Journal of High Speed Networks 5 (1996) 243–258
IOS Press

Novel structures of the optical node in multihop transparent optical networks using deflection routing*

Alberto Bononi**
Department of Electrical and Computer Engineering, S.U.N.Y., Buffalo, Amherst, NY 14260, USA
Tel.: +1 716 645 2422 ext. 2133, Fax: +1 716 645 3656, E-mail: bononi@eng.buffalo.edu

Paul R. Prucnal
Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA
Tel.: +1 609 258 5549, Fax: +1 609 258 2158

Abstract. Novel single-receiver/single-transmitter/single-buffer node structures for ultrafast two-connected multihop transparent optical packet-switching networks with deflection routing are introduced. A Shufflenet topology in uniform traffic is used as a benchmark to compare several shared optical memory schemes and their control algorithms. These simple structures minimize the number of crossbar switches needed at each optical node and have moderate control complexity, while still yielding large throughput and small delay. The minimization of the number of crossbar switches results in an improved optical power budget. An analytical model, obtained by an extension of the existing theory, provides a design tool to search for efficient node control strategies that minimize the number of deflections and packet misses for each constrained node structure.

1. Introduction

Transparent Optical Networks (TONs) have recently become the focus of much research towards high-speed high-capacity multiuser communications [1–3].

The basic idea behind TONs is to modulate the digital information onto a lightwave carrier and let it propagate through the network to its end destination, without intermediate electronic regeneration. Therefore, nodes in multihop TONs do not bear the burden of processing through-going messages not destined to them.

Both circuit-switching and packet-switching have been considered for TONs. Wavelength Division Multiplexing (WDM) and hybrid Space/Wavelength Division Multiplexing (S/WDM) have proven to be attractive for circuit switching applications [2], but present technological limitations on tunable lasers, optical filters and all-optical wavelength shifters make space switching the only viable approach for very high speed packet-switching applications.

This paper considers pure Space Switching (SS) multihop TONs, in which the optical nodes are connected by dedicated optical fibers, and there is time-sharing of a single wide-band wavelength channel within each of these fibers.

*This paper was presented in part at IEEE INFOCOM '94, Toronto, Canada, June 1994.
**Corresponding author.

Switching is performed in space by taking advantage of lithium-niobate ($LiNbO_3$) crossbar space switches, which represent a mature technology and can have reconfiguration times of a fraction of a nanosecond [4] [1].

In SS TONs extremely high data rates per user can be transmitted, having each user the whole fiber bandwidth (actually the optical amplifiers' bandwidth) at its disposal. However, for ultrafast packet-switching, the control unit in each node must be extremely fast and the control algorithms as simple as possible.

As the in/out node degree of multihop SS topologies is increased, network reliability and throughput increase, but node and control complexity increase as well, becoming a bottleneck at ultrahigh bit rates. For this reason, an in/out degree of 2 is considered in this paper.

Early work on the structure of all-optical nodes in SS multihop networks concentrated on the optical processing of the packet header and the related control of the crossbar routing switch [5]. Those schemes readily apply to ring networks, where switching is performed by an add/drop crossbar switch for reception and transmission of local traffic. Extensions to two-connected mesh networks, in which each node has two optical inputs and two optical outputs, were presented in [6] and [7]. The nodes consisted of an add/drop switch at each optical input and a main routing block at the core of the node to perform the output switching function. Nodes were capable of simultaneously absorbing packets from both links, which was obtained by either providing two optical receivers per node or a sufficient number of local-reception, off-line optical buffers.

2×2 crossbar space switches are key elements in the implementation of such optical nodes. Minimization of the number of crossbars at the node is mandatory to maintain low optical power loss for through-going packets and to limit the cost of the node. To this aim, compact optically-integrated structures are desirable. Optical amplification can be provided at the node output, or switch by switch, to compensate for this power loss. However, optical amplifiers introduce noise in proportion to their gain. This noise accumulates from node to node in these non-regenerative networks. At extremely high bit rates this noise imposes severe limits on the geographical span of the network [8].

Fast-access optical buffers can be implemented with fiber delay loops. Store-and-forward (S&F) is not feasible in very high bit rate TONs, due to the limited number of optical buffers that can be added at each node to keep a low power loss and low control complexity. Optical amplification in the memory loop – a noisy and costly process – can be avoided with deflection routing [9] if buffered packets are allowed to recirculate in each memory loop only once [7,10].

This paper presents novel structures of the optical nodes in two-connected multihop Space Switching Transparent Optical Networks in which only one optical receiver (RX) and one optical transmitter (TX) are provided at each node. Extremely simple nodes, with a few crossbar switches, using non-priority deflection routing with only one fiber-loop optical memory are presented, along with their buffer control algorithms. Shufflenet (SN) [11], a well-known multihop topology in the framework of deflection routing, shown in Fig. 1 for 8 nodes, is selected as a benchmark to compare the new structures and their control algorithms in terms of throughput and average number of hops in uniform traffic. Even though more realistic non-uniform traffic models could be used, uniform traffic is selected since a known, simple analytical model [12] can be extended to these structures and provides a fast tool to test and optimize the control algorithms for each selected constrained node structure.

The paper is organized as follows. Section 2 introduces the novel node schemes and Section 3 specifies their control algorithms. The proposed control algorithms trade delay at the node for throughput efficiency, and are thus meaningful at very high signaling rates, where the link propagation delay is much longer than the packet duration. Section 4 details the extensions to the standard analytical model [12] to find the network steady state in uniform traffic. Section 5 presents numerical results and compares the various structures.

[1] Alternatively, space switches can be built as gated matrix switches using semiconductor optical amplifiers (SOA) as amplifying on/off gates. Even this approach provides reconfiguration times below a nanosecond, according to the on/off switching time of the SOAs. This approach will not be discussed in the paper. More information can be found in [4].

2. Novel schemes for the optical node

Each node in Fig. 1 has two input fibers and two output fibers. This section introduces novel schemes for these nodes.

Figure 2(a) shows the node scheme used in [7], adapted for single TX/RX and no buffers. Solid lines indicate optical fibers and dashed lines electronic controls. Each optical input has an add/drop switch for local traffic, and a routing block performs the output switching. Only one add/drop switch at a time is used for TX/RX operations. If the two lines from the TX and those to the RX are wired together, a 3 dB power loss for both the TX and RX signal is incurred. Also, when receiving a packet and simultaneously transmitting one using the same add/drop switch, part of the TX power loops back to the RX together with the incoming packet. This known interference could in principle be canceled out, but in practice two extra off-line switches will be added for the TX and RX to toggle between add/drop switches. The advantage of this parallel configuration is that only one switch in the add/drop block is crossed by each through-going packet. An alternative cascaded configuration is shown in Fig. 2(b). Here there is no splitting of the TX/RX signals. However, two switches are crossed by one input channel, producing an unbalance of the optical power level at the routing switch, which is not a desirable feature. This unbalance can be eliminated by artificially inserting an equalizing loss in the upper branch. Both figures show that the node consists of a 3 × 3 optical switch. A minimum of *three* 2 × 2 switches are needed to form a rearrangeably non-blocking 3 × 3 switch. This structure will be referred to as 3*Shp*, as it implements deflection routing without buffers (*h*ot-*p*otato [9]).

