Performance Model of Multichannel Deflection-routed All-Optical Networks with Packet Injection Control

C. Y. Li,[1] P. K. A. Wai,[2] and Victor O. K. Li[1]
[1]Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China, {cyli, vli}@eee.hku.hk
[2]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, China, alex.wai@polyu.edu.hk

Abstract— **Deflection routing is a feasible approach to resolve the output contention problem in packet-switched networks when buffering of packets is not practical. In this paper, we investigate the performance of multichannel deflection-routed networks with no packet injection control, strict packet injection control, and a simple token-bucket-based packet injection control. The analytical performance models of multichannel deflection-routed networks with strict packet injection control are derived. Simulation results show that the analytical models can accurately predict the performance regardless of the network topology, number of channels, and packet injection control methods. We observed that the end-to-end throughput-delay and the packet re-transmission performance at sources can be largely improved by using simple packet injection control mechanisms such as the proposed token-bucket-based method.**

I. INTRODUCTION

Deflection routing has been recognized as a potential candidate for output contention resolution in packet-switched networks when buffers are not practical or too expensive to implement [1-3], such as in all-optical networks. In deflection-routed networks, packets that lose in the contention for their desired outputs are deflected through available outputs to neighboring nodes. The deflected packets are then routed to their destinations by the neighboring nodes. Deflection routing simplifies the implementation and operation of packet-switched networks because buffers are no longer needed.

Lowering the cost per packet deflection and reducing the packet deflection rate are two valid approaches to improve the performance of deflection-routed networks. Lowering the cost of each deflection typically requires nodes to select outputs that will lead to the next least cost routing paths for the deflected packets. Thus the nodes must have sufficient computational capability for complicated packet processing and this is difficult to achieve if the data transmission rate is high. Although various approaches have been proposed to reduce the required computations and hardware complexity, the advantages are often offset by different practical concerns [3].

In contrast, lowering the packet deflection rate generally does not require complicated packet processing but needs more hardware or network-wide status information. For example, substantial throughput-delay performance improvement can be obtained by replacing a single high speed channel by multiple lower speed ones with the same aggregate capacity [4]. The main concern of multichannel deflection-routed networks is the implementation cost. In optical networks, larger optical switches, wavelength converters, wavelength multiplexers and demultiplexers, and/or timeslot interchangers (TSIs) will be additionally required to implement the multichannel capacity on deflection-routed networks. We may therefore need to keep the number of channels per link small even if a larger number of channels will provide better system performance.

Packet injection control, which requires basic network status information, is another handy approach to improve the network performance [5], [6]. Packet injection control has been a powerful tool to improve system performance for networks with static packet routing paths such as multiple protocol label switching (MPLS) and asynchronous transfer mode (ATM) networks, [5], [6]. So far, lowering the cost per packet deflection such as the deflection preference in [3] seems to be more attractive for improving the performance of deflection-routed networks. There are only a handful of studies on packet injection control for deflection routing communication networks [7-9]. As far as the required hardware and processing capability are concerned, however, packet injection control can be a viable solution.

In this paper, we propose to combine multichannel deflection-routing with different packet injection control methods to improve the performance of communication networks, especially all-optical networks. The contributions in this paper are:

- The first to propose combining multichannel approach with packet injection control to improve the performance of deflection-routed networks.
- The first to derive the analytical performance models of the multichannel deflection-routed networks with strict packet injection control.
- By comparing with simulation results, it is shown that the analytical models can provide accurate performance predictions regardless of the network topology, number of channels, and packet injection control methods.
- Two packet injection control schemes have been proposed: the strict packet injection control and the token-bucket-based packet injection control.
- Strict injection control has better end-to-end delay throughput performance but its local packets may have large retransmissions at the source node in most of the

loading ranges apart from the very high loading cases.

- The token-bucket-based method is proposed because of its flexibility. Its performance can be similar to normal deflection routing (no packet injection control) when loading is low and that of strict packet injection control when loading is high.
- Simulation results show that packet injection control is an effective way to improve the performance of deflection-routed networks. Both the end-to-end delay-throughput and the packet retransmission performance at sources can be improved by using simple packet injection control mechanisms such as the proposed token-bucket-based method.

The rest of the paper is organized as follows. The background of multichannel deflection-routed networks, packet injection control, and the structure of optical network nodes with wavelength converters and TSIs are first reviewed in Section II-A. Strict packet injection control is discussed in Section II-B. Token-bucket-based control is further proposed in Section II-C to allow more flexibility in performance tuning. As shown in Section IV-B, its performance can be similar to deflection routing without packet injection control when loading is low and that of strict packet injection control when loading is high. The analytical performance model for all-optical multichannel deflection-routed networks with strict packet injection control is derived in Section III. To facilitate the discussion of equation derivation, all variables are listed in the Appendix. Those who are not interested in details of the model derivation can go directly to Section III-B6 which numerically solves the probabilities in the model to evaluate the network throughput-delay performance. The accuracy of the analytical performance model is verified using simulations in Section IV-A. The performance of different kinds of packet injection controls are demonstrated in Section IV-B. A conclusion is given in Section V.

II. Multichannel Deflection-Routed Networks and Packet Injection Control

A. Background

Packet deflection was first proposed in hot-potato routing [10] for distributed communication networks. It has been later used in massively parallel machines such as Connection Machine [11] to facilitate interconnection between processing boards. These systems are equipped with buffers though packet deflection has reduced the required buffering and processing of packets. To totally eliminate the need of buffers in intermediate nodes, [1] first proposed a slotted deflection-routed network with features of (1) packet deflection, (2) same in-degree and out-degree of a node, (3) packets carried in fixed size timeslots transmitting at the same rate everywhere in the network, (4) transit packets always having priority over local packets, and (5) synchronized timeslot transmission at each node. Deflection routing was original proposed for slotted packet-switched networks [1-3], [7-17] but the concept of deflection has been widely applied in other kinds of communication networks for performance

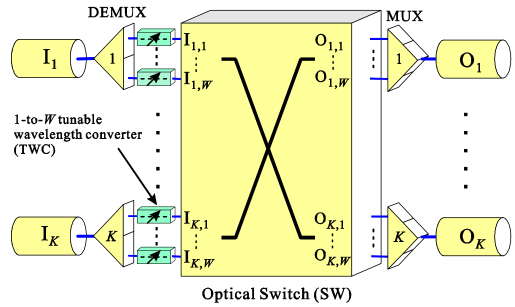

Fig. 1 A K-degree node in networks with W wavelength channels per link.

improvement [18-21]. Deflection routing has become a general term for networks solving output contention by deflection regardless of the deflected data units being packets (with fixed size [1-3], [7-17] or with variable length [18], [19]), or even bursts of packets [20], [21].

Deflection routing is popular in all-optical networks because optical packet buffering is not practical yet and the concern of optical-electrical conversion (O/E) overhead makes electronic buffering and processing packets unattractive [22]. If the networks are electric, however, deflection routing as well as the proposed schemes will also be useful to performance improvement but may be less attractive because packet buffering and processing can be handled by electrical devices in most situations. Nevertheless, multiple channel capability is readily available in all-optical networks. As shown in Fig. 1, a K-degree node of a wavelength division multiplexing (WDM) network of W-wavelength will need K $W\times 1$ wavelength multiplexers and K $1\times W$ demultiplexers, KW 1-to-W wavelength converters, a $KW\times KW$ optical switch, and KW transmitters/receivers [22], [23]. If optical time division multiplexing (OTDM) instead of WDM is used to provide multichannel capability, the optical network will be single wavelength with W-timeslot OTDM frame. A K-degree node will only need a $K\times K$ optical switch, K transmitters/receivers, and an optical time slot interchanger (TSI) at each input of the optical switch. Owing to large optical TSI being not practical yet [24-27], OTDM-based deflection-routed optical networks in general provide fewer channels though it requires fewer optical components than that of WDM-based networks

Packet injection control will be an effective approach if the lowering of the whole network average deflection rate is desired [7-9]. Unlike other types of communication networks [5], [6], packets in deflection-routed networks can have dynamic routing paths due to packet deflection. Reducing the local packet injections at a node does not guarantee smaller packet deflection rates at its downstream nodes since the deflections may be caused by packets deflected from other nodes. Nodes in a deflection-routed network can benefit from packet injection control only if most of them have properly controlled their packet injections.

