFOCOM'98*, CA, Mar. 1998, pp. 102–109.
[23] N. F. Maxemchuk, "Routing in the Manhattan street network," *IEEE Trans. Commun.*, vol. 35, pp. 503–512, May 1987.
[24] N. F. Maxemchuk and R. Krishnam, "A comparison of linear and mesh topologies—DQDB and the Manhattan street network," *IEEE J. Select. Areas Commun.*, vol. 11, pp. 1278–1289, Oct. 1993.
[25] A. G. Greenberg and J. Goodman, "Sharp approximation models of deflection routing in mesh networks," *IEEE Trans. Commun.*, vol. 41, pp. 210–223, Jan. 1993.
[26] K. L. Yeung and T.-S. P. Yum, "Node placement optimization in shufflenets," *IEEE/ACM Trans. Networking*, vol. 6, pp. 319–324, June 1998.
[27] S. Banerjee and B. Mukherjee, "Algorithms for optimized node arrangements in shufflenet based multihop lightwave networks," in *Proc. IEEE INFOCOM'93*, CA, USA, Mar. 1993, pp. 557–564.
[28] P. To, T.-S. P. Yum, and Y.-W. Leung, "Multistar implementation of expandable shufflenets," *IEEE/ACM Trans. Networking*, vol. 2, pp. 345–351, Aug. 1994.
[29] J. Iness, S. Banerjee, and B. Mukherjee, "GEMNET: A generalized, shuffle-exchange-based, regular, scalable, modular, multihop, WDM lightwave network," *IEEE/ACM Trans. Networking*, vol. 3, pp. 470–476, Aug. 1995.
[30] G. B. Brewster and M. S. Borella, "Multicast routing algorithms for the WDM shufflenet local optical network," in *Proc. IEEE ICC*, P.Q., Canada, June 1997, pp. 111–115.
[31] S.-H. G. Chan and H. Kobayashi, "Asymptotic performance of a buffered shufflenet with deflection routing," in *Proc. 1994 IEEE GLOBECOM*, CA, Dec. 1994, pp. 1935–1942.

S.-H. Gary Chan (M'91) received the Ph.D. degree in electrical engineering (with a minor in business administration) from Stanford University, Stanford, CA, in January 1999 and the B.S.E. degree in electrical engineering from Princeton University, Princeton, NJ, in June 1993.

He is currently an Assistant Professor in the Department of Computer Science at the Hong Kong University of Science and Technology, Hong Kong. Prior to this, he was a Visiting Assistant Professor in networking at the University of California, Davis, for a year.

From 1993 to 1994, he was a William and Leila Fellow at Stanford University. From 1992 to 1993, he was a research intern at the NEC Research Institute, Princeton, NJ. His research interests include multimedia networks, services, and systems; high-speed communications networks; and network protocols.

Dr. Chan is a member of Tau Beta Pi, Sigma Xi, and Phi Beta Kappa. At Princeton University, he was the recipient of the Charles Ira Young Memorial Tablet and Medal, and the POEM Newport Award of Excellence in 1993.

Hisashi Kobayashi (F'77) received the B.S.E. and the M.S.E. degrees from the University of Tokyo, Japan, in 1961 and 1963, respectively, and the Ph.D. degree from Princeton University, Princeton, NJ, in 1967, all in electrical engineering.

From 1963 to 1965, he was a Radar Engineer at Toshiba Electric Co., Kawasaki, Japan, prior to coming to Princeton University as an Orson Desaix Munn Fellow. From 1967 to 1982, he was with IBM T. J. Watson Research Center, Yorktown Heights, NY, where he worked on data transmission, digital magnetic recording, performance modeling of computers and communication systems, and queueing network theory. He is the inventor (1971) of a high-density magnetic recording method, now widely known as PRML (partial-response, maximum-likelihood decoding) scheme, a co-inventor (1974) of *relative address coding* for image compression. He also developed the convolutional algorithm (1975) for a multiclass queueing network. He served as Senior Manager of Systems Analysis and Algorithms (1974–1980), and Department Manager of VLSI Design (1981–1982). From 1982 to 1986, he was the Founding Director of the IBM Tokyo Research Laboratory. In 1986, he joined Princeton University as Dean of Engineering and Applied Science (1986–1991), and as the Sherman Fairchild University Professor of Electrical Engineering and Computer Science. His current research interest includes: transceiver design for wireless communications; mobility and traffic modeling for wireless Internet services; optical network architectures and protocols; and teletraffic theory. He is the author of *Modeling and Analysis: An Introduction to System Performance Evaluation Methodology* (Reading, MA: Addison-Wesley 1978), and has authored more than 120 technical papers.

Dr. Kobayashi is the recipient of the Humboldt Prize from Germany (1979), IFIP's Silver Core Award (1980); IBM Outstanding Contribut ion Awards (1975, 1984), and IBM Invention Achievement Awards (1971 and 1973). He has been a member of the Engineering Academy of Japan and a Governor of ICCC since 1992. In the fall of 1998, he was a Visiting Fellow of British Columbia's Advanced Systems Institute (BC-ASI) at the University of Victoria, B.C., Canada.