The routing block may contain optical buffers. A simple output shared optical memory for this block, making use of fiber delay loops, has been introduced in [10] and an efficient control scheme for high-speed applications

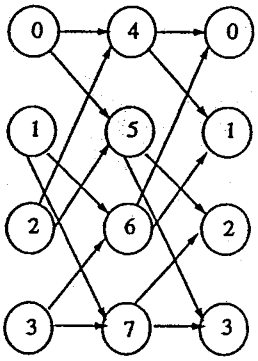
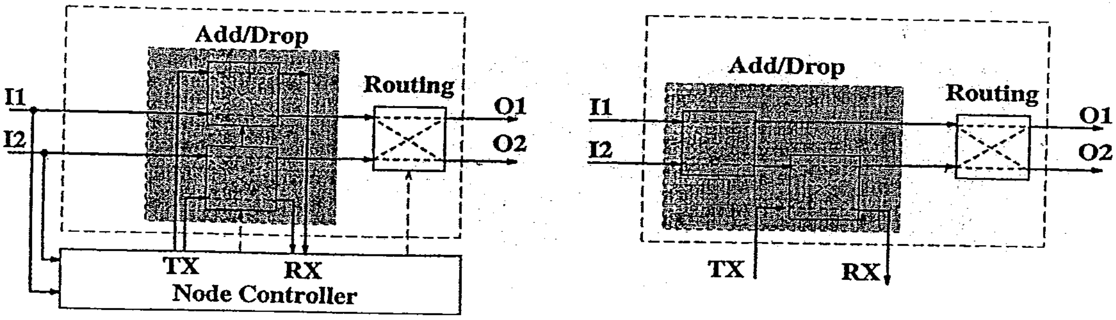


Fig. 1. 8-node Shufflenet (SN).



(a)

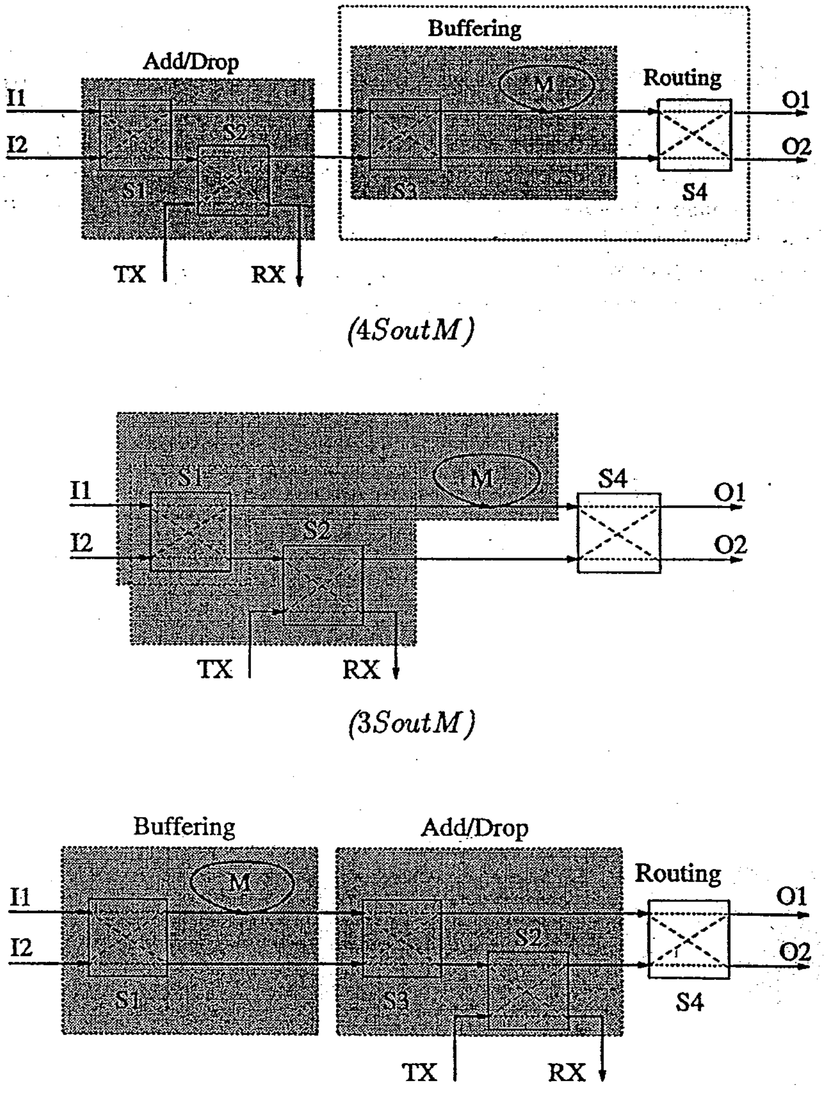
(b)

Fig. 2. 3*Shp* node structure with (a) parallel and (b) serial configuration of the add/drop block.

has been proposed and analyzed in [7]. Similar structures have been studied for optical time-slot interchangers [13,14], and have been proposed for optical ATM packet switches [15–17]. The novelty in our scheme is the integration of the node access block and the routing block.

Figure 3 (top) shows the structure of the node with the above mentioned memory, where a one-packet fiber delay is used as a buffer. The add/drop block can be either of the two shown in Fig. 2. This output shared memory node with four switches will be referred to as 4*SoutM*.

If switch S3 in Fig. 3 (top) is removed, a 3-switch node is obtained in which switch S1 is shared between TX/RX and buffering operations. This is shown in Fig. 3 (center) and will be referred to as 3*SoutM*. Here the add/drop block must take the serial configuration. This is the simplest possible structure for a node with a single-transmitter/single- receiver/single-buffer.



$(4SoutM)$

$(3SoutM)$

$(4SinM)$

Fig. 3. Single-buffer optical node structures. Top: 4*SoutM*; center: 3*SoutM*; bottom: 4*SinM*.

Unfortunately, in the previous structures the receiver does not have access to the optical buffer. If two packets for the node are received simultaneously, one of them will be missed. Figure 3 (bottom) shows that, if the buffering block in Fig. 3 (top) precedes the add/drop block, the buffer can be *shared* between routing and RX operations. Simultaneous reception from both input links is now possible by storing one of the packets. The add/drop block can be either of the two shown in Fig. 2. This structure will be referred to as 4*SinM*, since the buffer has been shifted to the input of the node, before the add/drop block.

3. Buffer control under non-priority deflection routing

Having defined the structures, the next step is to specify how to control the switches according to the destination of the packets at the input, the packet awaiting transmission, and the destination of the packet in the buffer. The controller is an electronic processor capable of performing all routing decisions and switch settings within the duration of a packet. If this is too demanding, computations can be broken into sequential steps and pipelined, provided that the processing time of the slowest step is shorter than the packet duration [5].

A slotted network operation will be assumed in the following, since the complexity of the controller is much lower than in asynchronous arrivals. There is a trade-off between synchronization complexity and routing control complexity. Also, asynchronous (or non-slotted) unbuffered deflection routing has recently been shown to provide extremely low throughput even at low loads [18]. This is intuitively clear, since if packets do not arrive aligned at the routing switch, the switch cannot change state unless no packet is flowing through it. At moderate-to- high loads, long streams of time-overlapped packets from the two inputs will be flowing through the switch, which is therefore blocked and will cause on average deflections of every other packet.