B. *Strict packet injection control*

We can improve the performance of a deflection-routed network without extra hardware by controlling the time and condition that a newly generated local packet is sent into the network. The simplest method is called strict packet injection control which does not require any network-wide status information. In strict packet injection control, a local packet will be rejected from entering the source node whenever its desired output is not available. We assume that the rejected local packets will be retransmitted from the user machines a random time later (please refer to Section III-B3 for detail). It is similar to the pre-routing access scheme in [7] and the transmit hold access technique in [8] but without requiring source buffering and with the capability to handle arbitrary network topology with non-uniform traffic distributions. By forcing the source nodes to send local packets only to the desired outputs, we guarantee that no local packets will be deflected at the sources. Once a packet enters the network, it may also encounter output contentions during its trip to the destination but the number of deflections will be substantially reduced because of the reduction in the number of deflected packets. Strict packet injection control can significantly improve the end-to-end packet throughput-delay performance, but local packets may have a large number of retransmissions at the sources.

C. *Token-bucket-based packet injection control*

The packet injection by strict control approach is rigid and conservative. No packet deflection at source is allowed in all loading ranges and traffic situations of the network. It also lacks the flexibility in adjusting packet retransmission rates even if we can tolerate some increment of end-to-end delay. Other packet injection control methods are therefore needed but they should not increase the implementation complexity by too much. Therefore, proposals assuming sophisticated processing such as session communications, end-to-end signaling, and collaboration between nodes will not be appropriate [9]. From the performance analysis (either by simulations or by the analytical models shown in Section III-B), we observe that strict packet injection control will have smaller retransmission rate and better end-to-end throughput-delay performance than that of networks without the injection control if the loading of the system is high. This suggests a new and simple packet injection control approach by mixing the two packet injection control approaches, i.e., the sources impose little control on packet injection when the system is lightly loaded and automatically tightens the packet injection control when the system loading increases. To simplify the implementation of such packet injection control, we adopt the idea from the token-bucket mechanism that has been widely used in ATM networks to regulate the traffic flows [5]. Packet deflections at the sources of the proposed deflection-routed networks will be controlled and regulated by a token-bucket-based method, an early version of which has been given in [28].

At each source, there is a counter TOKEN_POOL that increases with time of A_T tokens per timeslot until the tokens in the counter is equal to a predefined value POOL_SIZE. We define P as the required number of tokens for a source to handle a packet deflection/rejection. A local packet will need P tokens from the TOKEN_POOL to be sent (deflected) into the network if its desired output is not available. Unlike traditional token-bucket approaches, however, P tokens will also be subtracted from TOKEN_POOL even if the packet transmission fails, i.e., there is no available output. Note that both packet rejection and deflection consume P tokens in TOKEN_POOL. As long as tokens arrive at TOKEN_POOL at a constant rate, increase of packet rejections will automatically reduce packet deflections at the sources. The token arrival rate A_T in general is one token per timeslot but can be set to other values to refine the control of packet injection. For example, we can set $A_T = 2$, $P = 3$ to have the equivalent effect of $A_T = 1$, $P = 1.5$.

The token-bucket-based approach offers a simple and flexible way to control packet injections. For example, it becomes strict packet control if we set POOL_SIZE = 0 and P > 0, and no packet control if POOL_SIZE > 0 and P = 0. In general, a large P lowers the end-to-end delay but increases the local packet rejections at the sources. Large POOL_SIZE reduces the local packet retransmissions but causes large end-to-end delay. With proper settings of POOL_SIZE and P, the performance of the proposed method can be similar to that of normal deflection routing when loading is low and that of strict packet injection control when the loading is high. We will continue the discussion in Section IV-B.

III. PERFORMANCE OF MULTICHANNEL DEFLECTION ROUTING

We only show the derivation of analytical model for deflection-routed networks with strict packet injection control because the model without packet injection control has already been reported in [29]. Note that independency between timeslots is the main assumption required in the analytical modeling. Token-bucket-based packet injection control can cause high dependency between timeslots on output channels of nodes. We omit the analytical model for token-bucket-based method because of its inaccurate results. As no packet injection control and strict packet injection control can be treated as the two extreme cases of token-bucket-based packet injection control, an alternative is to use the results from the two analytical models (no control and strict control) as the performance bounds when discussing the performance of token-bucket-based packet injection control.

A. *The network model*

For illustration convenience, we adopted the terminologies of a WDM-based deflection-routed network. The results are also applicable to TSI-based deflection-routed networks. In the analytical performance model, the multichannel network is slotted. It has N nodes with arbitrary topology and W wavelength channels per link. One important feature of deflection-routed networks is that once a packet is admitted into the network, it will no longer be dropped [1]. To provide this packet lossless feature, the requirements of slotted deflection-routed networks introduced in Section II-A are assumed. Different nodes can have different degrees, i.e., numbers of input/output links. Packets are checked timeslot by timeslot at the input links of a node to determine whether the packets should be received or forwarded to the output links (for transit packets). A destination node with K-degree can receive up to KW packets per timeslot. However, we assume there are at most M local packets per timeslot generated at each node regardless of the node degree. Generally, M should not be larger than W times the minimum node degree of the network. Otherwise, the system will be easily congested. Conceptually, local packets are inserted into a node at each timeslot through a local fiber link with M wavelength channels. On the average, $\rho_{z,0}$ ($0 \le \rho_{z,0} \le 1$) packets arrived in a timeslot on each channel of the local fiber link of Node z, where $z = 1, ..., N$. We assume that a fraction $\alpha_{z,v}$ of the packets from the user(s) connected to Node z are sent to Node v, where $\alpha_{z,z} = 0$ and $\sum_{v \neq z} \alpha_{z,v} = 1$.

Local packets in different channels or timeslots of the local link are independent of each other. For networks without packet injection control, the node will insert all local packets into the network if the total number of transit packets and local packets is not larger than the number of output channels, i.e., KW channels of a K-degree node. For networks with strict packet injection control, however, only those local packets with their preferred output links available will be processed even if other output links are available. In either case, if the number of local packets exceeds the number of available output channels, some of the local packets will be randomly rejected from entering the source node, i.e., no source buffering is assumed. Note that we assume the rejected local packets will be kept at the user's machines and retransmitted to the source nodes a random time later. The detailed local packet retransmission model is provided in Section III-B3.

Each packet contains sufficient information for a node to determine the most suitable output link for the packet. We assume full wavelength conversion at the nodes. If more than one channel is available at the packet's preferred output link, a node will assign the packet to one of the channels at random. Similarly, all deflected transit packets and the deflected local packets (in networks without packet injection control) will be randomly assigned (deflected) to any available output channel. Note that randomized packet deflections not only provide fairness between packets of different destinations but also the most handy way to eliminate the live-lock problem, i.e., endless packet circulation in the network [30], [31]. It provides probabilistic livelock free guarantee, i.e., deflected packets have probability one of being finally deflected to destinations but no deterministic travelling time guarantee [13], [30], [31]. Other methods such as prioritizing packets by age may be used if deterministic livelock guarantee is desired [15], [17].

B. The analytical performance model

1) The throughput–delay performance

For a channel at input link I_i of Node z, $\ell_{z,i}(v)$ is defined as the probability of finding a packet destined for Node v in the channel, where z and v can be any node of the network, $i = 1,\ldots, K_z$, and K_z is the degree of Node z. The throughput $TH(v)$ of Node v is the average number of packets that Node v receives from all input links in a timeslot. $TH(v)$ can be computed as

$$TH(v) = W \times \sum_{i=1}^{K_v} \ell_{v,i}(v). \tag{1}$$

Note that Node v receives packets from all other nodes in the network. Since packets will not be lost once inside a deflection-routed network, $TH(v)$ in the steady state will be equal to the average total number of local packets entering the network per time unit from all other nodes destined for Node v. Therefore, Little's Law [32] will be a handy tool to compute the end-to-end packet travelling delay [3], [12-14]. Considering the whole network as a virtual queue with packets entering the virtual queue from all other nodes and leaving the virtual queue through Node v, we can compute the average end-to-end packet travelling delay $DELAY(v)$ from all other nodes to Node v as {the per unit time average total number of packets in network destined for Node v} / $TH(v)$, or

$$DELAY(v) = TH(v)^{-1} \times W \times \sum_{z=1}^{N} \sum_{i=1}^{K_z} L_z(i) \ell_{z,i}(v), \tag{2}$$

where $L_z(i)$ is the length (from the upstream node connecting to Node z with input link I_i, in number of timeslots) of input link I_i of Node z, and $W \times \ell_{z,i}(v) L_z(i)$ is the total number of packets destined for Node v at the W channels of input link I_i of Node z. $DELAY(v)$ is the average delay of the mixed packets from all other nodes destined for Node v. Equation (2) can also compute the average number of hops from all nodes to Node v if we set $L_z(i) = 1$ for all input links of all nodes.

