Analysis of One-Buffer Deflection Routing in Ultra-Fast Optical Mesh Networks

A. Bononi, F. Forghieri, and P. R. Prucnal
Department of Electrical Engineering
Princeton University
Princeton, NJ, 08544

Abstract

The steady state behavior of regular two-connected multihop networks in homogeneous load under hot-potato and single-buffer deflection routing is analyzed for ultra-fast optical applications. Manhattan Street Network and ShuffleNet are compared in terms of throughput, delay and deflection probability both analytically and by simulation. It is analytically verified that single-buffer deflection routing recovers in both networks more than 60% of the throughput loss of hot-potato with respect to store-and-forward when packets are generated with independent destinations. This gain, however, decreases to below 40% when the average message length exceeds 20 packets.

1 Introduction

Multihop packet-switching networks with regular two-connected mesh topologies, such as Manhattan Street Network (MS) [1] and ShuffleNet (SN) [2], have been proposed for all-optical implementation at very high bit rates [3], [4]. While in conventional electronic networks buffering of hopping packets at intermediate nodes is commonly used with conventional store-and-forward routing, the same is not true of all-optical networks, where the only fast access optical memories available are simple recirculating fiber delay loops which require optical amplification, thus becoming impractical. Deflection routing [5], with its inherent limited-time buffering, can eliminate the need of optical amplifiers in the optical memory [6]. Even more dramatic simplification is obtained with Hot-Potato [7], which is a special case of deflection routing where buffers are not provided at all. This paper analyzes the steady state behavior of two-connected mesh networks under deflection routing. The one-packet analytical model appearing in [8], [9] for hot-potato routing is reviewed and extended to the single-buffer memory configuration proposed in [6], which is particularly attractive for optical implementation. Simulation results are provided to confirm the validity of the analytical models and stress the consequences of violating some of the underlying assumptions. Section 2 reviews deflection routing and describes node operation. Section 3 provides a detailed analysis of the steady state behavior of two-connected regular mesh networks under both hot-potato and single-buffer deflection routing. The single-buffer optical memory is described and a simplified control algorithm is introduced by neglecting the buffering delay. In Section 4, analytical results for MS and SN are discussed and checked against simulation results. These two topologies are compared for 64 node and 400 node sizes and the improvement achievable with single-buffer deflection routing with respect to hot-potato is evaluated. The degradation on the achievable improvement caused by transmission of long streams of consecutive packets from the same node to a fixed destination is evaluated by simulation.

2 Deflection routing and network operation

A two-connected network is one in which each node has two input links and two output links. In this paper the behavior of two-connected networks using deflection routing is investigated. Deflection routing [5] is a shortest path routing algorithm where buffer overflow is handled without discarding packets. Assume a first-in-first-out (FIFO) buffer with N_b one-packet memory elements is provided on each output link. Routing and buffering proceed as in store-and-forward up to the time where one of the two queues overflows. At that point the overflowing packet is deflected onto the other queue. This is possible since the two queues cannot be full at the same time. Actually, only one shared output queue turns out to be enough [10]. Deflection routing is thus a variation on store-and-forward where no packet loss occurs and the queueing delay remains bounded by the number N_b of memory elements. When buffering is not provided at all, the routing is called Hot-potato [7]. Here when contention occurs one of the two packets, chosen randomly or by low priority, is deflected onto the other link instead of being assigned to its desired output. It will have a chance to find its way at the next node. Since this work is motivated by very high bit rate all-optical applications of deflection routing, only the cases of a single buffer and no buffers will be analyzed.

Consider now a two-connected regular mesh network, such as the 16 node MS (MS16) or the 8 node SN (SN8)

0743-166X/93 $03.00

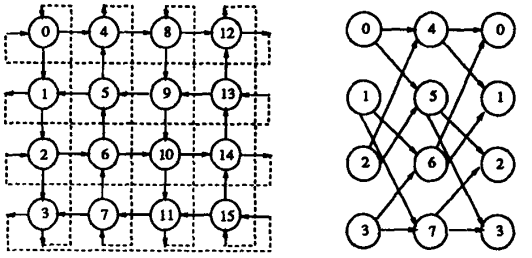

Figure 1: *16 node Manhattan Street network and 8 node ShuffleNet.*

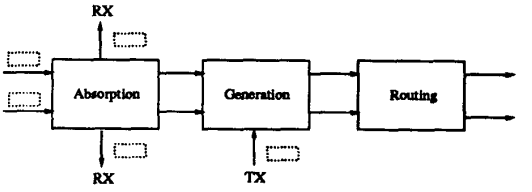

Figure 2: *Logical node structure.*

shown in Fig. 1. A common clock is distributed to all nodes, so that node operations are performed in fixed length time slots, and the time axis is discrete. The logical structure of each node is shown in Fig. 2.

During each slot, each node performs the following operations:

1) *absorption* - Incoming packets destined to the node are absorbed. It is assumed that absorption can be performed on both links at the same time.
2) *generation* - If a new packet is ready for transmission, and if after the absorption block at least one of the two links is free, the new packet is inserted for transmission. It is assumed that only one new packet can be inserted per slot at the node.
3) *routing* - Transiting and locally generated packets are routed to the output links or possibly buffered when buffering is provided.