The most general structure of a node with two inputs, two outputs, single TX/RX and n output shared buffers is shown in Fig. 4. All the proposed 1-buffer structures fit the model with $n = 1$, where only a subset of the possible switch permutations are allowed. Slots arriving at inputs I1 and I2 and from the buffers $M = \{m_1, \ldots, m_n\}$ at each clock can be empty (E), can carry a packet destined for the node (FN), caring to exit on output O1 (C1) or on output O2 (C2) or can have a *don't care* (DC) packet when both outputs provide equivalent shortest-paths to its destination. The same holds for packets ready at the TX, except that FN is not a valid option. Deflections occur when two or more packets contend for the same output and there is not enough memory to store the losers. Don't care packets are totally equivalent to empty packets as far as routing is concerned. Their presence helps avoid deflections. However, DCs and Es are not equivalent as far as transmission is concerned. This is because a polite access scheme is assumed, so that packets waiting at the TX can be injected only in empty slots, and cannot preempt through-going packets.

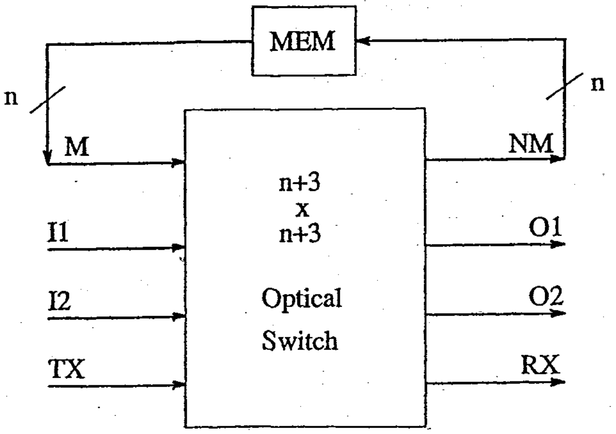


Fig. 4. General node structure.

The objective of the controller is to maximize the node's throughput T and minimize H, the average number of node-to-node hops for packets to reach their destination. Little's law [19] relates T and H as

$$T = \frac{2u}{H}, \tag{1}$$

where u is the link utilization, i.e., the probability that an input slot carries a packet. To maximize T, the controller has to minimize the number of deflections and the number of FN misses, which minimizes H.

The injection strategy for TX packets also has an impact on both H and u. If for instance at all nodes transmitters rarely inject packets, deflections are certainly minimized, but the throughput is low since u is low. A sensible access strategy, called *transmit-hold* (TXH), avoids injecting a TX packet when such an injection causes a deflection. An analytical treatment of TXH was presented in [20] for node structure 3*Shp*. This strategy gives a 5% throughput improvement in a 64-node Shufflenet with no buffers, and the improvement decreases as the network size increases. In the presence of optical buffers the analysis gets rather more complicated, but simulations show the improvement is essentially the same. Since we are interested in a comparison among these structures, in this paper we adhere to the standard access strategy of injecting a TX packet whenever there is room for it in the optical node, whether or not this injection causes a deflection.

If in Fig. 4 all switch permutations are allowed – which implies the buffers are random-access – an efficient controller should do the following to decrease H:

(1) Sort all slots at the input of the switch, except TX, into SORT bins: Es, DCs, C2s, C1s, FNs. Set TX-possible flag ON if there are Es or FNs.

(2) Absorb FN packets first. This will also make room for injections. A miss will occur only if all sorted slots are FN.

(3) If TX-possible, join TX packet to SORT bins.

(4) Next assign packets to outputs O1 and O2. Priorities are, in decreasing order: (C1s and C2s), DCs, Es, FNs.

(4.1) First we try to route out Cares. Unavoidable deflections occur when the two inputs and all buffers care for the same output. It is thus essential to let care packets out as soon as possible. Hence a general rule is to preferably store Es, DCs, FNs, which are equivalent to empty slots for routing purposes.

(4.2) DCs are routed out after Cares. This may force DC packets to recirculate in the node's buffers for several slots. This extra delay does not add appreciably to their total delay if the link-propagation delay is much longer than the slot length, as in very high speed networks. However, such recirculations severely affect the attenuation experienced by DC packets.

(4.3) Route out FNs as a last resort. Since in SN misses add the same extra number of hops as deflections do, it is better to deflect a care through-going packet than to miss a FN packet, since a care packet in the memory may cause a deflection at the next slot, while a FN cannot, since it will be absorbed. Note also that in SN a packet that cannot be absorbed and has to be routed out will have a preferred output, i.e., it becomes a care packet.

(5) All remaining slots are stored in random order.

For the single buffer case, a rearrangeably non-blocking 4 × 4 optical switch can be built as a Benes network with a minimum of five 2 × 2 switches [4, p. 107]. Therefore, this complete shared memory structure with the above control will be referred to as 5*SshM*.

As the number of crossbar switches is reduced, the set of input/output permutations allowed by the node is reduced, and the control becomes less efficient. In the following, the control of structures 4*SoutM*, 3*SoutM* and 4*SinM* will be described.

In each structure, the setting of the TX/RX access switch must ensure correct absorption/injection of packets, according to the polite access scheme and injection strategy described above. The output routing switch S4 is controlled according to a non-priority hot-potato routing of the packets at its input: the switch is set according to the packets' preferences, unless they are both care and in conflict, in which case the state of the switch is randomized. For all controls, therefore, we will detail only the buffer updates and the settings of the buffer access switch.

The complexity of the search for the best control algorithm can be understood from the following considerations. The two inputs and the packet in the buffer can take five values each, E, DC, C1, C2, FN. The packet at the TX can take four values, E, DC, C1, C2. Overall, there are $5^3 \times 4 = 500$ different 4-tuples I1, I2, M, TX. For each of these 500 entries, the control has to specify the state of the buffer input crossbar switch. Therefore, there are $2^{500} \simeq 3 \times 10^{150}$ different possible control strategies. Out of this unmanageable number, we have to find the one (ones) that maximize the throughput. Obviously we cannot proceed by exhaustive search.

The integrated sofware that we developed uses a routine that produces the permutation of each input string M, I1, I2, TX over the corresponding output string NM, O1, O2, RX operated by the control. The permutations table is then passed to a routine that implements the analytical model to be presented in the next section. The analytical model produced the throughput curves for the specified control algorithms in a few minutes for network size up to 10 000 nodes on a Sun SPARC5 workstation.

It turns out that, in the process of finding a good control, for many of the input strings M, I1, I2, TX the best setting of the switches is obvious, since there are no conflicts. However, for many other input strings such best setting is not obvious at all since the deflection and miss probabilities depend both on the control and on the statistics of M, I1, I2, TX. These statistics in turn depend on the deflection and miss probabilities, and the exact effect of such a feedback on throughput must be settled numerically by resorting to the analytical tool.

Since there are too many possible control strategies that make sense, we could not attempt a global optimization. We divided the 500 possible strings into a number of meaningful distinct patterns and tried a local maximization over each of these patterns. Therefore, the controls presented next do not claim any global optimality, but do provide the best throughput figures provided by our heuristic search. As it will be shown in the results section, such controls provide throughput figures close to the upper bound provided by the *5SshM* structure.