Note that Little's Law provides the average packet delay only and it requires no packet loss inside the network. If we need the packet delay distribution or if packets may be lost inside the network, other approaches such as Markov chain modeling [4], [8] and packet age [15], [17] may be used but the computational requirement of these approaches will grow rapidly with network size.

2) The aggregated traffic method

From Eqs. (1) and (2), we can determine the throughput–delay performance of a multichannel deflection-routed network if we can solve the packet distribution probability $\ell_{z,i}(v)$ on all input links. To solve $\ell_{z,i}(v)$, the straightforward way is to define $\eta_{z,i}(v)$ as the probability of finding a packet destined for v at a channel of output link O_i of Node z. Surely, $\ell_{y,k}(v) = \eta_{x,h}(v)$ if the output link O_h of Node x is connected to the input link I_k of Node y. After determining all packet transfer probabilities between the input and output links of the nodes, we can then solve $\ell_{z,i}(v)$ on all input links iteratively, i.e., the results of $\eta_{x,h}(v)$ are substituted into $\ell_{y,k}(v)$ for the next round computation until the difference between the outcomes of two

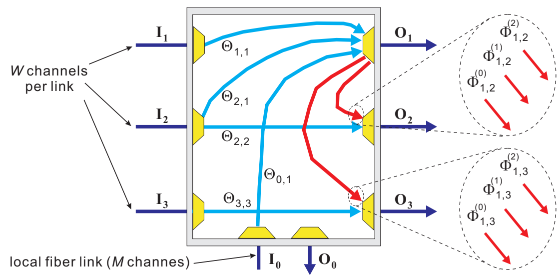

Fig. 2 An example of the aggregated incoming and deflection traffics inside a 3-degree node.

consecutive rounds is smaller than a predefined error tolerance. Details of numerically solving the probabilities can be found in Section III-B6.

Owing to the arbitrary network topology, the distribution probabilities of packets with different destinations on a link are in general different even if uniform loading is assumed for the network, i.e., $\ell_{z,i}(x) \neq \ell_{z,i}(y)$ if $x \neq y$. The computational requirement of directly solving $\ell_{z,i}(v)$ can be significant because we have to consider all interactions between different kinds of packets. There are at least $O(N^K)$ kinds of packet interactions for a K-degree node even if we assume one channel per link [17-19]. An alternative approach is to solve $\ell_{z,i}(v)$ indirectly from the aggregated traffics inside the nodes. Occasionally, computations with execution time of hours can be reduced to minutes using the aggregated traffic method [3].

As shown in Fig. 2, $\Theta_{i,k}$ is the aggregated traffic of packets from input link I_i with preferred output link O_k, and $\Phi^{(i)}_{k,h}$ is the aggregated traffic of packets from $\Theta_{i,k}$ but having been deflected to output link O_h. To simplify the illustration, we also define $\theta_{i,k}$ and $\phi^{(i)}_{k,x}$ as the probabilities of finding the packets of aggregated traffics $\Theta_{i,k}$ and $\Phi^{(i)}_{k,h}$ in a timeslot of a channel of links I_i and O_h of the node, respectively. Clearly, we have $\theta_{i,k} = \sum_{v\in \mathbf{C}_z(k)} \ell_{z,i}(v)$, where $\mathbf{C}_z(k)$ is the set of destinations for those packets at Node z requesting output link O_k. In Fig. 2, all packets in the aggregated traffics $\Theta_{0,1}$, $\Theta_{1,1}$,and $\Theta_{2,1}$ contend for the W channels of O_1 and some of them are deflected to O_2 and O_3, i.e., the aggregated traffics $\Phi^{(i)}_{1,2}$ and $\Phi^{(i)}_{1,3}$, where i = 0, 1, and 2. Note that solving for $\theta_{i,k}$ and $\phi^{(i)}_{k,x}$ of the aggregated traffics only requires us to consider $O(K^K)$ types of packet interactions. Actually, it will not be necessary to solve for the values of the aggregated traffic probabilities $\theta_{i,k}$ and $\phi^{(i)}_{k,x}$ for the link traffic distribution probability $\eta_{z,h}(v)$.

We first look at the busy probability of a timeslot on an output link x of Node z. The busy probability will be equal to $\sum_{v=1}^{N} \eta_{z,x}(v)$, or in the form of the aggregated traffic probability as $\sum_{i=1}^{K_z}\left[\theta_{i,x} + \sum_{k\neq x}\left(\phi^{(i)}_{k,x} - \phi^{(i)}_{x,k}\right)\right]$. Certainly, packets in the same aggregated traffic $\Theta_{i,k}$ will have the same transfer and deflection probabilities from input to output even if their destinations are different. For each input link I_i, we define $X_{z,i}(k, k)$ as the probability of packets included in the traffic $\Theta_{i,k}$ to be successfully transferred to their preferred output link O_k, and $X_{z,i}(k, h)$ as that of those to be deflected to an available output link $O_h \neq O_k$, i.e., joining the packets in traffic $\Phi^{(i)}_{k,h}$. Let the indicator function $g_{z,k}(v) = 1$ if output link O_k of Node z is the preferred output link of packets with destination v. Otherwise, $g_{z,k}(v) = 0$. Using the indicator function $g_{z,k}(v)$, we can represent the aggregated traffic probabilities without using the destination set $\mathbf{C}_z(k)$, e.g., $\phi^{(i)}_{k,x} = X_{z,i}(k,x)\sum_{v=1}^{N} \ell_{z,i}(v) g_{z,x}(v)$. After rewriting the busy probability of a timeslot on an output link x of Node z in terms of $X_{z,i}(k, x)$ and $g_{z,k}(v)$, we can derive the traffic distribution for a packet with destination v on an output link O_x as

$$\eta_{z,x}(v) = \sum_{k=1}^{K_z}\left[\frac{M}{W} X_{z,0}(k,x)\rho_{z,0,v} + \sum_{i=1}^{K_z} X_{z,i}(k,x)\ell_{z,i}(v)\right] g_{z,k}(v), \quad (3)$$

Note that O_x can be equal or not equal to O_k. $\rho_{z,0,v}$ is the probability of having a local packet with destination v in a timeslot of a channel on Node z's local fiber link. As local packets from the M channels of the local fiber link will be randomly sent to the W channels of an output link, we add a factor of M/W to the local packet traffic in Eq. (3) for computing $\eta_{z,h}(v)$. Note that $\sum_{x=1}^{K} X_{z,i}(k,x) = 1$ for $i \neq 0$, but $\sum_{x=1}^{K} X_{z,0}(k,x) \leq 1$. This is because all transit packets will be transferred to Node z's output links though some of them may not have obtained their preferred outputs. However, local packets will be rejected from entering into the network if all output links have been occupied by the transit packets.

To derive the $X_{z,0}(k, x)$ and $X_{z,i}(k, x)$ shown in Eq. (3), we first solve $r_{z,0,j}$, the fraction of all local packets arriving at Node z to have output preference O_j, j = 1, ..., K_z. $r_{z,0,j}$ in a system with packet injection control will depend on the final steady state traffic condition on links of the network and cannot be directly solved from $\alpha_{z,v}$. We will therefore first discuss the model for solving $r_{z,0,j}$ in Section III-B3 for systems with strict packet injection control. To simplify the presentation in the following derivation of the performance models, we will omit the subscript index of Node z in the equations, e.g., we will write $\ell_i(v)$ and $X_i(k, k)$ instead of $\ell_{z,i}(v)$ and $X_{z,i}(k, k)$.