Note that the slotted system allows polite access to the network. A new packet is not inserted if the input links are occupied by transiting packets. This provides an automatic form of flow control.

3 Steady-state analysis

The steady state behavior of a two-connected network under deflection routing will now be analysed. Assume the network has N nodes, so that the total number of links is $2N$. New arrivals at each node are collected in input buffers, waiting to be injected in the network. Arrivals are assumed to occur at the same rate and independently at each node. It is assumed that at each node the destination of new packets is chosen independently of other nodes and independently of previously admitted packets, and is drawn from a distribution that is uniform on all other nodes. The reasoning behind these assumptions is that this destination pattern helps the routing algorithm share the load evenly among all links. With this traffic homogeneity assumption the input queues are evenly served. Let g be the probability, equal for all nodes, that the input buffer has at least one queued packet per slot. Thus g is the probability that a new packet at each node is ready for transmission at every clock. It will be referred to as the generation probability per slot. Let λ be the network throughput, that is the average number of packets inserted/absorbed per slot in the network at equilibrium. During a time slot a transmitted packet propagates in a connecting link over a distance which will be called the spatial length of a slot. Define W as the ratio of the link length to the spatial length of a slot, i.e. the number of slots in flight on each link at any time. All links are assumed to have same length, and W is assumed to be an integer number, which means that the propagation delay on each link is an exact multiple of the slot time. In optical links, W is given by

$$W = \frac{\ell}{c/n}\frac{R}{M} \approx 10\, R[Gb/s]\,\ell[km]$$

where ℓ is the link length, c/n the light speed in optical fibers of refraction index $n = 1.5$, R is the bit rate and M is the packet size, and the numerical value is obtained for the Asynchronous Transfer Mode (ATM) packet size of 424 bits. In very high bit rate optical networks this ratio W is very high and the propagation delay dominates the queueing delay at intermediate nodes when W is much larger than the buffer size. At any clock time the network links contain $2NW$ slots. Let u be the probability that a spatial slot is occupied by a packet. By the balanced load assumption, u is the same for every slot in the network. Applying Little's theorem [11] to the whole network, the following balance equation is obtained

$$2NWu = \lambda D_s$$

where D_s is the average propagation delay in number of slots and $2NWu$ the average number of packets in the links at any time at equilibrium. If D indicates the average number of hops, then $D_s = WD$ and Little's formula is simply

$$2Nu = \lambda D\,. \qquad (1)$$

The total delay of a packet, once injected in the network, is the sum of the propagation delay D_s and of the queueing delay D_q. For very high bit rate optical networks D_q is small compared to D_s and can thus be neglected. The probability of packet absorption per slot on a given input link at a node is

$$a \triangleq \frac{\left[\begin{array}{l}\text{average number of absorbed}\\ \text{packets per input link per slot}\end{array}\right]}{\left[\begin{array}{l}\text{average number of packets per}\\ \text{input link per slot}\end{array}\right]} = \frac{\lambda/2N}{u} = \frac{1}{D} \qquad (2)$$

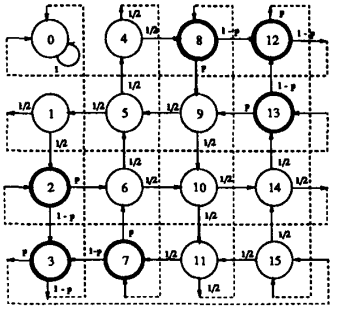

Figure 3: *State transition diagram for MS16.*

where the last equality is obtained from (1).

To get a steady state equation for the slot occupancy probability u, the approximation that packet arrivals at the two input links at every node are independent events will be introduced. This is a reasonable assumption in homogeneous traffic. The average number of newly transmitted packets per node is obtained as the probability of having a new packet times the probability that at least one of the two inputs is free

$$\frac{\lambda}{N} = g\left[1 - u^2(1-a)^2\right] \quad . \tag{3}$$

Equations (1), (2) and (3) yield

$$u = \frac{\sqrt{a^2 + g^2(1-a)^2} - a}{g(1-a)^2} \quad . \tag{4}$$

Note that even for $g = 1$ the value of u is less than one. The reason is that two packets per slot can be received, but only one new packet can be inserted.

The expected number of hops D noticeably depends on the routing algorithm. For store-and-forward with infinite buffers D is a minimum, since packets always take the shortest path to destination, and is independent of the link load u. Therefore by (1) the throughput is a maximum for a given u. However the queueing delay D_q can diverge to infinity when the network approaches saturation, that is when g tends to one. For deflection routing the queueing delay remains bounded, but packets may be deflected to non-optimal paths and thus D becomes an increasing function of u. The throughput is thus lower than with store-and-forward.