1) 4*SoutM*. RX/TX operations are not coordinated with buffering operations. First, FN packets are removed and TX packets injected if there are empty slots. Figure 5 (top) shows the settings for the buffer access switch S3, given the contents of its inputs I1 and I2. Buffering proceeds as in [7]: when two inputs of the same kind (E and DC are not distinguished) are present, switch S3 is randomized. When two care non-conflicting packets are present and M is E or DC, switch S3 is also randomized; the undesired buffering of a care packet in such a non-conflict situation is the only inefficiency of this scheme. Care packets at the inputs of S3 are stored if in conflict with the packet in M to avoid a deflection. Otherwise DCs or Es are stored when present to reduce the probability of a deflection at the next slot. Missed FN packets in SN have a preferred output, O1, so that they behave as C1 packets at the router.

2) 3*SoutM*. The access switch S2 is set in cross only to receive FN packets or to inject TX packets when an injection is possible, i.e., when an E or a FN enters switch S2 from S1. The settings of the memory access switch S1 are described in Fig. 5 (center). We refer to a *line* as an if/elseif statement. Randomization of S1 in line 1 ensures equal treatment of both channels. Randomization is also applied when two care non-conflicting packets are present and the memory is E/DC. With no information about the next TX packet, either care can be stored. If the next TX packet is known, the right care could be stored to avoid a conflict at the next slot. Lines 2, 3 ensure that FN packets are absorbed. Also, E slots at the input get routed to the TX for a possible injection. When an E and a Care packet are present at the input while the TX is full, the E is routed to the TX for injection, while the Care is stored. This is referred to as *TX-priority*. TX-priority trades an increase of the injection probability at the present clock for an increase of the deflection probability at the next clock. At high load, most empty slots for the TX are provided by absorptions of FNs. Thus TX-priority has a positive effect only at low loads, when Es are frequent, and yields larger throughput than a non TX-priority rule in which Es are preferably stored to avoid deflections at the next slot. As shown in the final lines, conflicts of Care packets with the memory are resolved by storing them; otherwise Es and DCs are stored if possible.

```
   /* Control Algorithm for 4SoutM */
begin
   /* Setting of switch S3: X=cross; B=bar; R=randomized */

     if          (I1=I2) or (I1,I2)∈{(E,DC),(DC,E)}
             or (M∈{E,DC} and (I1,I2)∈{(C1,C2),(C2,C1)})   then   S3:=R
   elseif   (M∈{C1,FN} and I1=C1) or (M=C2 and I1=C2)   then   S3:=B
   elseif   (M∈{C1,FN} and I2=C1) or (M=C2 and I2=C2)   then   S3:=X
   elseif                    I1∈{E,DC}                   then   S3:=B
    else           /* only cases I2∈{E,DC} left */              S3:=X;
end
```

```
   /* Control Algorithm for 3SoutM */
begin
   /* Setting of switch S1: X=cross; B=bar; R=randomized */

     if                    (I1=I2) or
            (M∈{E,DC} and (I1,I2)∈{(C1,C2),(C2,C1)})    then   S1:=R
   elseif          (I1=FN) or ((I1,I2)=(E,DC)) or
                   (TX≠E and (I1,I2)=(E,C))             then   S1:=X
   elseif          (I2=FN) or ((I1,I2)=(DC,E)) or
                   (TX≠E and (I1,I2)=(C,E))             then   S1:=B
   elseif   (M∈{C1,FN} and I1=C1) or (M=C2 and I1=C2)   then   S1:=B
   elseif   (M∈{C1,FN} and I2=C1) or (M=C2 and I2=C2)   then   S1:=X
   elseif                    I1∈{E,DC}                   then   S1:=B
    else           /* only cases I2∈{E,DC} left */              S1:=X;
end
```

```
   /* Control Algorithm for 4SinM */
begin
   /* Setting of switch S1: X=cross; B=bar; R=randomized */

     if      (I1=I2) or (I2,I2)∈{(E,DC),(DC,E)}      then   S1:=R
   elseif    (I1,M)∈{(C1,C1),(C2,C2),(FN,FN)}        then   S1:=B
   elseif    (I2,M)∈{(C1,C1),(C2,C2),(FN,FN)}        then   S1:=X
   elseif       ((I1,I2)=(FN,C1) and M≠C1)
              or ((I1,I2)=(FN,C2) and M≠C2)           then   S1:=X
   elseif       ((I1,I2)=(C1,FN) and M≠C1)
              or ((I1,I2)=(C2,FN) and M≠C2)           then   S1:=B
   elseif            M≠FN and I1=FN                   then   S1:=X
   elseif            M≠FN and I2=FN                   then   S1:=B
   elseif   M≠C and [[(I1,I2)=(C1,C2) and TX=C1]
              or [(I1,I2)=(C2,C1) and TX=C2]]         then   S1:=B
   elseif   M≠C and [[(I1,I2)=(C1,C2) and TX=C2]
              or [(I1,I2)=(C2,C1) and TX=C1]]         then   S1:=X
   elseif                I1∈{E,DC}                    then   S1:=B
    else        /* only cases I2∈{E,DC} left */              S1:=X;
end
```

Fig. 5. Control algorithms.

3) 4*SinM*. The main advantage of shifting the buffering block ahead of the add/drop block is that the miss probability gets drastically reduced at almost no expense of deflections, since stored FNs are equivalent to empty slots in most input/TX configurations. Switch S3 has the only purpose of routing FNs and Es to the TX, and switch S2 serves for absorption/injection. Figure 5 (bottom) describes the settings of the buffer access switch S1. In lines 2, 3 there is proper buffering to avoid immediate deflections or misses. Lines 4, 5 show that absorption of FN packets can be delayed to avoid buffering Care packets. This slightly increases the miss probability for a decrease of the deflection probability at the next clock. The presence of the TX-access switch S3 allows avoiding conflicts with the TX at the present slot, as seen in lines 8, 9.

The next section presents the analytical procedure to evaluate the throughput of these structures.

4. Steady state analysis in uniform traffic

4.1. Definitions and assumptions

The steady state behavior in uniform traffic of a two-connected regular mesh network will now be analyzed.

Regularity means that each node is topologically equivalent to all other nodes. Since in uniform traffic all nodes have identical statistical behavior, it is enough to focus on a single node for the analysis. SN is an example of regular network.

A common clock is distributed to all nodes, so that node operations are performed in fixed length time slots.

New arrivals at each node are collected in an electronic FIFO TX queue, waiting to be injected in the network. We assume the input queue to be of infinite size. 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. This is the uniform traffic pattern. The assumed regularity of the network and the randomness associated with deflection routing help keep this homogeneous traffic pattern.

Up to saturation of the input queue, the arrival rate at the queue equals the node throughput T, i.e., the average number of packets inserted/absorbed per slot by the node at equilibrium. The throughput and the number of hops H taken on average by a packet to reach its destination are related by Little's law (1) to the link utilization u, which is the probability that an input link is occupied by a packet at each clock. Network regularity and uniform traffic pattern ensure that u is the same for both inputs.

