Dynamic Reconfiguration and Routing Algorithms for IP-Over-WDM Networks With Stochastic Traffic

Andrew Brzezinski, *Student Member, IEEE*, and Eytan Modiano, *Senior Member, IEEE*

***Abstract*—We develop algorithms for joint IP-layer routing and WDM logical topology reconfiguration in IP-over-WDM networks experiencing stochastic traffic. At the wavelenght division multiplexing (WDM) layer, we associate a nonnegligible overhead with WDM reconfiguration, during which time tuned transceivers cannot service backlogged data. The Internet Protocol (IP) layer is modeled as a queueing system. We demonstrate that the proposed algorithms achieve asymptotic throughput optimality by using frame-based maximum weight scheduling decisions. We study both fixed and variable frame durations. In addition to dynamically triggering WDM reconfiguration, our algorithms specify precisely how to route packets over the IP layer during the phases in which the WDM layer remains fixed. We demonstrate that optical-layer constraints do not affect the results, and provide an analysis of the specific case of WDM networks with multiple ports per node. In order to gauge the delay properties of our algorithms, we conduct a simulation study and demonstrate an important tradeoff between WDM reconfiguration and IP-layer routing. We find that multihop routing is extremely beneficial at low-throughput levels, while single-hop routing achieves improved delay at high-throughput levels. For a simple access network, we demonstrate through simulation the benefit of employing multihop IP-layer routes.**

***Index Terms*—Birkhoff–von Neumann switches, circuit switching, frame scheduling, Internet Protocol (IP), IP-over-WDM networks, matrix decomposition, multihop routing, network control, packet switching, queueing network, reconfiguration overhead, stochastic coupling, tunable transceivers, tuning latency, wavelength division multiplexing (WDM), WDM reconfiguration.**

I. Introduction

We consider an optical network architecture consisting of nodes having Internet Protocol (IP) routers overlaying optical cross connect (OXC), with the nodes interconnected by optical fiber, as in Fig. 1(a). This constitutes the physical topology of the network. Optical add/drop multiplexers (ADMs) and OXCs allow individual wavelength signals to be either dropped to the electronic routers at each node or to pass through the node optically. The logical topology consists of the lightpath interconnections between the IP routers and is determined by the configuration of the optical ADMs and transceivers at each node.

By enabling the transceivers[1] at the nodes to be tunable, the network allows for changes in the logical topology configuration. This capability is attractive, because it allows for dynamic reconfiguration algorithms to be employed in order to improve the throughput and delay properties of the network, as well as recover from network failures. In essence, a tradeoff emerges between lightpath reconfiguration at the wavelength division multiplexing (WDM) layer and routing at the electronic layer. Fig. 1 depicts the architecture of interest, for a particular five-node physical topology. Fig. 1(b) and (c) shows the cross-layer connections corresponding to two feasible logical topologies on the physical topology of Fig. 1(a).

The ability to reconfigure the logical topology requires tunable transceivers and OXCs. The effectiveness of an algorithm employing reconfiguration will depend on the speed with which reconfiguration takes place. In this paper, we do not require that the transceivers be fast tunable.

A. Performance Tradeoff Example

In an earlier study [1], the gains associated with dynamic topology reconfiguration under changing traffic were considered, resulting in algorithms for incremental reconfiguration to balance link loads. Consider a three-node line network, with a single transceiver per node. There are two possible ring logical configurations, as in Fig. 2.

If the traffic matrix T (corresponding to transmission requests) is given by

$$T = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

then by routing the traffic along $\mathcal{C}_1$, each logical link experiences a load of 2, while for $\mathcal{C}_2$, each logical-link load is 1. Clearly, the gain from reconfiguration in this scenario is a link-load reduction by a factor of 2.

In the stochastic setting, where traffic variations are characterized as random processes, and the system is subject to reconfiguration overhead, packet service delays are affected

Manuscript received December 15, 2004; revised April 29, 2005. The work of A. Brzezinski and E. Modiano was supported in part by the Defense Advanced Research Projects Agency under Grant MDA972-02-1-0021 and by the National Science Foundation (NSF) under Grants ANI-0073730 and ANI-0335217.

The authors are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: brzezin@mit.edu; modiano@mit.edu).

Digital Object Identifier 10.1109/JLT.2005.855691

[1] We use the words transceiver and port interchangeably in this paper. Thus, a single transceiver consists of an input port and an output port.