3.1 Evaluation of the expected number of hops

To solve for D at steady state under deflection routing, a reference node is now chosen. By the assumed regularity of the network this choice is arbitrary and node zero will be chosen. The trajectory of a test packet generated uniformly at random among all other nodes in the network and destined to node zero will be followed [8]. Because of the homogeneity of the load, the independence approximation and the fact that the routing is memoryless, the random walk of the test packet towards node zero can be modeled as a homogeneous absorbing Markov chain $n(k)$, representing the node visited by the test packet at the end of its k-th hop. The state transition diagram of the chain for MS16 is drawn in Fig. 3. For a given destination node, all nodes whose output links are both on a shortest path to destination are *don't care* for a packet with that destination. The other nodes are *care* nodes. Packets at a care (don't care) node for their destination will be referred to as care (don't care) packets. Care nodes for the test packet are marked by bold circles in the figure. The transition probabilities at a don't care node are both 1/2. In fact, in the assumption of uniform distribution of destinations, a care packet entering a node together with the test packet will prefer either output with probability 1/2, and randomization is applied when both packets are don't care. At a care node, define p as the probability that the test packet is deflected, so that the transition probability on the preferred branch is $(1-p)$. Note that zero is an absorbing state, in that once entered it is never left. Let $\Pi = \{\pi_{ij}\}$, $i,j = 0..N-1$ be the NxN matrix of transition probabilities. The labels on the branches of Fig. 3 show these transition probabilities. Each element π_{ij} represents the probability that the test packet will move to node i at its $(k+1)$-th hop, being at node j after the k-th hop. In uniform traffic, this matrix is independent of the hop number k, except for the first hop, as explained in the Appendix.

Let $\mathbf{p}(k)$ be the state vector at time k, whose elements $p_i(k)$ represent the probability that the test packet will arrive at node i at its k-th hop. Given the distribution $\mathbf{p}(k)$ at time k, the state at time $k+1$ is given by

$$\mathbf{p}(k+1) = \Pi\,\mathbf{p}(k) \quad . \tag{5}$$

The state $[1\ 0\ \ldots\ 0]^T$ is the solution to which the chain converges as $k \to \infty$, and in fact it is the eigenvector associated with the eigenvalue $\mu = 1$ of the Markov matrix Π. To interpret the information given by the time evolution of the state vector, define

$$I_i(k) \triangleq \begin{cases} 1 & \text{if test packet is at node } i \text{ at time } k \\ 0 & \text{else} \end{cases} \quad .$$

Thus $\{I_i(k); k = 0,1,2\ldots\}$ is a stochastic process representing the passage of the test packet through node i. The mean of this process is

$$\mathrm{E}I_i(k) = p_i(k) \qquad k = 0,1,2\ldots\ .$$

Now define the random variable V_i as the number of times the test packet visits node i in its travel towards node zero

$$V_i \triangleq \sum_{k=0}^{\infty} I_i(k) \qquad i = 1,2,\ldots N-1.$$

Note that when the packet arrives at node zero, it remains there forever, so that $I_0(k)$ is a step function jumping from

zero to one at the random arrival hop d of the packet. Therefore

$$\sum_{k=1}^{\infty} k\,[I_0(k) - I_0(k-1)] = \sum_{k=1}^{\infty} k\delta(k-d) = d$$

where $\delta(k)$ is unity at $k = 0$ and zero otherwise. The random variable d represents the total number of hops taken by the test packet in its travel. The expected values of these random variables have interesting interpretations:

$$EV_i = \sum_{k=0}^{\infty} p_i(k) = \begin{bmatrix}\text{avg. \# of times}\\ \text{the test packet}\\ \text{visits node } i.\end{bmatrix} \quad , i = 1, 2, \ldots N-1 \tag{6}$$

$$Ed = \sum_{k=1}^{\infty} k\,[p_0(k) - p_0(k-1)] = \begin{bmatrix}\text{avg. \# of hops}\\ \text{of the test}\\ \text{packet.}\end{bmatrix} \triangleq D \quad . \tag{7}$$

From this, $p_0(k)$ is seen to be the cumulative distribution function of the random number of hops d taken by the test packet. One more observation. Indicating by $\mathcal{DC}$ the set of don't care nodes, the random variable

$$V_{dc} \triangleq \sum_{i \in \mathcal{DC}} V_i$$

is the total number of times the test packet visits a don't care node in the experiment. Now, d is also the number of times that nodes not coinciding with the destination node are visited in the experiment

$$d = \sum_{i=1}^{N-1} V_i \quad . \tag{8}$$

Hence the long-run fraction of time the test packet is at a don't care node, referred to as the don't care probability P_{dc}, is

$$P_{dc} = \frac{\sum_{i \in \mathcal{DC}} EV_i}{\sum_{i=1}^{N-1} EV_i} = \frac{EV_{dc}}{Ed} = \frac{\sum_{k=0}^{\infty} \sum_{i \in \mathcal{DC}} p_i(k)}{D} \quad . \tag{9}$$

where the last expression on the RHS is the operative formula. Since in a homogeneously loaded network the test packet is a typical packet, P_{dc} represents also the probability that a packet entering a node together with the test packet is in a don't care state. The quantities P_{dc}, a, and u all depend on p through the transition matrix Π. On the other hand, it will be shown in the next two sections that the deflection probability p can be derived as a function of the above three quantities, and of the generation probability g

$$p = f\,(P_{dc}(p), a(p), u(p), g)\,. \tag{10}$$

This nonlinear equation in p can be solved recursively [8] using a fixed point algorithm for a given value of g and initial state vector $\mathbf{p}(0) = [0, \frac{1}{N-1}, \ldots, \frac{1}{N-1}]^T$ to preserve load balance. Note that, in homogeneous traffic, p is the deflection probability of any packet at a care node. The long-run fraction of deflections in the network is thus obtained by the law of total probability as

$$p_{net} = p\,(1 - P_{dc}(p)) \tag{11}$$

since a packet is never deflected at a don't care node by definition. p_{net} will be referred to as network deflection probability. In the next two sections, equation (10) will be obtained in the case of no output buffers and for a single output buffer.