The total delay of a packet is given by

$$D = EQ + H(W + D_q), \tag{2}$$

where EQ is the average queuing time at the node's input buffer, W is the propagation delay in slots/hop and D_q is the queueing delay at the optical buffers of each visited node. For deflection routing, D_q is of the order of the number of buffers provided in the optical routing block. For very high bit rate optical networks W can be as large as 100 [2], so that D_q can safely be neglected.

Let g be the probability that the node's TX buffer has at least one queued packet per slot.

Let r be the probability that an input link contains a packet for the node, given that the link is full. It will be called *reach probability*.

Let P_{dc} be the probability that an input link contains a through-going don't care packet, given that the link is full. It will be called *don't care probability*.

The standard assumption of the model is that arrivals at the two input ports $\mathrm{I1}(k)$ and $\mathrm{I2}(k)$ at every time slot k are independent random variables [12]. This approximation makes sense in regular mesh networks in uniform traffic and with a random routing rule like deflection routing.

At each clock, each node's input link can be E with probability $1 - u$, DC with probability uP_{dc}, FN with probability ur and care (C1 or C2) otherwise. It is assumed that C1s and C2s are equally likely [3]. This is the probability distribution of the arrivals I1 and I2.

Also, each packet presented by the TX to the network can be E with probability $1 - g$, DC with probability gP_{dc0}, and care otherwise, being C1s and C2s equally likely in generation by the uniform traffic assumption. P_{dc0} is the fraction of network nodes that can be reached from either output link of the transmitting node in the same minimum number of hops, which just depends on the selected regular topology. This is the probability distribution of the arrivals at the TX.

[2] The propagation delay in fiber optic links is about 5 μs/km, so that for an average 10 km hops, and for packets of 500 bits at a binary signalling rate of $R = 1$ Gb/s, $W = 100$.

[3] This has been shown to be a correct assumption for SN, independently of the size and of the load of the network [21].

To deal with buffered deflection routing, we introduce a stronger assumption: arrivals at the same input link are independent slot-by-slot, i.e., for any slot-times k_1 and k_2, the random variables $Ij(k_1)$ and $Ij(k_2)$ are independent, $j = 1, 2$ [7,21]. With this assumption:

i) in the single-buffer schemes the buffer content is a Markov process $M(k)$ whose states are E, DC, C2, C1, FN. The transition probabilities are determined by the distributions of the inputs I1, I2 and of the TX, and by the specific control algorithm.

ii) in the general case of n buffers, the memory content is an n-dimensional Markov process $M(k) = \{m_1(k), \ldots, m_n(k)\}$, with 5^n states. The number of states grows so quickly with n that we are computationally limited to a small number of buffers n.

While the assumption on independence of I1(k) and I2(k) at all times k is justified by the regularity of the network and the uniformity of the traffic, the slot-by-slot independence of arrivals on a link is clearly violated since at high loads buffers tend to correlate successive arrivals. The effect of neglecting such correlations is that the analytical model tends to over-estimate the throughput, the error being larger for large buffers and large networks [21]. However, we concentrate here mainly on single-buffer structures, in which such correlations tend to be very weak. Also, this 'biased' throughput estimation is common to all structures and should not be of much concern since our main objective is to *compare* constrained node structures and their controls.

Finally, we have to deal with the presence of the TX input queue.

It is customary in the deflection routing literature to choose g as the *free* parameter to obtain throughput and average number of hops [12,21–23]. However, blocked TX packets are usually supposed to be cleared and discarded. In the presence of an input first-in-first-out (FIFO) TX queue, blocked packets remain in the queue, trying an access at the next slot. If slot-by-slot correlations on the input links can be neglected, the rate at which the input queue is served is independent of the instantaneous queue occupancy and only depends on the average number of packets presented to the transport part of the network by each node, namely g. In networks with links which are long as compared to the slot length, slot-by-slot correlations are only due to the effect of in-line buffers (the delay lines) and turn out to be very weak, as our simulations results show. Therefore, in the slot-by-slot independence assumption, the service time is independent slot-by-slot and the FIFO TX queue can be treated as a discrete-time single-server queue with geometric service time and geometric arrivals at rate T. Similar arguments about TX input buffering can be found in [23].

By noting that the average number of injected packets per slot, i.e., T, is given by

$$T = gPr[\text{injection is possible}], \tag{3}$$

the probability that service is given at the end of each slot is $Pr[\text{injection is possible}] = T/g$. Relation (3) is exact when C and DC packets have the same injection probability. It is indeed an approximation for the *5SshM* scheme.

Hence, EQ is the average waiting time at a Geo/Geo/1 queue with early arrivals/ late departures, with arrival rate T and average service time g/T. By applying the Pollaczek–Khinchin formula [19] we find:

$$EQ = \left(\frac{g}{T} - 1\right) / (1 - g). \tag{4}$$

The next section will derive T as a function of g. Thus, we will be able to express EQ as $EQ(g)$.

4.2. Solution procedure

The distribution of the TX is the known forcing of the network. The parameter g will be treated as the free network parameter and the objective is to express all other quantities as a function of g only. In particular, throughput $T(g)$ and hop-delay $H(g)$ will be found. From (2) we get $D = D(g)$. Finally, we will be able to plot delay $D(g)$ vs. throughput $T(g)$.

The solution procedure is based on the following steps:

1) Given the probability distribution of I1, I2 and TX, the steady state distribution of the memory $M(k)$ is obtained by solving the corresponding Markov chain, whose transition probabilities depend on the buffer control algorithm.

2) By conditioning on all possible input triplets of independent random variables $\{I1, I2, TX\}$ and on all possible memory configurations $\{m_1, \ldots, m_n\}$ and by averaging out using their known distributions, it is possible, for a specific node structure and control, to find the quantities:

(i) $a =$ probability that an input FN packet is absorbed.
(ii) $d_m =$ probability that an input FN packet is missed and deflected.
(iii) $d =$ probability that an input care packet is deflected.
(iv) $d_0 =$ probability that an injected TX packet is deflected.
(v) $T_{\text{in}} =$ probability per slot of absorbing a packet.
(vi) $T_{\text{out}} =$ probability per slot of injecting a packet.

For instance, in the single-buffer case, we have

$$a \triangleq Pr\{\text{I1} = \text{FN absorbed}\} = \sum_{i_2}\sum_{tx}\sum_{m} \mathrm{I}_{\{\text{I1} = \text{FN absorbed}\}}(i_2, tx, m)\, Pr(i2, tx, m),$$

where $\mathrm{I}_{\{\mathcal{E}\}}(x)$ is the indicator function of event $\mathcal{E}$, taking value 1 when x is such that $\mathcal{E}$ is true, and 0 otherwise; the summations are extended to all possible values of I2, TX, M; and $Pr(i2, tx, m) = P(i2)P(tx)P(m)$ is the joint probability of the triplet $\{\text{I2} = i2, \text{TX} = tx, \text{M} = m\}$, and can be factorized by the independence assumption. By the symmetry of the control we also have $a \triangleq Pr\{\text{I2} = \text{FN absorbed}\}$. The other quantities $d_m, d, d_0, T_{\text{in}}, T_{\text{out}}$ are computed similarly. All of these quantities depend *only* on the distributions of I1, I2 and TX, i.e., on the *parameters* g, P_{dc0} and on the *unknowns* u, r, P_{dc}. Such symbolic computations have been automatized using Mathematica™.