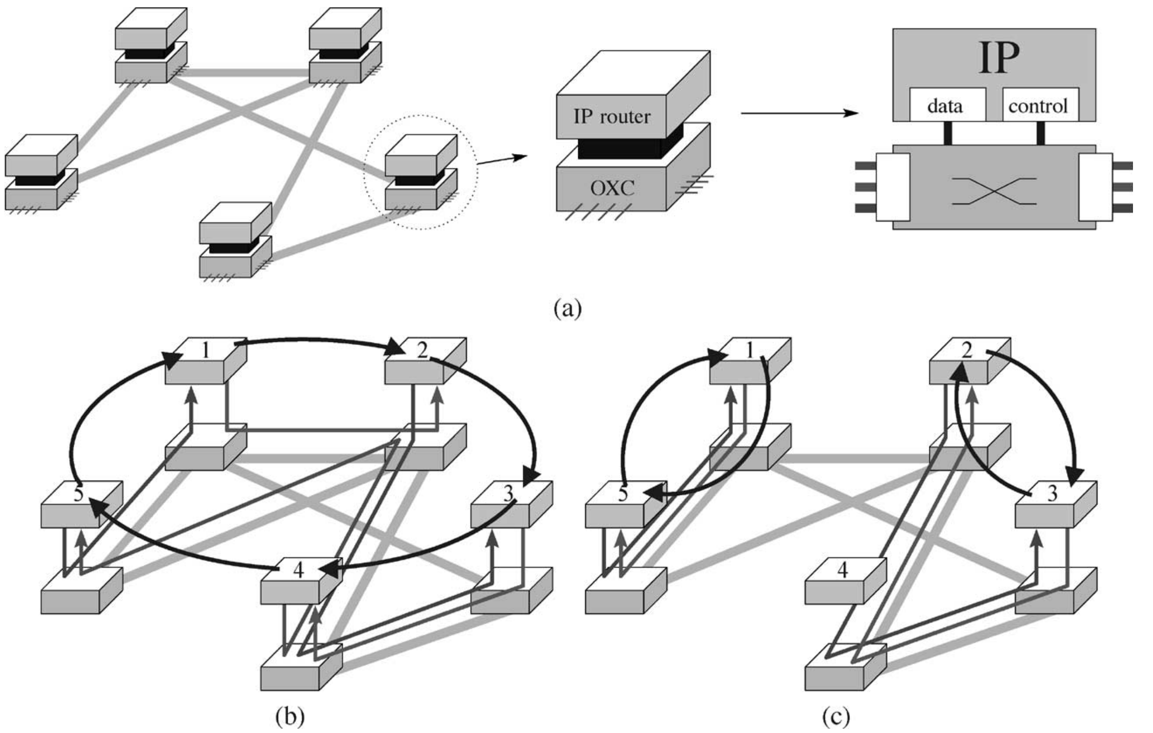

Fig. 1. Sample physical topology and feasible logical topologies for three wavelengths per fiber, one transceiver per node. (a) IP-over-WDM network architecture, with each node consisting of an optical crossconnect and an IP router. The network at the left is a five-node physical topology. (b) Ring logical topology $\{1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 1\}$. (c) Disconnected logical topology $\{1 \leftrightarrow 5, 2 \leftrightarrow 3\}$.

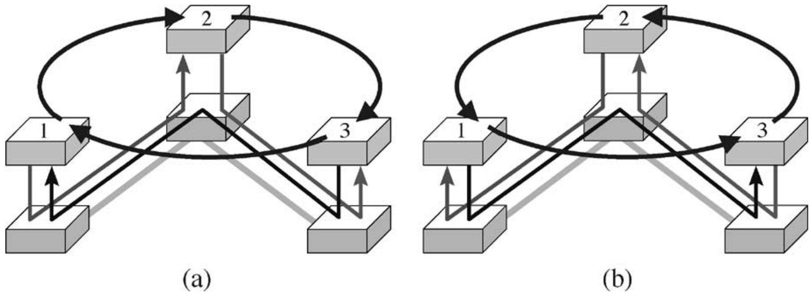

Fig. 2. Lightpath interconnections for three-node rings on a line physical topology. (a) $\mathcal{C}_1$: Ring $1 \rightarrow 2 \rightarrow 3$. (b) $\mathcal{C}_2$: Ring $1 \rightarrow 3 \rightarrow 2$.

by the joint algorithm for WDM topology reconfiguration and IP-layer packet routing. In this setting, the traffic configuration is characterized by an arrival rate matrix λ, where the entry on the ith row and jth column represents the long-term rate of exogenous arrivals of packets to node i destined for node j, in packets per time slot.

To demonstrate the important delay tradeoff between incurring reconfiguration overhead and additional load from IP-layer routing, consider arrival rate matrices λ_1 and λ_2 under the three-node network of Fig. 2

$$\lambda_1 = \begin{bmatrix} 0 & 0.2 & 0.5 \\ 0.5 & 0 & 0.2 \\ 0.2 & 0.5 & 0 \end{bmatrix}, \quad \lambda_2 = \begin{bmatrix} 0 & 0.4 & 0.5 \\ 0.5 & 0 & 0.4 \\ 0.4 & 0.5 & 0 \end{bmatrix}.$$

Under λ_1, if we fix the topology to be $\mathcal{C}_1$, each logical link has a long-term arrival rate of 1.2, which exceeds the maximum service rate of 1.0 for each link. Thus, under $\mathcal{C}_1$, the system becomes overloaded with unserviced traffic as time progresses. If $\mathcal{C}_2$ is employed, each logical link experiences a long-term rate of arrivals of 0.9, which is indeed sufficient to guarantee the stability[2] of the network.

It is not always possible to exclusively make use of a single logical topology configuration. Consider the arrival rate matrix λ_2. If we service traffic exclusively on $\mathcal{C}_1$, all links experience a long-term arrival rate of 1.4, while if $\mathcal{C}_2$ is exclusively chosen, the link arrival rates are each 1.3. In either case, the system becomes overloaded with unserviced traffic as time progresses. However, a time division multiplexing (TDM) schedule using only single-hop routes allocating at least 40% of its time to $\mathcal{C}_1$ and at least 50% of its time to $\mathcal{C}_2$ is sufficient to guarantee that the network is stable, so long as the contiguous service time allocated to each logical ring is adequately long to make the reconfiguration overhead negligible. Because the TDM schedule employs only single-hop routes, this ensures a long-term service rate of at least 0.4 packets per time slot

[2] We formally define the notion of network stability in Section III-A. It is sufficient here to say that the network is stable if the buffer backlogs at each node remain finite for all time.

to buffers associated with $\mathcal{C}_1$ [buffers for source–destination pairs (1,2),(2,3),(3,1)] and a long-term service rate of at least 0.5 packets per time slot to buffers associated with $\mathcal{C}_2$ [buffers for source–destination pairs (1,3),(2,1),(3,2)].