3) Local packet generation model

Deflection-routed network performance analysis typically assumes that the local packet arrival (generation) rate ρ_0 per timeslot is constant [3], [4], [8], [12-17]. Also, the destination distribution α_v (or the ratios of local packets' output preferences $r_{0,j}$) of local packets does not change during the performance evaluation. Since local packets have lower priority than transit packets, they will not be assigned an output if the outputs are

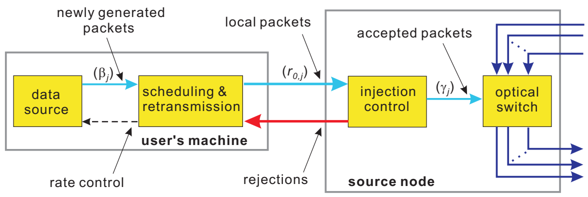

Fig. 3 Model of new packet generation: different kinds of new packet and ratios of output preference.

not sufficient for both kinds of packets. In such cases, the local packets will be rejected from entering the source nodes if either there is no source buffer [3], [4], [8], [12-14], [16], or the source buffers are full [15], [17]. For data communication networks, however, all rejected local packets in principle must be resent to the system to ensure no data loss. Hence, a reasonable interpretation should be the local packet generation model shown in Fig. 3. The rejected local packets are actually kept at the user machines and are resent to the source node but the total transmission process of local packets (the *local packets* in Fig. 3 including newly generated and retransmitted local packets) to the source node remains unchanged with the assumption that the inter-packet time and the average packet transmission rate ρ_0 are already set by constraints such as channel and processing capacities. We therefore define $\beta_j = \sum_{v\in \mathbf{C}(j)} \alpha_v$ as the original fraction of output preference O_j of the newly generated packets. $r_{0,j}$ is the ratios of output preferences of all local packets (including the newly generated and retransmitted ones) submitted to the node. γ_j is the fraction of output preferences of the accepted local packets that have been transferred into the system. For networks with strict packet injection control, a node will inject a local packet into the network only if a channel in the preferred output link is available. Owing to the different packet retransmission rates, $r_{0,j}$ may be different from β_j but γ_j should always be equal to β_j. Essentially, if the local packets with output preference O_x encounter different rejection rate than others, the output preference fraction $r_{0,x}$ of the local packets must take a value such that the output preference fraction γ_j of the accepted packets will remain the same as that of the newly generated packets β_j for $j = 1, \ldots, K$. Therefore, we can derive the required $r_{0,j}$ from the requirement of $\gamma_j = \beta_j$.

We define τ_j as the success probability of all local packets in Fig. 3 with output preference O_j to be sent into the network, i.e., $(1 - \tau_j)$ is rejection rate of the packets. As $(\gamma_j / \gamma_k) = (\beta_j / \beta_k)$, we have $(\rho_0 r_{0,j}\tau_j)/(\rho_0 r_{0,k}\tau_k) = (\beta_j / \beta_k)$. For networks with strict packet injection control, the success probability τ_j is equal to $X_0(j, j)$. We have

$$r_{0,j} = \frac{\tau_K \beta_j}{\tau_j \beta_K} r_{0,K}\text{, for } j = 1,\ldots, K-1. \tag{4}$$

As $r_{0,1} + \cdots + r_{0,K} = 1$, we further have

$$r_{0,K} = \left(1 + \frac{\tau_K}{\beta_K}\sum_{j=1}^{K-1}\frac{\beta_j}{\tau_j}\right)^{-1} \tag{5}$$

Using Eqs. (4) and (5), we can compute and update the local packet output preference ratios $r_{0,j}$ in each iteration when solving for the throughput and delay performance of deflection-routed networks with strict packet injection control. Once all $r_{0,j}$ have been solved, it is easy to compute $\rho_{0,v}$ ($\rho_{z,0,v}$ for Eq. (3)). For Node z, assuming O_j is the preferred output link of packets with destination v, i.e., $v \in \boldsymbol{C}_z(j)$, we have

$$\rho_{0,v} = \rho_0 r_{0,j}\frac{\alpha_v}{\beta_j}\,. \tag{6}$$

The number of retransmissions of local packets with output preference O_j in the user machine is a geometric random variable (of failures before the first success) with success probability τ_j. Hence, the average number of retransmissions of a local packet in the user machine before being admitted into the network can be simply computed by

$$\Lambda = \sum_{j=1}^{K}\frac{\beta_j}{\tau_j}(1-\tau_j) \tag{7}$$

where τ_j is equal to $X_0(j, j)$ as we discussed in Eqs. (4) and (5).

For networks without packet injection control, local packets with different destinations will encounter the same rejections. The packet retransmission will not change the output preference ratios of local packets from the newly generated packets, i.e., $r_{0,j} = \beta_j$ and $\gamma_j = \beta_j$ for $j = 1,\ldots, K$. Assuming constant ratios between local packets, an analytical model for no packet injection control has already been developed in [29]. Eq. (7) for a system without packet injection control will reduce to $\Lambda = (1 - \tau)/\tau$ where $\tau = \tau_j = \sum_{k=1}^{K} X_0^{(n)}(j,k)$ for any $1 \le j \le K$.

4) *The transfer probabilities of transit packets*

Transfer probability $X_i(k, k)$ with $i \ne 0$ is simply the ratio of $\Theta_{i,k}$ traffic that can be transferred to output link O_k. In Fig. 2, $X_1(1, 1)$ is equal to $(|\Theta_{1,1}| - |\Phi_{1,2}^{(1)}| - |\Phi_{1,3}^{(1)}|)/|\Theta_{1,1}|$, where $|x|$ is the average number of packets per timeslot of traffic x on a specified link. Assuming that packets in different timeslots and different wavelength channels are independent of each other, the number of transit packets arriving at input link I_i of a K-degree Node z in a slot time will be a binomial random variable of mean $W\rho_i$, where W is the number of wavelength channels per link and ρ_i is the average timeslot utilization of transit packets on input link I_i. The number of packets offered to the $\Theta_{i,k}$ traffic per time slot is also a binomial random variable of mean $W\rho_i r_{i,k}$, where $r_{i,k}$ is the fraction of transit packets having output preference O_k on input link I_i.

We only need to consider the interactions between transit packets when computing $X_i(k, k)$ because local packets do not affect channel reservation of transit packets. We define $m_{i,k}$ as the number of packets from $\Theta_{i,k}$ at a particular timeslot, where $0 \le m_{i,k} \le W$. Hence, a total of $m_{all,k} = m_{1,k} + \ldots + m_{K,k}$ packets (from aggregated traffics $\Theta_{1,k}$ to $\Theta_{K,k}$) will contend for the W output channels of O_k. The probability of any packet being blocked from entering output link O_k can be computed as

$B_k = \max(0, 1 - W/m_{all,k})$

We further define $\mathbf{m_{(i)}} = (m_{i,1}, m_{i,2}, \ldots, m_{i,K})$ as the status vector showing the output preference distribution of the packets on input link I_i. The probability distribution of $\mathbf{m_{(i)}}$ will be a multinomial distribution. As we mentioned before, the number of transit packets arriving at I_i in a timeslot $m_{i,all} = m_{i,1} + \ldots + m_{i,K}$ is a binomial random variable of mean $W\rho_i$, i.e.,

$$P_i(m_{i,all}) = \binom{W}{m_{i,all}} \rho_i^{\,m_{i,all}} (1-\rho_i)^{W-m_{i,all}} ,$$

where $0 \le m_{i,all} \le W$. The probability of having $m_{i,j}$ transit packets at input link I_i to have output preference O_j (j = 1 to K) can therefore be written as

$$F_i(\mathbf{m_{(i)}}) = P_i(m_{i,all}) \times \frac{m_{i,all}!}{m_{i,1}! \cdots m_{i,K}!} \prod_{j=1}^{K} r_{i,j}^{\,m_{i,j}} ,$$

where $\sum_{\mathbf{m_{(i)}}} F_i(\mathbf{m_{(i)}}) = 1$ for an input I_i. The average amount of traffic $\Theta_{i,k}$ that can be successfully transferred to output link O_k under all conditions can be written as $\sum_{\sigma_1 \le W} \cdots \sum_{\sigma_K \le W} (1-B_k)\, m_{i,k} \prod_{j=1}^{K} F_j(\mathbf{m_{(j)}})$. As $|\Theta_{i,k}| = W\rho_i r_{i,k}$, we can simply write $X_i(k, k)$ with $i \neq 0$ as

$$X_i(k,k) = \frac{1}{W\rho_i r_{i,k}} \sum_{m_{1,all} \le W} \cdots \sum_{m_{K,all} \le W} (1-B_k) m_{i,k} \prod_{j=1}^{K} F_j(\mathbf{m_{(j)}}) , \quad (8)$$