3) At equilibrium, quantities (v) and (vi) must be equal. This provides the first relation binding the three unknowns. For instance, we can get an expression for the link utilization $u = u(g, P_{\text{dc0}}, r, P_{\text{dc}})$. Two more equations are needed to solve for r, P_{dc}.

4) A procedure is described next to derive r, P_{dc} as functions of the absorption and deflection statistics a, d_m, d, d_0. As a byproduct, H will also be found. The procedure appeals to the uniform traffic assumption, in which every packet is a 'typical' packet. It is thus a matter of following the trajectory of a *typical* or *test* packet hopping towards its final destination in a 'uniform gas' of competing packets. The random walk of the test packet can be visualized as an absorbing Markov chain whose states coincide with the network nodes [7,22].

For some topologies, like SN, it is possible to speed up the computation by drastically reducing the number of states in the chain. This is done by combining together in a single state all nodes with the same distance to destination. The test packet thus performs a random walk on the integers $0, 1, \ldots, d_{\max}$ where $d_{\max}$ is the maximum distance to destination [24].

Whether or not a reduced-state chain can be used, the procedure yields r, P_{dc}, D as functions of a, d_m, d, d_0, and its improved algorithm is detailed in the appendix. Since from step 2) a, d_m, d, d_0 depend on r, P_{dc}, we have obtained two more equations that allow to solve for r, P_{dc}.

Summarizing, from steps 1) to 4) a 3×3 system of nonlinear equations in the unknowns u, r, P_{dc} is available, whose solution can be found numerically for every value of g. By back-substitution, the curve $T(g)$ can be found from (v) and (vi) and verified by Little's law (1).

5. Results

The curves presented next have been found for a Shufflenet topology by the previous analytical procedure.

Figure 6 summarizes delay/throughput results for the proposed structures in a 64-node SN. The delay has been normalized by W, the per-hop propagation delay.

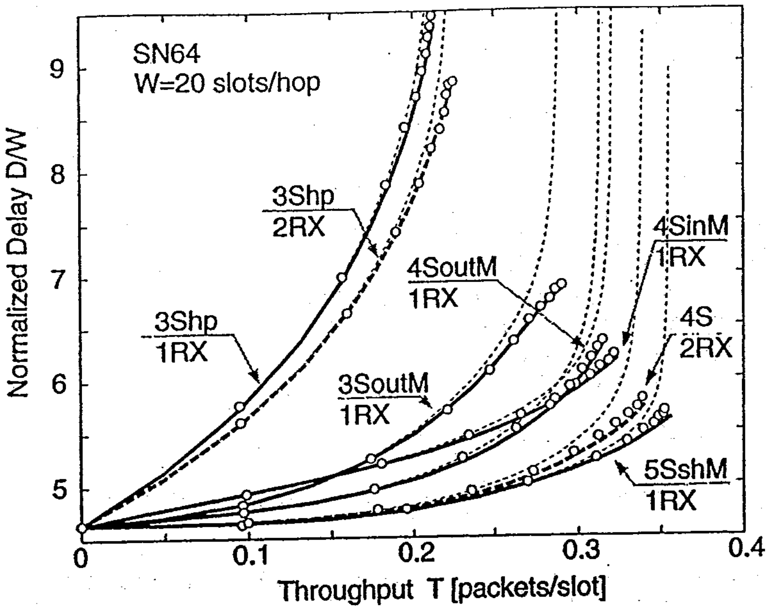


Fig. 6. Delay vs. throughput in a 64-node Shufflenet.

For all the analyzed schemes, the overall delay is shown by the dotted light curves, for a value $W = 20$ slots. Bold curves represent delay neglecting the input buffering component EQ. It is verified that input buffering delay becomes dominant only very close to saturation, and can be safely neglected for all other loads.

Simulations were run for 64-node SN to validate the model, and are marked with circles in the plot. Simulations were run for a system without input TX buffers, for $g = 0, 0.1, 0.2, \ldots, 1$. Simulation statistics were collected for 10 000 clock cycles, after discarding 1000 initial cycles to allow for transients to die out. There is a very good agreement between the analytical model and the simulations.

Let's compare the various schemes based on the bold approximate curves.

First consider the curves for 3*Shp*, 4*SoutM* and 4*SinM*. Dashed lines refer to the same structures but with two receivers [7], where misses do not occur. With two receivers, structures 4*SoutM* and 4*SinM* exhibit the same throughput (curve 4S,2RX). The gap between the 1RX curve and the corresponding 2RX curve accounts for the effect of missing FN packets. The gap is wider in the 4-switch structures, i.e., when buffers are present.

When only one RX is present, and at large loads, structure 4*SinM* reduces the miss probability with respect to 4*SoutM* without significantly degrading the deflection probability, so that it achieves larger throughput than 4*SoutM*. This proves the positive effect of shifting buffering before the add/drop block for single TX/RX nodes. However, at low loads 4*SoutM* is better.

Structures with fewer switches exhibit worse performance since the control is less flexible. However, note how well structure 3*SoutM* compares with the 4-switch nodes.

The curve for the non-blocking switch 5*SshM* with a single buffer and single receiver provides an upper bound for all the single-receiver/single- buffer structures. Its performance is even better than 4S with 2 RXs, since 1) it better handles two non- conflicting input care packets and 2) it reduces blocking in the TX FIFO queue by storing, when possible, TX packets in the shared optical buffer.

However, in a practical optical implementation, buffered packets may need to cross the 4 × 4 switch several times, each time crossing three 2 × 2 switches. The power loss on such buffered packets could turn out to be unacceptably large. The great advantage of the other structures is to have a number of 2 × 2 switch crossings per input channel no larger than three (using the parallel structure of the add/drop block when possible). Most importantly, buffered and unbuffered packets will experience the same loss. This allows to keep a small dynamic range at the optical receiver.