It is clear that in order to ensure stability and provide excellent delay properties under a broad class of traffic processes, it is essential to balance the idleness associated with reconfiguration against the additional load incurred from multi hopping along the IP layer.

B. Related Work

The reconfigurable network architecture has been approached in the literature from several angles. Many studies aim to achieve, in some sense, a balanced set of link loads [1]–[4]. The work of Labourdette and Acampora [2] considers a reconfigurable multihop WDM network subject to deterministic nonuniform traffic. The goal of this study is to determine an algorithm for joint reconfiguration and routing with desirable throughput properties. The authors suggest that minimizing the maximum link load (a minimax formulation) is an effective means of achieving strong throughput properties. A mixed integer program is provided for the joint optimization, and a heuristic separating the reconfiguration and routing problems and iterating between them is provided. In [1] and [3] branch-exchange algorithms are introduced to incrementally adjust the logical topology towards a desired configuration. Here, Labourdette *et al.* [3] approaches the problem essentially in a deterministic setting, by considering an initial WDM configuration as well as a fixed target configuration, and seeking a suitable sequence of two-branch exchanges[3] to transition between the two configurations with little overall disruption to the network. In [1], the problem is approached under dynamic traffic. This work recognizes that two-branch exchanges may leave the logical topology disconnected, which is undesirable under dynamic traffic, opting instead for three-branch exchanges, which are guaranteed to maintain connectivity. The work of Baldine and Rouskas [4] imposes at each time slot a cost for reconfiguring the logical topology and a reward that depends on the degree of load balancing for the current logical topology. An average-reward dynamic program is then formulated with the total reward at any time equal to a weighted sum of the cost and reward for that particular time.

The literature characterizing the ultimate throughput properties of optical networks subjected to dynamic/stochastic traffic is significantly sparser. The time-domain wavelength interleaved networking (TWIN) architecture of [5] and [6] looks at the network at the burst level, and reduces the optical transport network to essentially a crossbar switch with link delays. TWIN is a WDM-layer protocol only, relying on a fixed underlying tree-based logical topology configuration to execute single-hop end-to-end burst transmissions. TWIN is shown in [6] to enjoy asymptotically optimal throughput in optical networks with nonnegligible link transmission delays. The key technology for TWIN is ultra-fast tunable transceivers, and an assumption of negligible transceiver reconfiguration overhead.

C. Summary of Work

In one of our motivating studies [1], logical topology reconfiguration was initiated at regular intervals in order to deal with changing traffic. Furthermore, the reconfigurations were incremental, and made no guarantees about the stability of the system. In this paper, we provide the first systematic approach to the dynamic reconfiguration and routing problem under stochastic traffic in the presence of reconfiguration overhead. We determine stable algorithms employing IP-layer routing in order to elicit an understanding of the performance tradeoffs between reconfiguration at the optical layer and packet routing at the IP layer. The following are our major contributions.

1) We develop mechanisms for dynamically triggering WDM reconfiguration under stochastic traffic. Our algorithms are based on maximum weight scheduling decisions, and specify precisely when and how to reconfigure the WDM layer as well as the IP routing employed between reconfigurations.
2) We demonstrate the asymptotic throughput optimality of our frame-based algorithms in the presence of reconfiguration overhead.
3) For multiple transceivers per node, we demonstrate the stability region by providing a novel algorithm extending Birkhoff–von Neumann matrix decompositions to this setting.
4) Using delay as a performance metric, we employ simulations to demonstrate the important tradeoff between WDM reconfiguration and IP-layer routing. Our simulations point to the advantage of packet switching at low-throughput levels and circuit switching at high-throughput levels.

Additionally, we provide a preliminary analysis questioning the use of multihop routing for the case of negligible reconfiguration overhead. Furthermore, we analyze a class of algorithms that use random selection of logical rings as the underlying WDM topology, and demonstrate their throughput suboptimality. For an access network, we present simulation results demonstrating the tremendous advantage of IP-layer routing.

II. Reconfigurable Network Model

Consider an optical WDM network consisting of N nodes, labeled $1, 2, \ldots, N$, physically interconnected by optical fiber in an arbitrary topology. We assume that node i is equipped with P_i transceivers for $i = 1, \ldots, N$, and thus, at any time, may have at most P_i incoming and P_i outgoing logical links. For the most part (except where we explicitly say otherwise), we will restrict the values to $P_i = 1$ for all i. Under this distribution of ports, we assume that there exist a sufficient number of wavelengths to allow any arbitrary logical interconnection of nodes. Each node is equipped with $(N-1)$ virtual output queues (VOQs) in which data are held prior to transmission across the network, with $\mathrm{VOQ}_{i,j}$ containing the backlogged data at node i destined for node j. Time is assumed to be slotted,

[3] A two-branch exchange tears down two existing logical links $s_1 \rightarrow d_1$, $s_2 \rightarrow d_2$ and establishes the new logical links $s_1 \rightarrow d_2, s_2 \rightarrow d_1$.

and for simplicity of exposition, data units are in the form of fixed-length packets, each requiring a single slot for transmission. The network allows a maximum of one packet to be transmitted across any logical link during a slot. At any time, the network may initiate a logical topology reconfiguration, under which, existing lightpaths are torn down and new ones reestablished to form a new logical topology. Transceivers that are tuned are forced to be idle for the reconfiguration time of D slots, while links that are unaffected may continue to service traffic during reconfiguration.