In principle, we may similarly derive $X_i(k, h)$ because it is by definition equal to $|\Phi_{k,h}^{(i)}|/|\Theta_{i,k}|$. To solve $|\Phi_{k,h}^{(i)}|$, however, we have to consider the status of all output links instead of only O_k and O_h. For example, one can observe that both $|\Phi_{1,2}^{(1)}|$ and $|\Phi_{1,3}^{(1)}|$ in Fig. 2 will change with $\Theta_{2,2}$ and $\Theta_{3,3}$ under the relatively simple case of local packets having only output preference of O_1. Following the style of Eq. (8), we can write $X_i(k, h)$ with $i \neq 0$ for $h \neq k$ as

$$X_i(k,h) = \frac{1}{W\rho_i r_{i,k}} \sum_{m_{1,all} \le W} \cdots \sum_{m_{K,all} \le W} m_{i,k} B_k U_h \prod_{j=1}^{K} F_j(\mathbf{m_{(j)}}) , \quad (9)$$

where U_h is the probability for the blocked packets from output O_k to be deflected to O_h. Note that U_h is independent of the outputs blocking the packets but depends on the ratio of the unreserved channel capacities of the output links. To derive U_h, we need to know the number of channels unreserved by all packets (transit and local) for each output O_j. This is equal to $\max(0, W - q_j - m_{all,j})$, where q_j is the number of accepted local packets with output preference O_j. If $m_{0,j}$ is defined as the number of local packets with output preference O_j generated at Node z's local fiber link, q_j will be smaller than or equal to both $m_{0,j}$ and the number of channels on O_j not reserved by transit packets, i.e., $q_j \le u_j = \min(m_{0,j}, \max(0, W - m_{all,j}))$ for $j = 1, \ldots, K$. Note that the probability distribution of $m_{0,j}$ can be solved by evaluating $\mathbf{m_{(0)}}$ and $F_0(\mathbf{m_{(0)}})$ similarly to that of $\mathbf{m_{(i)}}$ and $F_i(\mathbf{m_{(i)}})$ in Eq. (8), i.e.,

$$F_0(\mathbf{m_{(0)}}) = \frac{\rho_0^{\,m_{0,all}} (1-\rho_0)^{M-m_{0,all}} m_{0,all}!}{m_{0,1}! \cdots m_{0,K}!} \binom{M}{m_{0,all}} \prod_{j=1}^{K} r_{0,j}^{\,m_{0,j}} ,$$

where M is the number of channels in the node's local fiber link, and $m_{0,\text{all}}$ is the total number of local packets. We define $\mathbf{q} = (q_1, q_2, \ldots, q_K)$ as the vector showing the status of a node's accepted local packets. $Q(\mathbf{q}|(\mathbf{m_{(0)}}, C))$ is the probability to have a specified $\mathbf{q}$ in a system conditioned on $\mathbf{m_{(0)}}$ when the total available output channel capacity C is smaller than the number of local packets. $Q(\mathbf{q}|(\mathbf{m_{(0)}}, C))$ can be computed as

$$Q(\mathbf{q}|(\mathbf{m_{(0)}}, C)) = \left(\prod_{j=1}^{K} \binom{u_j}{q_j} \right) \binom{u_1 + \ldots + u_K}{C}^{-1} ,$$

We further define $S(\mathbf{u}, \mathrm{C})$ as the possible set of $\mathbf{q}$ for a specified upper bound value vector $\mathbf{u} = (u_1, \ldots, u_K)$ when the number of available output channels is C. The probability U_h then becomes

$$U_h = \sum_{m_{0,all} \le M} \left[F_0(\mathbf{m_{(0)}}) \sum_{\mathbf{q} \in S(\mathbf{u},C)} \left(\omega_h Q(\mathbf{q} \mid (\mathbf{m_{(0)}}, C)) \right) \right] , \quad (10)$$

where ω_h is defined as the ratio of the number of unreserved channels of output link O_h to the total number of unreserved channels on all output links, i.e.,

$$\omega_h = \max(0, W - q_h - m_{all,h}) / \sum_{j=1}^{K} \max(0, W - q_j - m_{all,j}) .$$

For a 3-degree node with $W = 4$, $\mathbf{m_{(0)}} = (2, 1, 1)$, $\mathbf{m_{(1)}} = (1, 1, 2)$, $\mathbf{m_{(2)}} = (0, 1, 2)$ and $\mathbf{m_{(3)}} = (0, 0, 3)$, the node can accept only two local packets due to its ten transit packets. Hence, the possible set of $\mathbf{q}$ will be {(1, 1, 0), (2, 0, 0)} for $\mathbf{u}$ = (2, 1, 0). $Q(\mathbf{q}|(\mathbf{m_{(0)}}, C))$ and ($\omega_1\ \omega_2\ \omega_3$) will be {⅔, ⅓}, and {(⅔, ⅓, 0), (⅓, ⅔, 0)} if $\mathbf{q}$ = (1, 1, 0) and (2, 0, 0), respectively.

5) The transfer probabilities of local packets

The lower priority of local packets in output channel reservation complicates the expression of $X_0(k, k$. Let $n_k = \min(W, m_{all,k})$ be the number of channels on O_k reserved by transit packets. The probability of n_k channels at output link O_k, $k = 1, \ldots, K$, having been reserved by transit packets when the total number of transit packets is $m_{trans} = m_{1,all} + \ldots + m_{K,all}$, can be computed from the arrival probability distributions $F_i(\mathbf{m_{(i)}})$ of the transit packets as

$$R(\mathbf{n}, m_{trans}) = \sum_{(\mathbf{m_{(1)}}, \cdots, \mathbf{m_{(K)}}) \in T(\mathbf{n}, m_{trans})} \prod_{i=1}^{K} F_i(\mathbf{m_{(i)}}) . \quad (11)$$

To simplify the notations of Eq. (11), we have defined $\mathbf{n} = (n_1, n_2, \ldots, n_K)$. $T(\mathbf{n}, m_{trans})$ is the set of combinations of transit packet status vectors $(\mathbf{m_{(1)}}, \ldots, \mathbf{m_{(K)}})$ when n_j output channels of O_j have been reserved by a total of m_{trans} transit packets. In

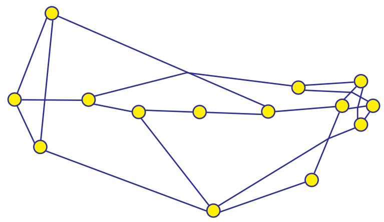

Fig. 4 The NSFNet (1991) network topology. The original map of the network is available from the Internet (ftp://ftp.uu.net/inet/maps/nsfnet/).

networks with strict packet injection control, the average number of local packets being transferred to O_k is the sum of q_k under all arrival situations of transit and local packets. We can therefore have the probability $X_0(k, k)$ of a local packet successfully obtaining its preferred output link O_k as

$$X_0(k,k)=\frac{1}{M\rho_0 r_{0,k}}\sum_{m_{0,all}\le M}\left[F_0(\mathbf{m}_{(\mathbf{0})})\sum_{m_{trans}=0}^{KW}\sum_{n_1+\cdots+n_K\le m_{trans}}\left[R(\mathbf{n},m_{trans})\sum_{\mathbf{q}\in S(\mathbf{u},C)}\left[q_k Q(\mathbf{q}\,|\,(\mathbf{m}_{(\mathbf{0})},C))\right]\right]\right]. \quad (12)$$

Note that the $r_{0,k}$ in the $F_0(\mathbf{m}_{(\mathbf{0})})$ of Eqs. (12) should be computed by using Eqs. (4) and (5) in Section III-B3. As we discussed before, $|\Phi_{k,h}^{(0)}|= 0$ with strict packet injection control because there is no blocked local packet deflected from any output link. Hence, for all $h \neq k$, we have

$$X_0(k, h) = 0 \quad (13)$$

6) Numerical solution of the probabilities

We have a set of $O(NK^K)$ nonlinear equations but we are seldom able to solve them exactly unless both N and K are small. Therefore, fixed-point iteration approach is used to numerically solve the equations [33]. The procedure of obtaining the throughput performance of deflection-routed networks is as follows:

1. Initialize the values of $\rho_{z,0}$, $\alpha_{z,n}$, and $\beta_{z,j}$ for all nodes (each Node z, z = 1, …, N) in the network as described in Section III-B3.
2. Initialize $\ell_{z,k}(v) = 0$ for all links of all nodes.
3. Initialize $X_{z,i}(k,x) = 0$ for all nodes.
4. Assign prev_$X_{z,i}(k,x)$= $X_{z,i}(k,x)$ for all nodes.
5. Update $r_{z,0,k}$ for all nodes using Eqs. (4) and (5).
6. Update $r_{z,i,k}$ for all nodes as described in Section III-B4 and III-B5.
7. Compute the new values of $X_{z,i}(k,x)$ node by node using Eqs. (8) to (13).
8. Update $\eta_{z,k}(v)$ ($\ell_{z,k}(v)$) for each link of each node in the network using Eqs (3) and (6).
9. Compare the difference between prev_$X_{z,i}(k,x)$ and $X_{z,i}(k,x)$ for all nodes. If the maximum difference among them is not smaller than a predefined error tolerance, go to Step 4.
10. Compute the throughput-delay performance of each node using Eqs. (1) and (2).