3.2 No buffers: hot-potato

The deflection probability for the case of no buffers can be found as shown in [8]. Refer to Fig. 2, and suppose the test packet entering the node from one of the two input links. A deflection at a care node for the test packet occurs if

i) A packet is present on the other link. By the assumed independence of the two inputs this event has probability $[u(1-a) + uag + (1-u)g]$, since the event occurs if a packet is present at the input link and not absorbed, or is present, absorbed, and a new packet is generated, or the input link is empty but a new packet is generated. Note that, since the event is conditioned on the presence of the test packet, a generation can occur only on the link not occupied by the test packet.

ii) the other packet is care, and the probability of this event is $(1 - P_{dc})$.

iii) the conflicting packet prefers the same output as the test packet, and this occurs with probability 1/2.

iv) the test packet loses the coin flip and the output is assigned to the competing packet. This occurs with probability 1/2.

Hence, by defining the probability of having a care packet on the other link as

$$P_c = [u(1-a) + uag + (1-u)g](1 - P_{dc}) \tag{12}$$

the desired nonlinear equation for p is

$$p = \frac{1}{4} P_c(p) \quad . \tag{13}$$

A different approach must be taken to handle the computation of the deflection probability at the first step of the chain, where the test packet is at its generation node and is trying to access the network. This case is treated in detail in the Appendix. After the first hop, the transition matrix Π becomes time independent and the Markov chain is thus homogeneous.

3.3 One buffer: optical solution

This section will derive equation (10) when use is made of the single-buffer memory proposed in [6], which lends itself to a simple optical implementation.

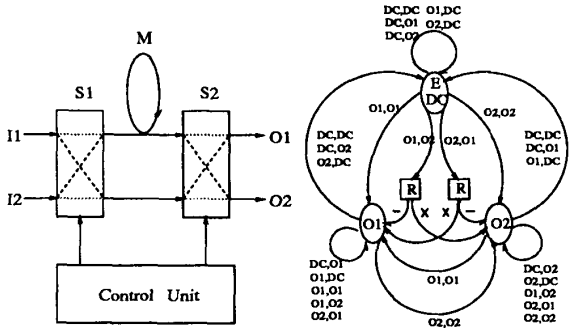

Figure 4: *Left: scheme of the optical implementation of the routing block. M is the delay line memory and S1, S2 are exchange-bypass switches. Right: state diagram of the control unit.*

M	I1	I2	S1	S2	NM
E/DC	E/DC	E/DC	R	R	E/DC
		O1	–	×	E/DC
		O2	–	–	E/DC
	O1	E/DC	×	×	E/DC
		O1	R	×	O1
		O2	R	–/×	O1/O2
	O2	E/DC	×	–	E/DC
		O1	R	×/–	O2/O1
		O2	R	–	O2
O1	E/DC	E/DC	R	–	E/DC
		O1	×	–	O1
		O2	–	–	E/DC
	O1	E/DC	–	–	O1
		O1	R	–	O1
		O2	–	–	O1
	O2	E/DC	×	–	E/DC
		O1	×	–	O1
		O2	R	–	O2
O2	E/DC	E/DC	R	×	E/DC
		O1	–	×	E/DC
		O2	×	×	O2
	O1	E/DC	×	×	E/DC
		O1	R	×	O1
		O2	×	×	O2
	O2	E/DC	–	×	O2
		O1	–	×	O2
		O2	R	×	O2

M State of the memory before switching
I1 State of packet at input I1
I2 State of packet at input I2

S1 State of switch S1
S2 State of switch S2
NM State of the memory after switching

– Bar state
× Cross state
R Random choice between bar and cross states
–/× State of S2 set equal to state of S1
×/– State of S2 set opposite to state of S1

Table 1: *Truth table of the control unit.*

Description

The scheme of the optical switch with memory is shown in Fig. 4. $S1$ and $S2$ are two switches whose cross/bar state is controlled by a control unit and the memory element M is a one-packet fiber delay line. The combined control of the switches allows reducing the probability of packet deflection by selecting which of the two input branches has to be delayed. The objective here is to reduce the probability of deflection, without caring about the fixed one-slot delay introduced on one of the two inputs, since this is negligible with respect to the propagation delay at very high bit rates. This allows treating don't care packets exactly as empty packets as far as routing is concerned, thereby reducing the complexity of the control unit. The control unit knows the state of both inputs, that is whether each input link is empty or contains a packet in a don't care state (EM/DC), or a packet wishing to exit on output 1 (O1), or on output 2 (O2). The state of the memory (EM/DC, O1 or O2) is also known to the control unit. The control unit implements a finite state machine whose states are the possible states of the memory. Table 1 gives the a truth table description of the machine where it is seen that the empty (EM) and don't care (DC) states are collapsed into a single state.