The queue-occupancy process $\{X(n)\}_{n=0}^{\infty}$ is defined as an infinite sequence of matrices where $X(n)$ is the queue-backlog matrix at time n and $X_{i,j}(n)$ is the number of packets at node i destined for node j at time n. This process evolves according to the matrix equation

$$X(n+1) = X(n) - u(n+1) + a(n+1) \tag{1}$$

for $n \geq 0$. In (1), u is the control matrix and a is the arrival matrix. Note that $X(0)$ must be defined as some initial queue-backlog matrix. In our model, the queues are not restricted to have finite capacity. The process $\{a(n)\}_{n=1}^{\infty}$ corresponds to the exogenous arrivals to the system, with $a_{i,j}(n) = k$ if there are k arrivals to $\text{VOQ}_{i,j}$ at time n. We require that each arrival process $\{a_{i,j}(n)\}_{n=1}^{\infty}$ satisfies a strong law of large numbers (SLLN) [7]: Define the cumulative arrival process $\{A(n)\}_{n=1}^{\infty}$ according to $A_{i,j}(n) \triangleq \sum_{m=1}^{n} a_{i,j}(m)$. Then

$$\lim_{n\to\infty} \frac{A_{i,j(n)}}{n} = \lambda_{i,j} \quad \text{a.s.} \tag{2}$$

for $i, j = 1, 2, \ldots, N$. We do not allow self-traffic, which implies that $A_{i,i}(n) = 0$ for all i, n and thus, $\lambda_{i,i} = 0$ for all i. The long-term arrival rates are stored in matrix $\lambda = (\lambda_{i,j}, i, j = 1, \ldots, N)$.

The process $\{u(n)\}_{n=1}^{\infty}$ tracks the control decisions in the system, in particular, the IP-layer-routing choices over time. Thus, a positive entry $u_{i,j}(n) > 0$ implies that a packet was either departed or forwarded[4] from $\text{VOQ}_{i,j}$ under the control decision at time $n-1$ (i.e., node i departed a packet destined for node j along a lightpath originating at node i). A negative entry $u_{i,j}(n) < 0$ implies that a forwarded packet arrived to $\text{VOQ}_{i,j}$ at time n following the control decision at time $n-1$ (i.e., node i received a packet destined for node j along a lightpath terminating at node i). The restriction of a single transceiver per node implies, for every time n, that every row of $u(n)$ must add to no more than unity and every column to no less than -1. In other words, this means that no more than one packet may be forwarded/departed from any node at any time, and no more than one packet may be sent to a particular node. If we define the cumulative control process $\{U(n)\}_{n=1}^{\infty}$ according to $U_{i,j}(n) \triangleq \sum_{m=1}^{n} u_{i,j}(m)$, the network evolution (1) may be equivalently described by

$$X(n+1) = X(0) - U(n+1) + A(n+1). \tag{3}$$

TABLE I
SUMMARY OF KEY VARIABLES/SETS FROM THE NETWORK MODEL

$X_{i,j}(n)$	Cumulative number of packets in $\text{VOQ}_{i,j}$ at time $n \geq 0$
$A_{i,j}(n)$	Cumulative number of arrivals to $\text{VOQ}_{i,j}$ by time $n \geq 0$
$\lambda_{i,j}$	Long term arrival rate to $\text{VOQ}_{i,j}$
$U_{i,j}(n)$	Cumulative control at $\text{VOQ}_{i,j}$ by time $n \geq 0$ (includes departures from $\text{VOQ}_{i,j}$ and internal arrivals from forwarding)
$v(n)$	Integer matrix indicating logical topology enabled at time n
$\mathcal{V}$	Set of allowed logical topology matrices

Throughout this work and irrespective of the transceiver counts $P_i, i = 1, \ldots, N$, the $N \times N$ integer matrix $v(n)$ will denote the logical topology selected at time n: If $v_{i,j}(n) = l \geq 0$, then l single-wavelength links exist from source node i to destination node j. The diagonal entries of this matrix have no meaning under our model–they can take any value without having an effect on the logical topology implied by the off-diagonal entries. We denote by $\mathcal{V}$ the set of allowed logical topologies, subject to optical-layer connectivity constraints (such as wavelength limitations, multiple transceivers per node, and particular routing and wavelength assignment algorithms). When we restrict the network to have a single transceiver per node with no wavelength constraints, each feasible logical topology is represented by a permutation matrix, and $\mathcal{V}$ is the set of $N \times N$ permutation matrices.

When we allow multihop routes along the IP layer, our network model is a particular case of the constrained queueing model of [8]. There exist a total of $L \triangleq N^2 - N$ directed logical links from which any logical topology is chosen (since there are $N^2 - N$ distinct feasible source–destination pairs in the network). We index these links with $1, \ldots, L$. For link i, the origin node is defined by $q(i)$ and the destination node is defined by $h(i)$.