As the fixed-point iteration approach is used to numerically obtain the results, the convergence time (number of iterations) required for $X_{z,i}(k,x)$ to approach their final values will dominate the throughput-delay computation. Like other network performance evaluations using fixed-point methods [33], the model will take longer to converge in some situations such as large network size, irregular topology, high system loading, and non-uniform traffic distribution. For the networks used in Section IV, however, the proposed model requires only several tens of iterations to obtain the results.

IV. PERFORMANCE EVALUATION

A. Accuracy of the models

We use simulations on the NSFNet (Fig. 4) network topologies to demonstrate the accuracy of the model we derived in Section III-B. In the simulations, we make all the assumptions of the network model in Section III-A, i.e., random packet arrivals, deflections, and retransmissions. Since the analytical model in Section III-B further makes other assumptions such as independent transit traffic and the local packet generation, it will not provide the correct result if such assumptions are not correct. Hence, we can therefore also verify the correctness of such assumptions made in Section III-B. Minimum hop count routing is used to assign the packets' desired output links for each node. At a source node, a newly generated packet (please refer to Fig. 3) will randomly select one destination from the rest of the nodes. The link propagation time is proportional to the link length with a minimum of 20 units in NSFNet. Figures 5 to 6 show the analytical and simulation throughput to end-to-end delay curves of the deflection routings. The analytical results of strict packet injection control in Fig. 6 are from Section III-B. Those with no packet injection control in Fig. 5 are from the model reported in [29], and are included for comparison. In the figures, the end-to-end delay is in number of hops, i.e., the number of links a packet passed before arriving at its destination. Hence, we can have a comparison of the end-to-end delay performance even if the topology and link propagation delay of the networks are different. The normalized throughput is the total number of packets received by the nodes in a timeslot divided by the number of nodes N and the number of channels per link W. The maximum number of local packet arrivals per node M is equal to $2W$, where 2 is the minimum node degree of NSFNet. Hence, each of the $2W$ channels of the node's local fiber link will generate a local packet per timeslot in the node with probability (loading) ρ_0. We increase the loading ρ_0 from 0.01 to 1.0, and record the throughput and end-to-end delay values. Note that the recorded throughput in principle can be larger than 1 (but must not be larger than M/W = 2) according its definition though we have not observed this in the simulation results.

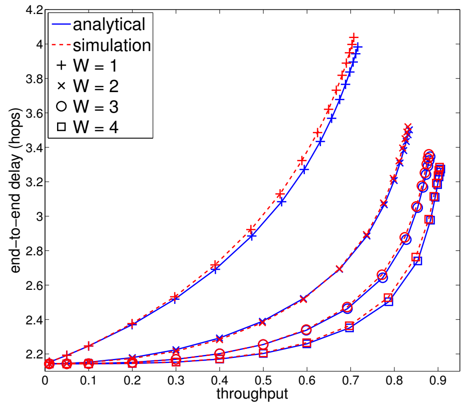

Fig. 5 The end-to-end throughput – delay curves of deflection routing on the NSFNet (Fig. 4) without packet injection control.

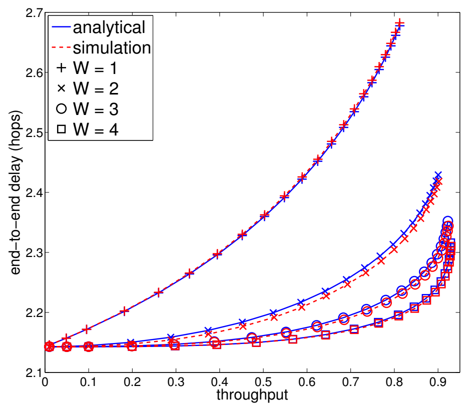

Fig. 6 The end-to-end throughput – delay curves of deflection routing on the NSFNet (Fig. 4) with strict packet injection control.

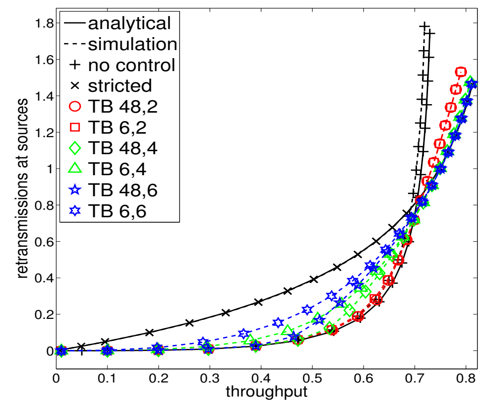

Fig. 7 The average packet retransmissions at sources (W = 1) of the NSFNet topology (Fig. 4) network with different packet injection control methods.

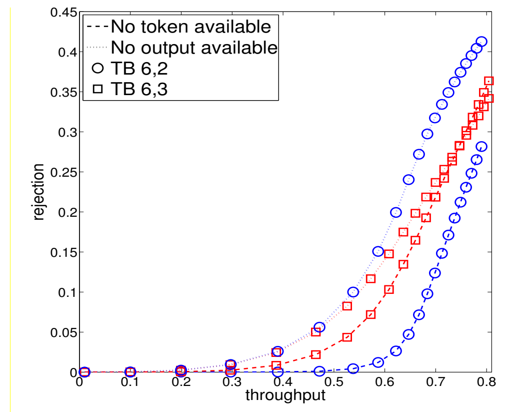

Fig. 8 The rejection rates (W = 1) of the NSFNet topology (Fig. 4) network with token-based injection control methods of POOL_SIZE = 6 and P = 2, 3.

In Figs. 5 and 6, the curves with pluses, crosses, circles, and squares are the results from networks with one, two, three, and four channels per link, respectively. We use solid curves for analytical results, and dashed curves for simulations. From the figures, the results from the analytical models generally match those of simulations very well. It shows that the traffics in different links are almost independent of each other regardless of the network topologies (we have observed similar results in 4×4 MSN [34]), and the number of wavelength channels per link. As shown in Fig. 5 (Fig. 6), we will have around 0.92/0.73 ≈26% (0.93/0.81≈15%) maximum throughput improvement if we send data using four 10 Gbps channels per link instead of with a single 40 Gbps channel per link.

B. *Performance improvement with packet injection controls*

Comparing Figs. 5 and 6, we observe that strict packet injection control can significantly improve the system throughput for the same end-to-end delay for all values of W though the improvement decreases with the increase of W because systems will have smaller number of defections per packet when W is large. However, the performance comparison will be incomplete if we do not consider the number of retransmissions encountered by local packets at the sources (please refer to Fig. 3). To show the features of different packet injection control methods, we focus on the performance evaluation of single channel (W = 1) deflection-routed networks. The numbers of local packet retransmissions in systems with and without packet injection control are obtained from both Eq. (7) and simulations. The throughput, end-to-end delay, and retransmission results for the token-bucket-based packet injection control are only from simulations.

In Fig. 7, the curves with pluses and crosses are the throughput to local packet retransmission curves of NSFNet topology networks with normal deflection routing and that with strict packet injection control, respectively. The curves with squares, triangles and hexagrams are those of deflection routing with token-bucket-based packet injection control of POOL_SIZE = 6, P = 2, 4 and 6, respectively. The curves with circles, diamonds and pentagrams are those of POOL_SIZE = 48, P = 2, 4 and 6, respectively. In the simulations, we assume that the token-bucket-based packet injection control counter TOKEN_POOL increases at a rate A_T of one token per timeslot until it reaches the value of POOL_SIZE.

From Fig. 7, normal deflection routing has smaller local

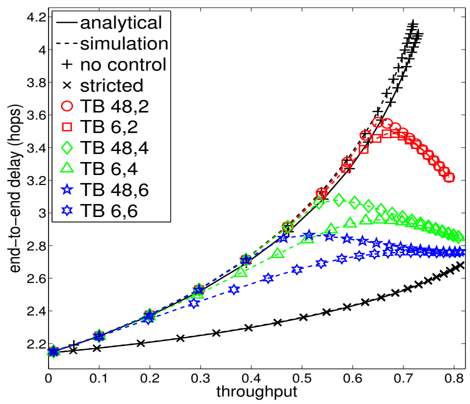

Fig. 9 The end-to-end throughput-delay curves (W = 1) of the NSFNet topology (Fig. 4) network with different packet injection control methods

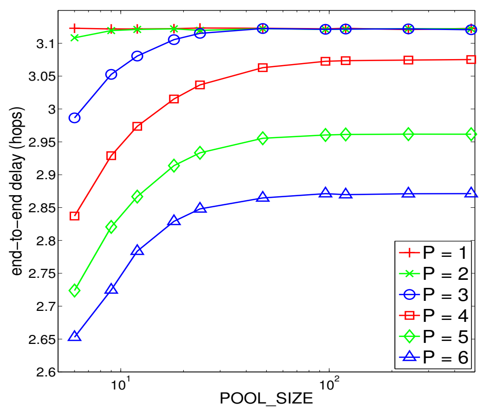

Fig. 10 The end-to-end delay (W = 1) of the NSFNet topology (Fig. 4) network using token-bucket-based packet injection control with different token pool sizes. The throughput of the system without packet injection control is 0.54.

packet retransmission rate than that of strict packet injection control if the throughput is below a threshold, e.g., 0.69. It has almost no local packet retransmissions at the user machines when the throughput is low, e.g., 0.4. Thus, in general, strict packet injection control should only be used in high throughput (system loading) range though it has much better end-to-end throughput-delay performance than that of normal deflection routing. Surely, strict packet injection control can also be used in the low throughput range if the link propagation delay is much larger than the time between packet retransmissions at the user machines because the end-to-end delay has dominated the total delay.