Deflections in this scheme are caused by collisions at the output switch S2. To minimize the number of collisions and thus reduce the probability of packet deflection, switch S1 is set according to the state of the memory. If the memory contains a care packet, and if at one of the two inputs a care packet with the same preferred output is present, this input packet will be stored to avoid a collision. Clearly, if both input links have packets wishing the same output as the memory packet, a deflection will occur. Since statistically it does not make any difference which of the three packets will be deflected, to simplify the control algorithm the packet in memory will be given priority over the two inputs and will set the output switch S2, so that one of the two input packets, chosen at random, will be deflected. On the other hand, if there is no possible conflict between memory and inputs, switch S1 will be set to preferably store a don't care packet to reduce the deflection probability at the next time slot. This *priority memory* strategy well matches the standard deflection routing algorithm implemented with output FIFO queues, as described in [10].

A faithful reproduction of the algorithm in [10] would require one more switch to access the memory loop. This would correctly handle the non-conflict situation where a O1 packet is present on I1 and a O2 packet on I2, or vice versa, being the memory empty or don't care, by bypassing the memory. In the scheme of Fig. 4, in this case, one of the two packets is delayed and thus the memory is occupied with a care packet, which slightly increases the probability of deflection of incoming packets at the next slot.

Analysis

To obtain an expression for the deflection probability for the above one-memory buffer, the further assumption of independence of arrivals in two consecutive slots on the same link, as well as on the other link entering the node, must be made. This assumption might be justified in a

practical realisation of an ultra-fast optical network, where the generation of a new optical packet cannot be done at the optical slot rate [4]. In section 4 the effect of violating this assumption is seen by simulation and compared to the analytical results. Refer to Fig. 2 again, supposing as before that the test packet is entering the node from one of the two input links. It will be deflected if

i) there is a care packet on the other link. This event has probability P_c given in (12).

ii) There is a care packet in memory. This event has probability P_{cm} to be found later.

iii) Both packets prefer the same output as the test packet. This occurs with probability 1/4 by independence and traffic homogeneity.

iv) The test packet loses the coin flip to access the memory. This occurs with probability 1/2.

Thus the desired nonlinear equation for p is

$$p = \frac{1}{8} P_c(p) P_{cm}(p) \quad . \tag{14}$$

To evaluate P_{cm}, the equilibrium probability distribution of the memory state before the test packet arrives at the node is needed. Hence this probability is *not* conditioned on the presence of the test packet, and the link occupancy after the absorption/generation blocks is u on both links. The probability of having a care packet on any link is

$$P_u \triangleq u(1 - P_{dc}) \quad . \tag{15}$$

Let the memory state probabilities be

P_1 = probability of having a O1 packet in memory

P_2 = probability of having a O2 packet in memory

P_e = probability of having an empty/don't care packet in memory.

By symmetry $P_1 = P_2$, and by definition $P_{cm} = P_1 + P_2$, so that $P_{cm} + P_e = 1$. A balance equation for one of these four probabilities is enough to solve for all of them. A packet O1 will be stored if

there is a O1 packet in memory and

the two inputs are care and their states are O1,O1 or O1,O2 or O2,O1

or one input is DC and the other is O1

or there is a O2 packet in memory and the two inputs are care O1,O1

or there is an EM/DC packet in memory and the two inputs are care and they are either O1,O1 or (O1,O2)(O2,O1) in which case the O1 will go in memory upon winning the coin flip.

In formulas

$$\begin{aligned} P_1 \quad = \quad & P_1 \left\{ \tfrac{3}{4} P_u^2 + 2P_u(1 - P_u)\tfrac{1}{2} \right\} + P_2 \left\{ \tfrac{1}{4} P_u^2 \right\} \\ & + P_e \left\{ \tfrac{1}{2} P_u^2 \right\} \quad . \end{aligned} \tag{16}$$

From this equation the value of P_{cm} is obtained

$$P_{cm} = 2P_1 = \frac{P_u^2}{1 - P_u + P_u^2} \quad . \tag{17}$$

4 Results

In this section, results obtained with the analytical method described in the previous section will be presented for MS and SN. These are compared for 64 nodes (MS64 vs SN64) and for a larger size of about 400 nodes (MS400 vs SN384) where the percentage size difference is less than 5%. All curves will compare hot-potato routing, where no buffers are provided, to single-buffer deflection routing, where the buffer is the one described and analysed in section 3.3. Simulation results will be provided to check the validity of the analytical models and to extend results beyond the limits of their applicability. An average description of the networks at steady state is given in terms of throughput λ and expected number of hops D versus the generation probability g. The network deflection probability p_{net} is also monitored. Simulations have been run for MS64 and SN64 according to the method described in [8] to support the analytical results. At every clock cycle, packets are generated independently at every node with probability g and with destinations chosen uniformly among all nodes not coinciding with the packet's source and independently of previous clock cycles. An excellent agreement with all the analytical curves presented and with the results given in [8] for throughput and delay curves in MS64 is observed. Fig. 5 shows throughput and delay versus packet generation probability g.

Throughput curves for store-and-forward with infinite buffers (shortest-path routing) are also provided as a reference. These are readily obtained from Little's formula (1) in which the link load u is evaluated by (4) and the zero-load delay is used. The throughput is higher for SN for all values of g and its gain over MS increases for larger networks. For instance, SN384, although smaller than MS400, has much higher aggregate throughput. Note also that one buffer is enough to fill a substantial portion of the throughput gap between store-and-forward and hot-potato, as already shown by simulation in [10] for MS.