At each time $n \geq 1$, define the activation matrix $E(n) = (E_{i,j}(n), i = 1, \ldots, L, j = 1, \ldots, N)$ by setting $E_{i,j}(n) = 1$ if at time n, link i was activated to serve packets destined for node j, and $E_{i,j}(n) = 0$ otherwise. Denote $E_{:,j}(n)$ as the jth column of $E(n)$. We define $\mathcal{E}$ as the set of all allowed matrices E. For each destination node $j = 1, \ldots, N$, packet routing along the IP layer is implemented through the routing matrix $R^j = (R^j_{k,l}, k = 1, \ldots, N, l = 1, \ldots, L)$. Here, $R^j_{k,l} = 1$ if the destination node along link l is k and $k \neq j$, $R^j_{k,l} = -1$ if the source node for link l is k, and $R^j_{k,l} = 0$ otherwise. Given this notation, the network evolution (1) becomes

$$X_{:,j}(n+1) = X_{:,j}(n) + R^j E_{:,j}(n) + a(n+1) \tag{4}$$

for $j = 1, \ldots, N$, where $X_{:,j}$ is the jth column of matrix X. Note that $u_{:,j}(n+1) = -R^j E_{:,j}(n)$ for $j = 1, \ldots, N$ and $n \geq 0$.

For convenience, we have summarized the key variables of this section in Table I.

A. Scheduling Under Tuning Latency, Propagation Delay, and Distributed Control

Since we are operating in a distributed mesh-network environment, it may not be practical to assume that each node is synchronized to a common clock. A key aspect of the

[4] A packet is forwarded when it is sent to an intermediate node along the IP layer.

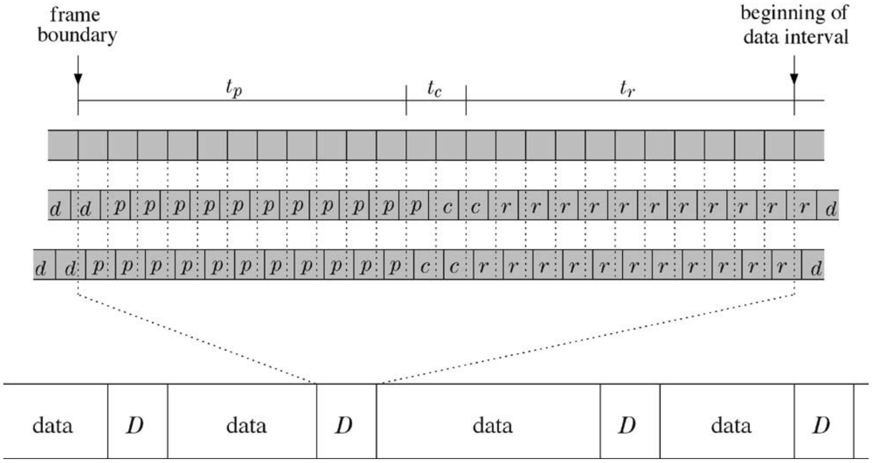
frame boundary
beginning of data interval
t_p
t_c
t_r
d d p p p p p p p p p p p c c r r r r r r r r r r r d
d d p p p p p p p p p p p c c r r r r r r r r r r r d
data
D
data
D
data
D
data
D

Fig. 3. To change the logical topology, a reconfiguration interval is used. The interval consists of t_p slots for propagation delay of the final packets of the last data interval (slots labeled p), t_c slots for passing control information in order to decide on a new logical topology (slots labeled c), and t_r slots to tune the transceivers and establish the new logical topology (slots labeled r). Slots labeled d are slots for packet transmission (corresponding to a data interval). The top sequence of slots corresponds to a common time reference according to which frame boundaries are set. The second and third sequences of slots correspond to distinct nodes in the network. As illustrated, these slots need not be synchronized to each other or to the common time reference. The frame-based scheduling is depicted at the bottom, with D used to indicate the reconfiguration interval of duration D, and data used to indicate the data interval.

reconfiguration and routing algorithms of this paper is that they employ frame-based scheduling, where logical links are held fixed over data intervals, and the logical topology is changed over reconfiguration intervals. A frame boundary occurs at the instant when the network initiates the sequence of controls to reconfigure the logical topology. This sequence includes: 1) the time for the final packets of the terminated frame to arrive at their respective destinations t_p (can be taken as a fixed value if we bound the delay over all possible logical links); 2) the time for information exchange in order to make a decision about the new logical topology to configure t_c (this information exchange may have occurred prior to the frame boundary, in which case $t_c = 0$); and 3) the time for tuning the transceivers to establish a new logical topology t_r. The value of t_p depends on the underlying fiber plant topology of the network, which in the case of wide area networks (WANs) is in the order of tens of milliseconds. The value of t_r depends on the transceiver technology, with current components requiring it to be in the order of tens of milliseconds for reconfiguration. Thus, we designate the reconfiguration overhead $D = t_p + t_c + t_r$.

Using tools from standard clock-synchronization algorithms [9], each node can be made aware of a common time reference. Rather than requiring that the electronics at each node be synchronized to this common reference, the reference is used to make nodes aware of frame boundaries. In the case of variable frame durations, this reference can be used to establish agreement between the nodes about each successive frame boundary. The frame boundary is initialized by having each node stop transmission of packets after the complete transmission of any packet being serviced at that time. We have illustrated the structure of a reconfiguration interval in Fig. 3.

III. Algorithms for Asymptotic Throughput Optimality

We begin our consideration of the control problem by demonstrating that the system is stable under a broad class of arrival processes. We first introduce two well-known algorithms, which when adapted to our model, jointly perform WDM reconfiguration and IP-layer routing. These algorithms are based on maximum weighted matchings (MWMs) and are known to stabilize the system for the special case of zero reconfiguration overhead $(D = 0)$. Since these algorithms have not been previously considered in the context of IP-over-WDM networks, our descriptions are somewhat extensive in order to make perfectly clear how they jointly perform IP-layer routing and WDM reconfiguration.