The packet retransmission rates of token-bucket-based packet injection control are between that of normal deflection routing and that of strict packet injection control when the throughput is below the threshold value, e.g., 0.69 in Fig. 7. It has larger values for small POOL_SIZE and large P (e.g., TB 6,4 has larger retransmission rate than TB 48,4 and TB 6,4 has larger retransmission rate than TB 6,2 in Fig. 7) and then becomes similar to that of strict packet injection control when throughput increases. In the heavily loaded systems, however, the token-bucket-based packet injection control with small P can have larger packet retransmission rates than that of large P, e.g., TB 6,2 has larger retransmission rate than TB 6,6 in Fig. 7 when throughput is above 0.69. This phenomenon is caused by the mixture of two kinds of local packet rejections, i.e., the first is caused by lack of tokens and the second by no available output. Figure 8 shows the two kinds of rejections (W = 1) of the NSFNet topology (Fig. 4) network with token-bucket-based packet injection control of POOL_SIZE = 6 and P = 2, 3. A token-bucket-based injection control with larger P will have a larger portion of rejection of the first kind. Although it has larger portion of rejection of the first kind, the increasing rate of rejection of the second kind with the throughput increase is smaller because rejections of the first kind will automatically reduce the number of deflected packets being sent into the network, as we discussed in Section II-B. The retransmissions in Fig. 7 are caused by both kinds of rejections and therefore can have small values with larger P at high throughput values. To determine the proper working parameters of token-bucket-based injection control method, we further check its throughput to end-to-end delay performance.

From Fig. 9, all token-bucket-based packet injection control systems have throughput-delay curves between that of normal deflection routing and that of strict packet injection control. In contrast with retransmission rates, their end-to-end delays increase with large POOL_SIZE and small P. Moreover, the delay decreases when throughput is above a threshold value, e.g., 0.45 for TB 48,6. As we have mentioned in Section II-B, both packet rejection and deflection at source consume P tokens from TOKEN_POOL. As long as tokens arrive at TOKEN_POOL at a constant rate, the increase in local packet rejection will automatically reduce the packet deflections. Hence, increasing the throughput (system loading) will increase the local packet rejections and therefore reduce the packet deflections at source, i.e., it shortens the end-to-end delay as shown in Fig. 9. Using a large P will have lower end-to-end delay but also more local packet rejections at the sources in low throughput range. On the other hand, large POOL_SIZE can reduce required local packet retransmissions at user machines but at the expense of more deflected packets in the network, i.e., larger end-to-end delay. A rule of thumb for choosing these values is needed.

Figure 10 shows the six end-to-end delay curves (P = 1, …, 6) of a deflection-routed NSFNet network using token-bucket-based packet injection control when POOL_SIZE is increasing from 6 to 450. The system will have throughput 0.54 if it is without packet injection control, i.e., the largest throughput of normal deflection routing in the stable region of Fig. 7. From Fig. 10, one can observe that the curves with different P values will have marked differences in end-to-end delay compared to the increments caused by increasing POOL_SIZE. According to this observation, we may first choose a large P to minimize the end-to-end delay and then use a large value of POOL_SIZE to reduce the retransmissions to an acceptable value, e.g., the token-bucket-based injection control with POOL_SIZE = 48 and P = 6 as shown in Figs. 7 and 9.

V. CONCLUSION

In packet-switched networks with limited or no buffers, deflection routing is one of the feasible approaches for output contention resolution. In this paper, we propose to combine multichannel approach with packet injection control to improve the performance of deflection-routed networks. Two methods, strict and token-bucket-based packet injection control have been proposed and discussed. To simplify the performance evaluation, the analytical performance model of strict packet injection control on multichannel deflection-routed networks is derived. From the results of simulations and analytical performance models, we observe that the proposed simple packet injection control methods can improve the end-to-end throughput-delay and packet re-transmission performance without substantially increasing the network implementation complexity.

ACKNOWLEDGMENT

This research was supported in part by a grant from the Hong Kong Polytechnic University (Project Number J-BB9M), and in part by the University of Hong Kong Strategic Research Theme of Information Technology.

APPENDIX – LIST OF SYMBOLS

B_k	the probability of any packet being blocked from entering output link O_k
$\mathbf{C}_z(k)$	the set of destinations for those packets at Node z requesting output link O_k of Node z
$DELAY(v)$	the average packet end-to-end travelling delay from all other nodes to Node v
$F_i(\mathbf{m_{(i)}})$	the probability distribution of the status vector $\mathbf{m_{(i)}}$ of input link I_i, i.e., $\sum_{\mathbf{m_{(i)}}} F_i(\mathbf{m_{(i)}}) = 1$.
$g_{z,k}(v)$	Indicator variable. It is equal to 1 if output link O_k of Node z is the preferred output link of packets with destination v; otherwise, it is equal = 0.
K	the degree of a node
K_v	the degree of Node v
$L_z(i)$	the length (in number of timeslots) of input link I_i of Node z
M	the maximum number of local packets that can arrive at a node per timeslot
$m_{all,k}$	the total number of packets arriving at all input link I_i in a timeslot for the output O_k
$m_{i,k}$	the number of packets from aggregated traffic $\Theta_{i,k}$ at a particular timeslot
$m_{i,all}$	the total number of packets arriving at input link I_i in a timeslot
m_{trans}	the total number of transit packets
$\mathbf{m_{(i)}}$	the status vector $(m_{i,1}, m_{i,2}, \ldots, m_{i,K})$ showing the output preference distribution of the transit packets on input link I_i
N	the number of nodes in network
n_k	the number of channels at output link O_k reserved by transit packets
$\mathbf{n}$	the status vector $(n_1, n_2, \ldots, n_K)$ of output link channels being reserved by transit packets
q_j	the number of accepted local packets in a timeslot with output preference O_j
$\mathbf{q}$	the status vector $(q_1, q_2, \ldots, q_K)$ showing the distribution of the node's accepted local packets
$Q(\mathbf{q}\|\mathbf{m_{(0)}},C)$	the conditional probability of $\mathbf{q}$ given the upper bound of $\mathbf{q}$ is $\mathbf{m_{(0)}}$
$R(\mathbf{n}, m_{trans})$	the probability of n_k channels at output link O_k, $k = 1, \ldots, K$, being reserved by transit packets when the total number of transit packets is m_{trans}
$r_{i,k}$	the fraction of transit packets having output preference O_k on input link I_i
$r_{z,0,j}$	the fraction of local packets arriving at Node z to have output preference O_j
$S(\mathbf{u}, C)$	the possible set of $\mathbf{q}$ for a specified upper bound value vector $\mathbf{u}$ when the number of available output channels is C
$TH(v)$	the average number of packets Node v receives in a timeslot from all input links
$T(\mathbf{n}, m_{trans})$	the set of combinations of transit packet status vectors $(\mathbf{m_{(1)}}, \ldots, \mathbf{m_{(K)}})$ when reserved output link channel status vector is $\mathbf{n}$ and total number of transit packet is m_{trans}
U_h	the probability that a blocked transit packet is deflected to O_h
u_j	the upper bound of q_j in a strict packet injection control system
$\mathbf{u}$	the upper bound vector $(u_1, \ldots, u_K)$ for $\mathbf{q}$ in a strict packet injection control system
W	the number of wavelength channels per link
$X_{z,i}(k, h)$	the probability of packets in aggregated traffic $\Theta_{i,k}$ to be transferred to an available output link O_h
$\alpha_{z,v}$	the fraction of newly generated local packets of Node z with destination of Node v
β_j	$\beta_j = \sum_{v\in \mathbf{C}(j)} \alpha_v$ the fraction of the newly generated packets with output preference O_j
γ_j	the fraction of accepted local packets with output preference O_j
$\eta_{z,i}(v)$	the probability of finding a packet destined for Node v at a channel of output link O_i of Node z
ρ_i	the average timeslot utilization of transit packets on input link I_i
$\rho_{z,0}$	the probability of having a local packet in a timeslot of a channel on Node z's local fiber link
$\rho_{z,0,v}$	the probability of having a local packet with destination v in a timeslot of a channel on Node z's local fiber link
$\ell_{z,i}(v)$	the probability of finding a packet destined for Node v at a channel of input link I_i of Node z
τ_j	the probability of local packets with output preference O_j to be successfully sent into the network
$\Theta_{i,k}$	the aggregated traffic of packets from input link I_i with preferred output link O_k
$\theta_{i,k}$	the probability of finding a packet of aggregated traffic $\Theta_{i,k}$ in a timeslot of a channel of link I_i of a