To give a better insight of network behavior, Fig. 6 presents curves of network deflection probability p_{net} as given in equation (11). In the simulations, p_{net} is obtained as an ergodic time average. At every clock cycle t, the number of packets after absorption and generation at all nodes is denoted as $n(t)$. Let $n_{def}(t)$ be the portion of these packets which are deflected at time t. In formulas

$$p_{net} = \frac{\sum_{t=1}^{K} n_{def}(t)}{\sum_{t=1}^{K} n(t)} \tag{18}$$

where K is the total number of clock cycles in the simulation. Actually, the summations were started after the network had reached steady state and the number K of clock cycles after the transient was chosen to be 10,000.

The fact that these simulated quantities match perfectly with the respective analytical values for the test packet confirms that the test packet is actually a "typical" packet, and that the network traffic is really homogeneous, so that the independence approximation at the node is accurate. Fig. 6 shows that for 64 nodes p_{net} in SN is slightly higher than in MS, while for 400 nodes p_{net} is much lower in SN. The reduction in p_{net} in both MS and SN using the single-buffer memory is remarkable. It can be observed that the reduction is never less than 60% in both MS and SN, both for 64 and 400 nodes, and is slightly higher for MS when the load approaches one. Store-and-forward with infinite buffers would achieve a 100% reduction but is not optically implementable. The single-buffer delay line memory is a simple, feasible way to achieve a substantial reduction of the total number of deflections.

As a final result, simulations for MS64 and SN64 are presented to check the effect of correlation between destinations of packets generated at the same node, as is the case when a message of several packets has to be transmitted to the same recipient. In the simulations, the message length M_l was chosen as $M_l = X+1$, where X is a Poisson random variable.
The left two graphs of Fig. 7 show throughput curves for SN and MS for an average value $E(M_l) = 1$, 5, 20. The $E(M_l) = 1$ curves are those already given where no correlation between successive packets exists and match the one-packet model curves. An elaboration of these curves is given in the two graphs on the right, which show the percentage of the throughput gap between store-and-forward and hot-potato recovered by single-buffer deflection routing. The surprising result that a single buffer is enough to get a substantial throughput gain over hot-potato was obtained in the assumption of uncorrelation among packet destinations. The potential 60% gain on throughput gap predicted at full load in the absence of correlation can actually decrease to less than 40% in the presence of long messages because one buffer only cannot efficiently handle successive conflicts arising from streams of consecutive packets with the same destination colliding at the node. More buffers are required in this case to substantially improve network performance.

5 Conclusions

This paper gives a detailed review of the one-packet model used to analyze the steady state behavior of regular multihop networks in homogeneous traffic under hot-potato routing and extends the method to include the analytical treatment of single-buffer deflection routing, which is of interest in very high bit rate all-optical networks. The analyzed buffer is particularly attractive for optical implementation for its simplicity, and the proposed control scheme takes full advantage of the ultra-fast network environment, since it does not distinguish between empty and don't care packets, thereby allowing further reduction of the routing complexity with respect to the original proposal [6], without appreciably affecting the overall delay. The analytical model is applied to MS and SN, which are compared in terms of throughput, delay and deflection probability, and all results are supported by simulations. The average analysis shows that SN has higher throughput than MS at all loads, and the difference increases with network size. The effectiveness of the single buffer is analytically quantified. It is verified that under the assumption of independence of packet destinations, the single-buffer deflection routing recovers more than 60% of the throughput loss of hot-potato with respect to store-and-forward. However, when messages of average length as high as 20 packets are transmitted to the same recipient, consecutive collisions arise and a single buffer cannot efficiently handle them anymore. The achievable gain in this case is reduced to below 40%.

Appendix

The transition matrix Π_0 describing the first hop of the test packet differs from the matrix Π during the rest of the walk only in the entries relative to care nodes. The initial probability p_0 that the test packet be deflected at the injection node will now be found. Refer to Fig. 2, and this time suppose the test packet waiting to access the network at the input of the generation block. At equilibrium, define $\mathcal{A}_0$, $\mathcal{A}_1$ and $\mathcal{A}_2$ as the event of having respectively 0,1 or 2 packets on the node links after the absorption block, whose probabilities are

$$\begin{aligned} P(\mathcal{A}_0) &= (1-u)^2 + 2u(1-u)a + u^2a^2 \\ P(\mathcal{A}_1) &= 2u(1-u)(1-a) + 2u^2a(1-a) \\ P(\mathcal{A}_2) &= u^2(1-a)^2 \end{aligned}$$

and are obtained reasoning as in (12). Since it is assumed that the test packet *is* injected in the network, and this is possible only if at least one link is free, its deflection probability is actually conditioned on the event $\mathcal{A}_0 \cup \mathcal{A}_1$. Its deflection is possible only if event $\mathcal{A}_1$ occurs, i.e. a transiting packet is present. Thus the probability of having a care packet together with the test packet at the injection node is

$$P_{c_0} = \frac{P(\mathcal{A}_1)}{P(\mathcal{A}_1) + P(\mathcal{A}_0)}(1 - P_{dc})$$

and from eq.(13) and (14) the initial deflection probability is

$$\begin{cases} p_0 = \frac{1}{4} P_{c_0} & \text{without buffers} \\ p_0 = \frac{1}{8} P_{c_0} P_{cm} & \text{with single buffer} \end{cases}$$

Therefore equation (5) describing the state vector at time $k+1$ becomes

$$\begin{cases} \mathbf{p}(k+1) = \Pi\, \mathbf{p}(t) & k \geq 1 \\ \mathbf{p}(1) = \Pi_0\, \mathbf{p}(0) \end{cases}$$

Acknowledgments

The authors are grateful to Andrea Fumagalli and Albert Greenberg for providing preprints of their papers [6] and [8].