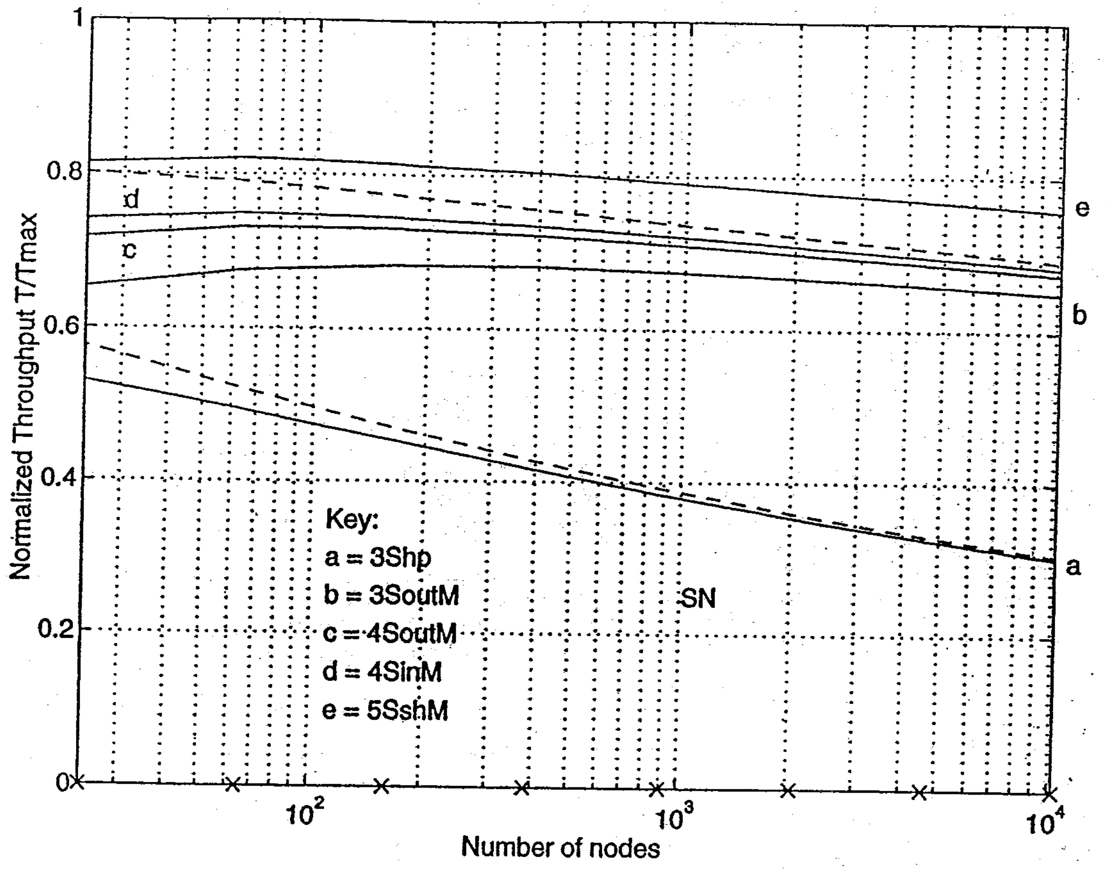


Fig. 7. Normalized saturation throughput vs. network size in Shufflenet.

The saturation throughput figures should be compared to the largest achievable value $T_{\max} = 2/D_{\min}$, where $D_{\min}$ is the average number of hops when deflections and misses never occur. Figure 7 shows analytical curves for the normalized throughput $T/T_{\max}$ vs. network size in a two-connected SN. The relative throughput for 3*Shp* quickly degrades from over 50% to about 30%. Buffered structures show a stable behavior instead, with little degradation with increasing network size. Structure 3*SoutM* shows a maximum for a 160-node SN. The throughput differences among 3*SoutM*, 4*SoutM* and 4*SinM* tend to level off for large networks, where their throughput settles around 70% for 10 000 nodes. The efficient 5*SshM* degrades less than all other structures, with relative throughput around 80%.

6. Conclusions

Novel low-attenuation single-receiver single-buffer optical node structures for deflection routing TONs have been proposed and analyzed in uniform traffic in a Shufflenet topology.

These novel structures point to the feasibility of extremely simple, low-attenuation optical nodes that allow very fast electronic routing control.

The effect on throughput of adding flexibility to the switching process has been analyzed by comparing nodes with three, four and five switches. Important differences in optical power loss per input channel among the various structures have been pointed out.

It has been shown that throughput results scale well with network size for buffered structures.

Slightly better throughput figures could be obtained by using distance- priority rules to solve contentions [25, 26]. It has been shown that such priority rules give negligible gain at very low deflection probability [26]. However, the nodes with such priority deflection routing should prove to be robust to traffic correlations arising from long bursts of packets with fixed destination, such as those arising in typical computer communications. In fact such correlations tend to increase the deflection probability, at which point the priority rules become effective.

References

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

[2] Special issue on broad-band optical networks, *IEEE J. Lightwave Technol.* 11 (May/June 1993).

[3] Issue on optically multiplexed networks, *IEEE Commun. Mag.* 32(12) (1994).

[4] H.S. Hinton, *An Introduction to Photonic Switching Fabrics*, Chs 2, 3, Plenum Press, 1993.

[5] P.R. Prucnal and P.A. Perrier, Optically-processed routing for fast packet switching, *IEEE LCS Mag.* 1 (May 1990), 54–67.

[6] D.J. Blumenthal, K.Y. Chen, J. Ma, R.J. Feuerstein and J.R. Sauer, Demonstration of a deflection routing 2 × 2 photonic switch for computer interconnects, *IEEE Photon. Technol. Lett.* 4 (Feb. 1992), 169–173.

[7] F. Forghieri, A. Bononi and P.R. Prucnal, Analysis and comparison of hot-potato and single-buffer deflection routing in very high bit rate optical mesh networks, *IEEE Trans. Commun.* 43(1) (1995), 88–98.

[8] A. Bononi, F. Forghieri and P.R. Prucnal, Design and channel constraint analysis of ultra-fast multihop all-optical networks with deflection routing employing solitons, *IEEE J. Lightwave Technol.* 11(12) (1993), 2166–2176.

[9] P. Baran, On distributed communications networks, *IEEE Trans. Commun. Syst.* 12 (Mar. 1964), 1–9.

[10] I. Chlamtac and A. Fumagalli, An all-optical switch architecture for Manhattan networks, *IEEE J. Select. Areas Commun.* 11 (May 1993), 550–559.

[11] A.S. Acampora, M.J. Karol and M.G. Hluchyj, Terabit lightwave networks: the multihop approach, *AT&T Tech. J.* 66 (Nov./Dec. 1987), 21–34.

[12] A.G. Greenberg and J.B. Goodman, Sharp approximate models of deflection routing in mesh networks, *IEEE Trans. Commun.* 41 (Jan. 1993), 210–223.

[13] R.A. Thompson and P.P. Giordano, An experimental photonic time-slot interchanger using optical fibers as reentrant delay-line memories, *IEEE J. Lightwave Technol.* 5 (Jan. 1987), 154–162.

[14] H.F. Jordan, D. Lee, K.Y. Lee and S.V. Ramanan, Serial array time slot interchangers and optical implementations, *IEEE Trans. Comput.* 43 (Nov. 1994), 1309–1318.

[15] M.J. Karol, Shared-memory optical packet (ATM) switch, in: *SPIE vol. 2024: Multigigabit Fiber Communication Systems*, July 1993.

[16] Z. Haas, The 'Staggering switch': an electronically controlled optical packet switch, *IEEE J. Lightwave Technol.* 11 (May/June 1993), 925–936.

[17] M. Calzavara, P. Gambini, M. Puleo, M. Burzio, P. Cinato, E. Vezzoni, F. Delorme and H. Nakajima, Resolution of ATM cell contention by multiwavelength fibre loop memory, in: *Proc. ECOC '94*, Florence, Italy, 1994, pp. 567–570.

[18] F. Borgonovo, L. Fratta and J. Bannister, Unslotted deflection routing in all-optical networks, in: *Proc. IEEE GLOBECOM '93*, Houston, TX, 1993, pp. 119–125.

[19] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1987.

[20] A. Bononi and P.R. Prucnal, Analytical evaluation of improved transmission techniques in deflection routing networks, in: *Proc. 28th Conf. on Inform. Sci. and Syst.*, session WP-1, Princeton, NJ, 1994.