	node
$\Phi_{k,h}^{(i)}$	the aggregated traffic of packets from $\Theta_{i,k}$ but having been deflected to output link O_h
$\phi_{k,x}^{(i)}$	the probability of finding a packet of aggregated traffic $\Phi_{k,h}^{(i)}$ in a timeslot of a channel of output link O_h of a node
Λ	the average number of retransmissions of a local packet in the user machine before passing the node to the network
ω_h	the ratio of the number of unreserved channels of output link O_h to the total number of unreserved channels on all output links

REFERENCES

[1] N. F. Maxemchuk "Regular and mesh topologies in local and metropolitan area neworks," AT&T Technical Journal, Vol. 64, No. 7, pp. 1659-1686, 1985.

[2] A. S. Acampora, and S. I. A. Shah, "Multihop lightwave networks: A comparison of store-and-forward and hot-potato routing," *IEEE Transactions on Communications*, Vol. 40, No. 6, pp. 1082-1090, 1992.

[3] C. Y. Li, P. K. A. Wai, X. C. Yuan, and V. O. K. Li, "Deflection routing in slotted self-routing networks with arbitrary topology," *IEEE Journal of Selected Area on Communications*, Vol. 22, No. 9, pp. 1812-1822, November 2004.

[4] A. Bononi, G. A. Castonon, and O. K. Tonguz, "Analysis of hot-potato optical networks with wavelength conversion," *Journal of Lightwave Technology*, Vol. 17, No. 4, pp. 525 – 534, 1999.

[5] D. E. McDysan, *ATM & MPLS theory & application: foundation of multi-service networking*, McGraw-Hill, New York, 2002.

[6] D. Wang and G. Li, "Efficient distributed bandwidth management for MPLS fast reroute," *IEEE/ACM Transactions on Networking*, Vol. 16, No. 2, pp. 468-495, 2008.

[7] T. D. Todd and A. M. Bignell, "Traffic processing algorithms for the SIGnet metropolitan area network," *IEEE Transactions on Communications*, Vol. 40, No. 3, pp. 568-576, 1992.

[8] A. Bononi and P. P. Prucnal, "Analytical evualuation of improved access techniques in deflection routing networks," *IEEE/ACM Transactions on Networking*, Vol. 4, No. 5, pp. 726-730, 1996.

[9] W. Olesinski and P. Gburzynski, "Service guarantees in deflection networks," *Proceedings of Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOT 2001)*, pp. 267-274, 2001.

[10] P. Baran, "On distributed communications networks", *IEEE Transactions on Communications Systems*, Vol. 12, No. 1, pp 1-9, 1964.

[11] W. Hillis, *The Connection Machine*, MIT Press, 1989.

[12] J. Bannister, F. Borgonovo, L. Fratta, and M. Gerla, "A versatile model for predicting the performance of deflection-routing networks," *Performance Evaluation*, Vol. 16, pp. 201-222, 1992.

[13] A. G. Greenberg and J. Goodman, "Sharp approximate models of deflection routing in mesh networks," *IEEE Transactions on Communications*, Vol. 41, No. 1, pp. 210-223, 1993.

[14] A. K. Choudhury and V. O. K. Li "An approximate analysis of the performance of deflection routing in regular networks," *IEEE Journal on Selected Areas in Communications*, Vol. 11, No. 8, pp. 1302-1316, 1993.

[15] Z. Zhang and A. Acampora, "Performance analysis of multihop lightwave networks with hot potato routing and distance-age priorities," *IEEE Transactions on Communications*, Vol. 40, pp.2571-2581, 1994.

[16] J. Brassil and R. Cruz, "Non-uniform traffic in Manhattan street network," *Performance Evaluation*, Vol. 25, pp. 233-246, 1996.

[17] E. Noel and K. W. Tang, "Peformance modeling of multihop network subject to uniform and nonuniform geometric traffic," *IEEE/ACM Transactions on Networking*, Vol. 8, pp.763-774, 2000.

[18] F. Borgonovo, L. Fratta, J. Bannister, "Unslotted deflection routing in all-optical networks," *Proceedings of Globecom 1993*. Vol. 1, pp. 119-125, 1993.

[19] T. Chich, J. Cohen, and P. Fraigniaud, "Unslotted deflection routing: A practical and efficient protocol for multihop optical networks," *IEEE/ACM Transactions on Networking*, Vol. 9, No. 1, pp. 47-59, 2001.

[20] C. F. Hsu, T. L. Liu, and N. F. Huang, "Performance analysis of deflection routing in optical burst-switched networks," *Proceedings of Infocom 2002*, Vol. 1, pp. 66-73, 2002.

[21] T. Phung-Duc, H. Masuyama, S. Kasahara, and Y. Takahashi, "Performance Analysis of Optical Burst Switched Networks with Limited-Range Wavelength Conversion, Retransmission and Burst Segmentation," *Journal of the Operations Research Society of Japan*, Vol. 52, No. 1, pp. 58--74, 2009.

[22] S. Sygletos, I. Tomaos, and J. Leuthold, "Technological challenges on the road toward transparent networking," *Journal of Optical Networking*, Vol. 7, No. 4, pp. 321-350, 2008.

[23] K. Vlachos, et al., "Photonics in switching: enabling technologies and subsystem design," *Journal of Optical Networking*, Vol. 8, No. 5, pp. 404-428, 2009.

[24] K. Onohara, H. Sotobayashi, K. I. Kitayama, and W. Chujo, "Photonic timeslot and wavelength-grid interchange for 10-Gb/s packet switching," *IEEE Photonics Technology Letters*, Vol. 13, No. 10, pp. 1121-1123, 2001.

[25] O. F. Yilmaz, et al., "Timeslot interchange of 40 Gbits/s variable length optical packets using conversion–dispersion-based tunable delays," *Optics Letters*, Vol. 33, Issue 17, pp. 1954-1956, 2008.

[26] C. Y. Li, P. K. A. Wai, and V. O. K. Li, "Multi-slot batch-transfer optical packet switch," *Proceedings of 50th Annual IEEE Global Communications Conference (Globecom 2007)*, pp. 2364 – 2368, Washington, DC, USA, 26-30 November 2007.

[27] C. Y. Li, P. K. A. Wai, and V. O. K. Li, "Parallel transfer optical packet switches," *IEEE Journal of Lightwave Technology*, Vol. 27, No. 12, pp. 2159-2168, 2009.

[28] C. Y. Li, and P. K. A. Wai, "A token bucket method for packet injection control in deflection-routed optical networks" *Proceedings of Opto-Electronics and Communications Conference (OECC 2010)*, Sapporo, Japan, 5-9 July 2010.

[29] C. Y. Li, P. K. A. Wai, and V. O. K. Li, "Performance model of deflection-routed multi-slot batch-transfer networks," *Proceedings of 51th Annual IEEE Global Communications Conference (Globecom 2008)*, paper ON03T3-4, 2008.

[30] N.F. Maxemchuk, "Problems arising from deflection routing: live-lock, lockout, congestion and message reassembly, *Proceedings of NATO Workshop on Architecture and Performance Issues of High Capacity Local and Metropolitan Area Networks*, pp. 209-234, France June 1990.

[31] A. K. Choudhury and V. O. K. Li, "Effect of contention resolution rules on the performance of deflection routing," *Proceedings Globecom 91*, pp. 1706-1711, 1991.

[32] J. D. C. Little, "A proof of the queuing formula: $L = \lambda W$," *Operations Research*, Vol. 9, pp. 383-387, 1967.

[33] W. C. Rheinboldt, *Methods for solving systems of nonlinear equations 2nd*, Siam, 1998.

[34] N. F. Maxemchuk, "Routing in Manhattan Street network," *IEEE Transactions on Communications*, Vol. 35, No. 5, pp. 503-512, 1987.