References

[1] N. F. Maxemchuk, "The Manhattan Street Network," in *Proc. GLOBECOM '85*, pp. 255–261

[2] A. S. Acampora, M. J. Karol, and M. G. Hluchyj, "Terabit lightwave networks: the multihop approach," *AT&T Tech. J.*, vol. 66, pp. 21–34, Nov./Dec. 87.

[3] J. R. Sauer, "An opto-electronic multi-Gb/s packet switching network," preprint, Optoelectronic System Center, University of Colorado, Feb. 1989.

[4] A. Bononi, F. Forghieri, and P. R. Prucnal, "Design and channel constraint analysis of ultra-fast multihop all-optical packet switching networks with deflection routing employing solitons," submitted to *J. Lightwave Tech.*, Sep. 92.

[5] N. F. Maxemchuk, "Regular mesh topologies in local and metropolitan area networks," *AT&T Tech. J.*, vol. 64, no. 7, pp. 1659–1685, Sep. 1985.

[6] I. Chlamtac and A. Fumagalli, "An all-optical switch architecture for manhattan networks," to be published in *IEEE J. Select. Areas in Commun.*, "Gigabit network protocols and applications".

[7] P. Baran, "On distributed communications networks," *IEEE Trans. Commun. Syst.* vol. 12, pp. 1–9, Mar. 1964.

[8] A. G. Greenberg and J. Goodman, "Sharp approximate models of adaptive routing in mesh networks," in *Teletraffic Analysis and Computer Performance Evaluation*. O. J. Boxma, J. W. Cohen, and H. C. Tijms, Eds. Elsevier, Amsterdam, 1986, pp. 255–270. — "Sharp approximate models of deflection routing in mesh networks," to be published in *IEEE Trans. Commun.*.

[9] A. S. Acampora and A. Shah, "Multihop lightwave networks: a comparison of store-and-forward and hot-potato routing," *IEEE Trans. Commun.*, vol. COM-40, pp. 1082–1090, June 1992.

[10] N. F. Maxemchuk, "Comparison of deflection and store-and-forward techniques in the Manhattan Street and Shuffle-Exchange networks," in *Proc. IEEE INFOCOM '89*, pp. 800–809, Apr. 1989.

[11] D. Bertsekas and R. Gallager, *Data networks*. Prentice Hall, 1987.

[12] E. Ayanoglu, "Signal flow graph for path enumeration and deflection routing analysis in multihop networks," in *Proc. IEEE INFOCOM '89* pp. 1022–1029, 1989.

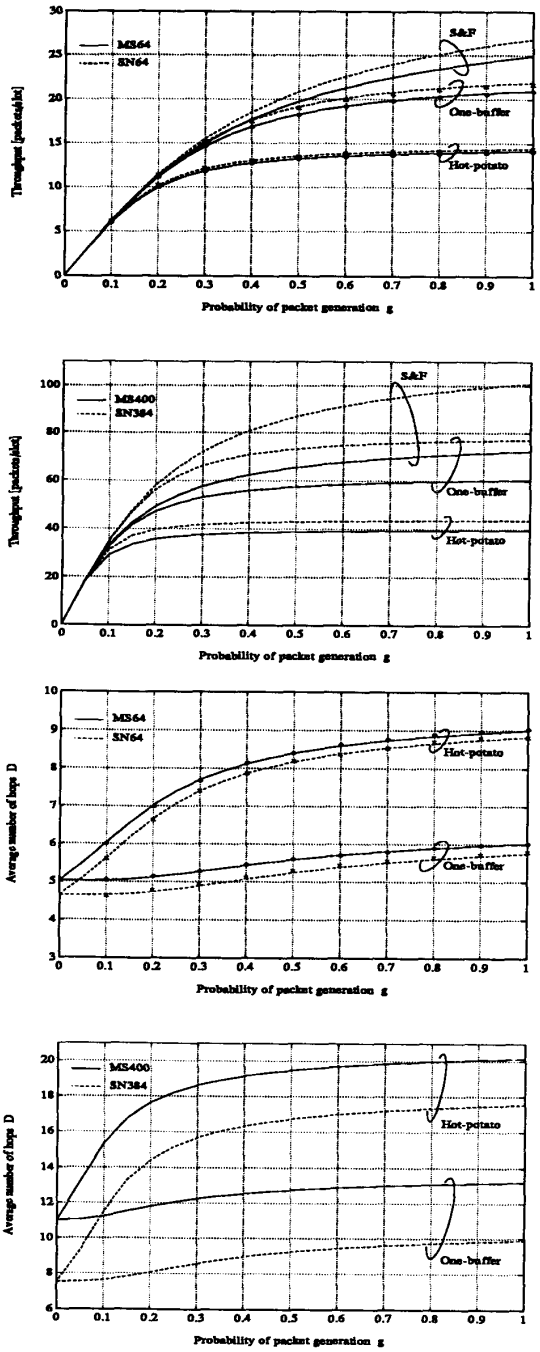

Figure 5: *Aggregate network throughput and delay in MS64, SN64 and MS400, SN384. Curves for store-and-forward with infinite buffers are provided as a reference.*