For $D > 0$, we prove that any stable algorithm for the case of $D = 0$ may be transformed into a frame-based algorithm that stabilizes the network. Furthermore, we introduce a bias-based algorithm that makes reconfiguration decisions by taking into account the current logical topology of the network. These algorithms are a natural extension of maximum weight scheduling algorithms to the case $D > 0$.

A. Preliminaries

Definition 3.1: Matrix $V = (v_{i,j}, i, j = 1, \ldots, N)$ is doubly substochastic if

$$\sum_i v_{i,j} \leq 1 \qquad \forall j, \quad \sum_j v_{i,j} \leq 1 \qquad \forall i. \tag{5}$$

If the inequalities in (5) are all strict inequalities, then V is called strictly doubly substochastic [10].

Definition 3.2: The system is stable if the backlog process $\{X(n)\}_{n=0}^{\infty}$ satisfies [11]

$$\limsup_{n\to\infty} E\left[\sum_{i,j} X_{i,j}(n)\right] < \infty.$$

In essence, every queue-backlog process must have finite expectation in the long run.

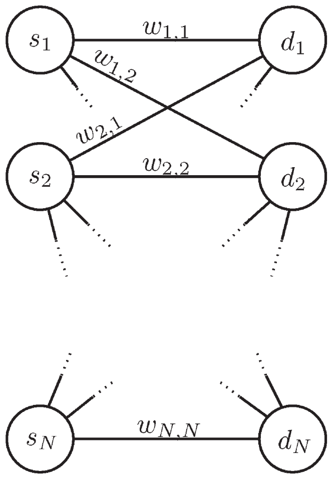

Fig. 4. Weighted complete bipartite graph for maximum weight scheduling.

B. Single-Hop Algorithm Using MWMs

We begin by introducing an important single-hop algorithm that is known to be stable for the case of $D = 0$. In switching theory, perhaps the most commonly studied algorithm is the MWM algorithm (described below). Essentially, MWM constructs a complete weighted bipartite graph, as in Fig. 4, where the left N nodes correspond to source nodes, and the right N nodes correspond to destination nodes. At time slot $n \geq 0$, MWM sets $w_{i,j} = X_{i,j}(n)$ for all i, j. The logical topology at time n is selected by determining a maximum weighted matching on this graph, with the edges of the matching established as logical links over the WDM physical topology. Under MWM, electronic-layer routing is restricted to single-hop paths, which means that for each logical link i, only VOQ$_{q(i),h(i)}$ may be serviced by departing packets along that link.[5]

Maximum Weighted Matching Algorithm (MWM)
At time slot $n \geq 0$, matrix $v(n) = (v_{i,j}(n), i,j = 1,\ldots,N)$ is chosen to maximize

$$\langle v(n), X(n)\rangle \triangleq \sum_{i,j} v_{i,j}(n) X_{i,j}(n)$$

subject to the constraints

$$\sum_j v_{i,j}(n) \leq 1 \qquad \forall i \tag{6}$$

$$\sum_i v_{i,j}(n) \leq 1 \qquad \forall j \tag{7}$$

$$v_{i,j}(n) \in \{0,1\} \qquad \forall i,j. \tag{8}$$

$v(n)$ corresponds to the logical topology selected at time n. The control $u(n+1)$ is then given by

$$u_{i,j}(n+1) = \begin{cases} v_{i,j}(n), & \text{if } X_{i,j}(n) > 0 \\ 0, & \text{if } X_{i,j}(n) = 0. \end{cases} \tag{9}$$

The power of MWM to stabilize the $N \times N$ crossbar switch is particularly well demonstrated in [12], with the following important stability result, adapted to our reconfigurable queueing-network model.

Theorem 3.1: For $D = 0$, and any arrival processes satisfying an SLLN with a strictly doubly substochastic arrival rate matrix λ, the network is stable under MWM.

Proof: This follows immediately from the proof of [12, Lemma 5]. ■

Since the set of doubly substochastic arrival rate matrices is the closure of all stabilizable arrival rate matrices, MWM is called throughput optimal for the network when $D = 0$.

C. Multihop Algorithm Using "Differential Backlogs"

Again considering the case $D = 0$, a powerful algorithm taking advantage of IP-layer routing and again making use of maximum weighted matchings was shown to be throughput optimal in [8]. We refer to this algorithm as DB (described below).

Differential Backlog Algorithm (DB)
At time slot $n \geq 0$,

1) For each link i and destination node j, calculate the quantity $d_{i,j}(n)$ according to

$$d_{i,j}(n) = \begin{cases} X_{q(i),j}(n) - X_{h(i),j}(n), & \text{if } h(i) \neq j \\ X_{q(i),j}(n), & \text{otherwise.} \end{cases} \tag{10}$$

Define matrix $Z(n) = (Z_{i,j}(n), i,j = 1,\ldots,N)$, with $Z_{q(i),h(i)}(n) \triangleq \max_j\{d_{i,j}(n)\}$ for $i = 1,\ldots,L$.

2) Select matrix $v(n)$ to maximize $\langle v(n), Z(n)\rangle$, subject to constraints (6)–(8). Define the maximum weighted activation vector $\tilde{c} = (\tilde{c}_i, i = 1,\ldots,L)$ according to $\tilde{c}_i \triangleq v_{q(i),h(i)}(n)$ for $i = 1,\ldots,L$.