[21] A.K. Choudhury and V.O.K. Li, An approximate analysis of the performance of deflection routing in regular networks, *IEEE J. Select. Areas Commun.* 11 (Oct. 1993), 1302–1316.

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

[23] A.G. Greenberg and B. Hajek, Deflection routing in hypercube networks, *IEEE Trans. Commun.* 40 (June 1992), 1070–1081.

[24] A. Krishna and B. Hajek, Performance of Shuffle-like switching networks with deflection, in: *Proc. INFOCOM '90*, vol. 2, San Francisco, CA, 1990, pp. 473–480.

[25] Z. Zhang and A. Acampora, Performance analysis of multihop lightwave networks with hot-potato routing and distance-age-priorities, in: *Proc. INFOCOM '91*, vol. 3, Bal Harbor, FL, 1991, pp. 1012–1021.

[26] A.K. Choudhury and V.O.K. Li, Effect of contention resolution rules on the performance of deflection routing, in: *Proc. GLOBECOM '91*, 1991, pp. 1706–1711.

[27] J.G. Kemeny, H. Mirkil, J.L. Snell and G.L. Thompson, *Finite Mathematical Structures*, Prentice Hall, 1959, pp. 404–409.

[28] S-H. Chan and H. Kobayashi, Performance Analysis of Shufflenet with deflection routing, in: *Proc. GLOBECOM '93*, vol. 2, Houston, TX, 1993, pp. 854–859.

[29] A.V. Ramanan, H.F. Jordan, J.R. Sauer and D.J. Blumenthal, An extended fiber-optic backplane for multiprocessors, in: *Proc. 27th Hawaii Int. Conf. on Syst. Sci.*, vol. 1, Maui, Hawaii, 1994, pp. 462–470.

Appendix

This procedure can be applied to any regular topology, whether or not a reduced state-space can be used. However, for illustration purposes, a SN topology will be used.

A specific example of the absorbing Markov chain describing the random walk of the test packet towards its destination is given in Fig. 8 for a 64-node SN$(2,4)$ topology.

As shown in Fig. 1, a SN(q,k) topology has $N = k\,q^k$ nodes arranged in k columns of q^k nodes each, and there is a perfect shuffle connection among nodes in adjacent columns [11]. The maximum distance between nodes is $d_{\max} = 2k-1$. Fix a destination node. All nodes reachable in less than $k+1$ hops proceeding backwards are Care with respect to that destination. All the remaining nodes, at distance $k+1,..,2k-1$ are don't care. A deflection of the test packet flowing towards that destination at a node at distance i brings the packet back to the set of nodes at distance $i+k-1$. A deflection at the destination node brings the packet back at distance $2k-1$, while a miss brings it back at distance $k-1$. Finally, there are q^i nodes at distance $1 \leqslant i \leqslant k-1$, and $q^k - q^{i-k}$ nodes at distance $k \leqslant i \leqslant 2k-1$.

Figure 8 refers to the initial step of the walk, where the packet is at its injection node. The labels are the transition probabilities, defined in Section 4.2. For every step after the first hop, in which the packet is at the TX port of the node, label d_0 changes in d. The nodes represent the distance in hops of the test packet to its destination. A fictitious absorbing state A has been added to take into account the possibility of missing the test packet at its destination.

For this model to hold, it is necessary that the controller's treatment of both input links be the same, so that it is not required to know which link the packet comes from.

For all steps $t = 1,2,\ldots$, the transition probabilities $\pi(l,m)$ from state m to state l, $l,m = A,0,1,\ldots,7$, can be organized in a transition matrix $\Pi = \{\pi(l,m)\}$. Analogously, a matrix Π_0 can be written for the injection step $t=0$.

Since A is the only absorbing state, if states are ordered as $A,0,1,..,7$ then matrix Π is in its canonical form. Taking off the first row and the first column, matrix Q is obtained. From this, the fundamental matrix of the absorbing chain $\mathcal{N} = (I-Q^{\mathrm{T}})^{-1}$ is obtained, where I is the 8×8 identity matrix. The entries of $\mathcal{N} = \{n(l,m)\}$, $l,m = 0,1,\ldots,7$, give the mean number of times in each nonabsorbing state m for each possible nonabsorbing state l after the first hop [27].

Consider the set of nonabsorbing states $0,1,..,7$. Define the column vectors $fm = [10000000]'$, $dcs = [00000111]'$ (ones in the positions corresponding to don't care states), and $all = [11111111]'$. Let $p(0)$ be the probability state vector at the injection step. By the uniform traffic assumption and the results on the number of nodes at each hop-distance given before, $p(0) = [0\,2\,4\,8\,15\,14\,12\,8]/63$. The state after the first hop is $p(1) = p(0) * \Pi_0$. Let p indicate $p(1)$ with the first component removed.

It is easy to see that :

1) the Expected Number of visits of state 0 before absorption is $EN_{fm} = p * (\mathcal{N} * fm)$;
2) the Expected Number of visits to don't care nodes at which the test packet is flow-through is $EN_{\mathrm{dc}} = p * (\mathcal{N} * dc)$; and

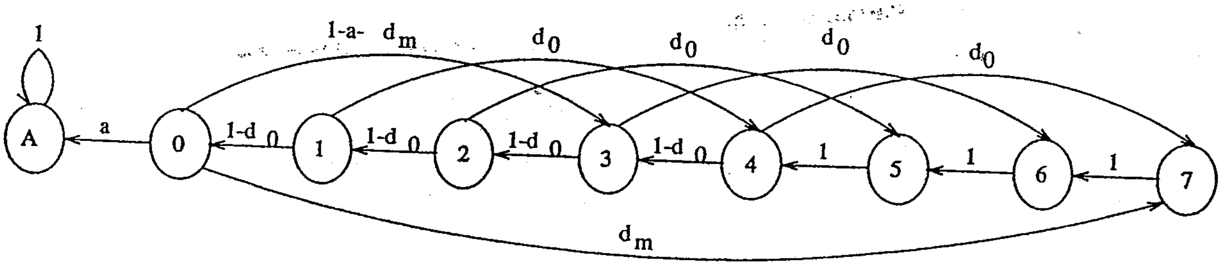


Fig. 8. Markov chain describing the random walk of the test packet in a SN(2, 4) topology.

3) the Expected Number of visits to any node before absorption, i.e., the average number of hops before reception, is $H = p * (\mathcal{N} * all)$.

From these, estimates of the don't care and reception probability are formed as

$$P_{\mathrm{dc}} = \frac{EN_{\mathrm{dc}}}{H}, \qquad r = \frac{EN_{fm}}{H}. \tag{A.1}$$

The first equation estimates P_{dc} as the fraction of time the test packet is don't care flow-through. It is immediate to see from the chain in Fig. 8 that $EN_{fm} = 1/a$. Since $r\,a$ is the unconditional probability of absorption, Little's law gives the known relation $r\,a = 1/H$, as expected.

With this procedure, making use of the fundamental matrix of the absorbing chain, we can also find closed-form expressions for H, r, P_{dc} as functions of d, d_0 using symbolic matrix inversion. Closed-form expressions for absorbing chains in SN that do not involve packet misses have been reported in [28] and [29] by neglecting the initial injection step.