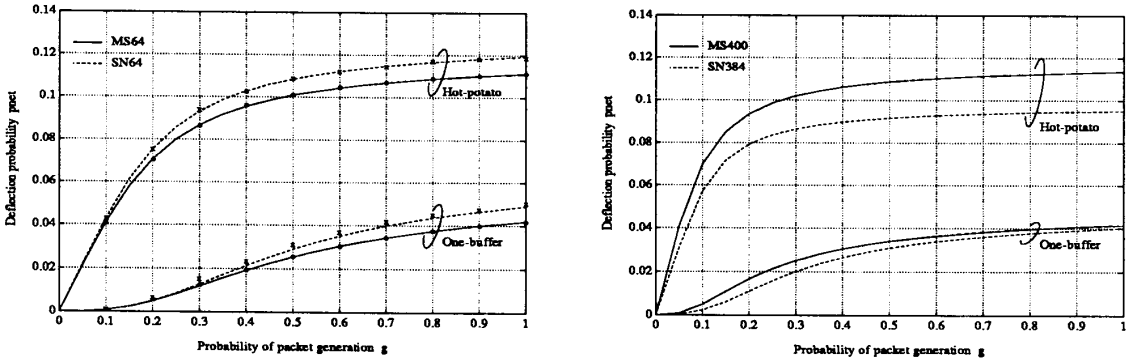

Figure 6: *Network deflection probability in single buffer deflection routing with respect to hot-potato.*

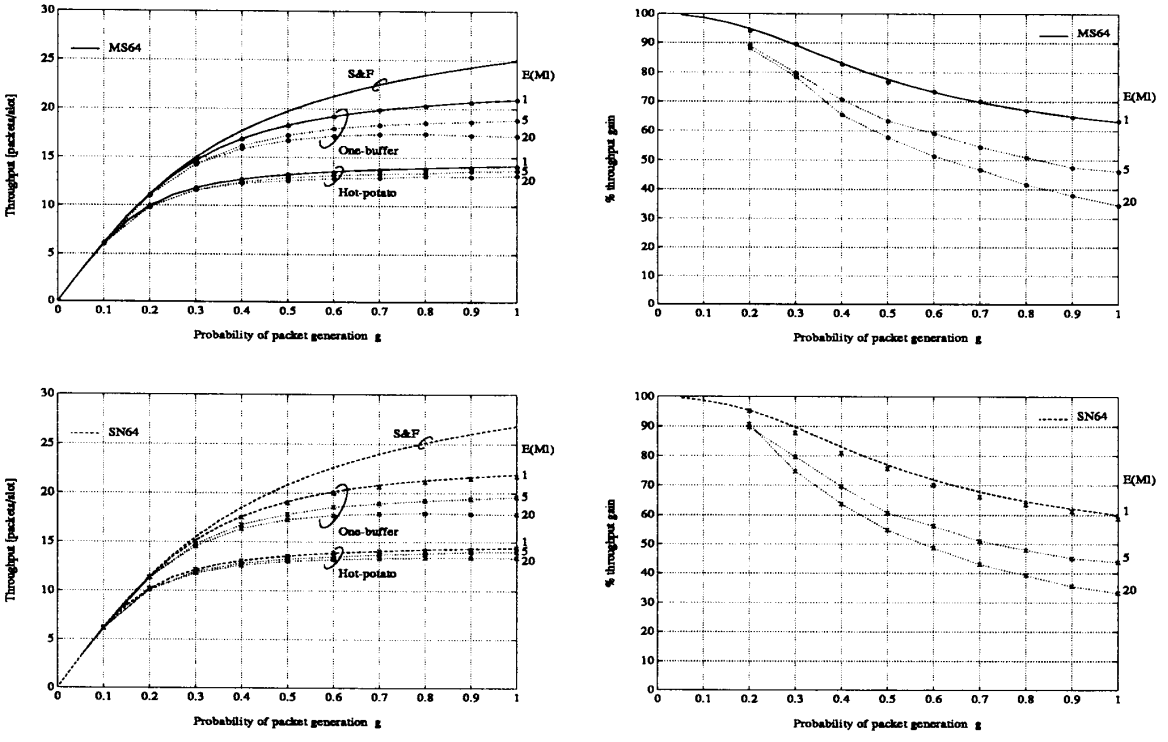

Figure 7: *(Left) Simulation results of aggregate network throughput for MS64 and SN64 for average message length* $EM_l = 5, 20$. *Theoretical curves for* $EM_l = 1$ *are given as a reference. (Right) Throughput difference between one-buffer and hot-potato, expressed in percentage of the throughput difference between store-and-forward and hot-potato.*