3) For each edge i, let $\hat{j}_i$ be a destination node satisfying $d_{i,\hat{j}_i}(n) = \max_j\{d_{i,j}(n)\}$. The matrix $E(n)$ is populated according to

$$E_{i,j}(n) = \begin{cases} 1, & \text{if } \tilde{c}_i(n) = 1, j = \hat{j}_i, X_{q(i),j}(n) > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

If we refer to each packet destined for a particular destination as a unit of a commodity that is specific to that destination, then the differential backlog at each link corresponding to a particular commodity is given by the difference of the backlog of that commodity at the source node of that link and the backlog of that commodity at the destination node of that link. Thus, referring to (10), $d_{i,j}$ is the differential backlog of commodity j on link i.

In words, for each time $n \geq 0$, DB may be described as follows. Step 1 considers in turn each possible logical link i, and calculates for that logical link the maximum differential backlog over all commodities. This value is placed in matrix $Z(n)$ at entry $(q(i), h(i))$. Next, the bipartite graph of Fig. 4 is enlisted in step 2, by setting $w_{i,j} = Z_{i,j}(n)$ for all i, j, and selecting a maximum weighted matching. Again, the edges of

[5] Recall from Section II that for directed link i, the origin node is denoted by $q(i)$ and the destination node is denoted by $h(i)$.

the matching are the logical links enabled at time n (topology reconfiguration), while the actual VOQ to service on each enabled link is given by the commodity that maximizes the DB for that link (electronic-layer routing). This process is summarized in the selection of matrix E in step 3.

Thus, it is clear that DB is inherently a joint algorithm for WDM-layer reconfiguration and IP-layer routing. We adapt the optimality result of [8] to our network model and summarize the result in Theorem 3.2.

Theorem 3.2: Consider any joint arrival process $\{A_{i,j}(n)\}_{n=1}^{\infty}, i,j = 1,\ldots,N$ given by independent identically distributed (i.i.d.) sequences of random variables, independent among themselves, with finite second moments, and a strictly doubly substochastic arrival rate matrix λ. Then, for $D = 0$, the reconfigurable queueing network is stable under DB.

Proof: This follows immediately from [8, Lemma 3.2 and Th. 3.2]. ■

D. Frame-Based Algorithms for $D > 0$

Given the above stabilizing algorithms (MWM and DB) for the case $D = 0$, it is intuitively clear that they may be adapted to the case of $D > 0$ using frame-based schemes, where reconfiguration decisions are only made at frame boundaries. In this section, we formalize this idea by providing a general result showing how any stabilizing scheme for $D = 0$ may be transformed into a stabilizing scheme for the case of any $D > 0$.

Frame-Stabilizing Algorithm for Algorithm P (F-P)
Given: an integer $F \geq 0$.
For each $k = 0, 1, \ldots,$

1) At time kF, make a reconfiguration decision according to the decision rule of algorithm P under the backlog matrix $X(kF)$.
2) Set $u_{i,j}(l) = 0$ for $l = kF, \ldots, kF + D - 1$ and all i, j, to allow for reconfiguration overhead.
3) Set $u(l) = u^{\mathrm{P}}(X(kF))$ for $l = kF + D, \ldots, (k+1)F - 1$. Here $u^{\mathrm{P}}(X)$ is the IP-layer-routing decision of algorithm P given backlog matrix X.
4) For each VOQ, batch exogenous arrivals over the frame, with the number of batched arrivals for $\mathrm{VOQ}_{i,j}$ at time $(k+1)F$ denoted by $B_{i,j}((k+1)F)$. At time $(k+1)F$, prior to the reconfiguration decision but after the arrival of new packets, remove the oldest

$$(F - D)\left\lfloor \frac{B_{i,j}((k+1)F)}{(F - D)} \right\rfloor$$

packets from the batch and place them in $\mathrm{VOQ}_{i,j}$. The leftover packets remain in the batch for the next frame.

For algorithm P and frame size F, the frame version of P is denoted by F-P, and is described above. The algorithm alternates regularly between idle and service intervals, as illustrated in Fig. 5. The algorithm operates as follows: at each frame boundary, under backlog matrix X, F-P makes the same WDM-reconfiguration decision that P makes under backlog X. Given this WDM logical topology choice, algorithm P has a control matrix (corresponding to electronic-layer routing) $u^{\mathrm{P}}(X)$. Algorithm F-P idles for D slots to allow for reconfiguration overhead, and then applies the control $u^{\mathrm{P}}(X)$ over the remaining slots in the frame. The arrival process is batched in order to ensure that control $u^{\mathrm{P}}(X)$ can be applied over the duration of the frame without running out of backlogs to service.

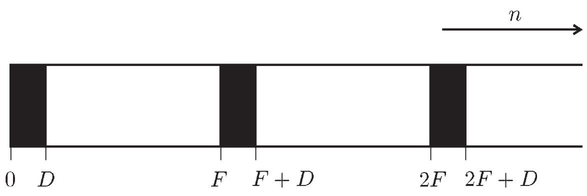

Fig. 5. Regular ON–OFF nature of the frame-based algorithm.

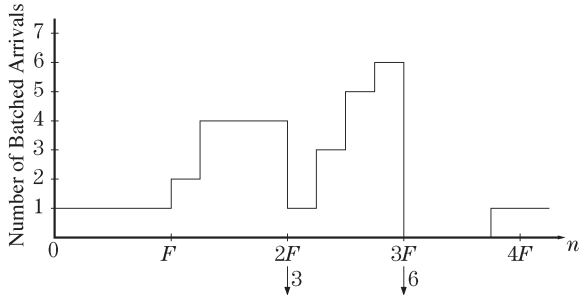

Fig. 6. Illustration of batch-size process for a particular VOQ.

As an example, suppose that $D = 1$ and $F = 4$. Fig. 6 shows how exogenous arrivals for a particular VOQ are batched before being released to that VOQ for service. All exogenous arrivals are batched and are not available for service until the frame boundary, when the maximum number of batched packets that are a multiple of $F - D = 3$ are released to the VOQ (here, we have three packets released for service at time $2F$ and six packets released at time $3F$). Thus, the batch-size process is nondecreasing over the frame interval, and decreases by a multiple of 3 at the frame boundaries. Because only three slots are allocated to servicing VOQs within each frame, this ensures that each VOQ backlog changes by an integer multiple of three packets over every frame. Thus, the frame scheme looks at the system only at the frame boundaries and considers the VOQ backlog processes divided by $F - D = 3$, and ties the resulting process back to the stabilizing scheme for $D = 0$.

Theorem 3.3: Suppose algorithm P stabilizes the network for $D = 0$ for some set of arrival processes $\mathcal{A}$. Then for each $D > 0$, if there exists F such that the cumulative arrival process $\{A(n)\}_{n=1}^{\infty}$ satisfies $\{\tilde{A}(n)\}_{n=1}^{\infty} \in \mathcal{A}$, where

$$\tilde{A}(n) = \left\lfloor \frac{A(nF)}{F - D} \right\rfloor$$

and then P is frame stabilizable. Specifically, algorithm F-P stabilizes the network.

Proof: The number of batched arrivals released to the system for service at each frame boundary, kF for $k = 1, 2, \ldots$, is given by $(F - D)(\tilde{A}(k) - \tilde{A}(k-1))$, which is clearly an integer multiple of $(F - D)$. Thus, since F-P services queues in batches of $(F-D)$ slots per frame, with the same control decision held over the duration of the frame, we are guaranteed that every queue backlog is an integer multiple of $(F-D)$ packets under F-P.

Define the process $\{\tilde{X}(n)\}_{n=0}^{\infty}$ with $\tilde{X}(n)$ equal to $1/(F-D)$ times the queue backlog at the beginning of slot nF under F-P. The evolution of $\{\tilde{X}(n)\}_{n=0}^{\infty}$ is defined according to the arrival process $\{\tilde{A}(n)\}_{n=1}^{\infty}$ (which we assume to be a member of the set $\mathcal{A}$), and scheduling decisions according to algorithm P at each n. Thus, the process $\{\tilde{X}(n)\}_{n=0}^{\infty}$ is equivalent to the backlog process under P for $D=0$ and exogenous arrival process $\{\tilde{A}(n)\}_{n=1}^{\infty}$. This implies the stability of $\{\tilde{X}(n)\}_{n=0}^{\infty}$ and consequently the stability of the queue-backlog process under F-P. ■

Given Theorems 3.1–3.3, we may immediately infer the existence of frame-based stable scheduling policies for any $D>0$. Define the value δ by

$$\delta = 1 - \max\left\{\max_i \sum_j \lambda_{i,j}, \max_j \sum_i \lambda_{i,j}\right\}.$$

Corollary 3.1: The frame-based version of MWM, which we refer to as F-MWM, is stable under any arrival process satisfying an SLLN with $\delta > 0$, if $F > D/\delta$.

Proof: Theorem 3.1 holds under any process satisfying $\delta < 1$. Thus, if we choose any process $\{A(n)\}_{n=1}^{\infty}$ with $\delta < 1$, then the process $\{\tilde{A}(n)\}_{n=1}^{\infty}$ must satisfy

$$\begin{aligned}\lim_{n\to\infty} \frac{\tilde{A}(n)}{n} &= \lim_{n\to\infty} \frac{1}{n}\left\lfloor \frac{A(nF)}{F-D} \right\rfloor \\ &= \frac{F}{F-D} \lim_{n\to\infty} \frac{A(nF)}{nF} \\ &= \frac{F}{F-D}\lambda\end{aligned}$$

where λ is the arrival rate matrix. For $\tilde{A}(n)$ to be stable under MWM, the matrix $F/(F-D)\lambda$ must be strictly doubly substochastic, which implies $F > D/\delta$. ■

Corollary 3.2: The frame-based version of DB, which we refer to as F-DB, is stable under any i.i.d. arrival processes that are mutually independent, with finite second moments, if $F > D/\delta$.

Proof: Similar to that of Corollary 3.1. ■

Since Corollaries 3.1 and 3.2 apply to any strictly doubly substochastic arrival rate matrix, but require a frame size F that depends on the value $\delta > 0$, we call the frame-based policies asymptotically throughput optimal.

It is intuitively clear that the extensions of F-MWM and F-DB (the frame versions of MWM and DB, respectively) that continue service during reconfiguration intervals, in which the underlying logical topology does not change, are stable. Furthermore, it is not necessary to go through the additional complications of tracking batched arrivals; instead, arrivals may be immediately placed in their VOQs ready for service. Stability also follows for the extension of F-DB, which instead of employing the same control decision through the frame interval, services the maximum weighted control subject to the fixed underlying logical topology. For these extensions of the frame-based algorithms, the proof of stability follows by the fact that the Lyapunov drift [13] under either F-MWM or F-DB is greater than under the corresponding refined algorithm.

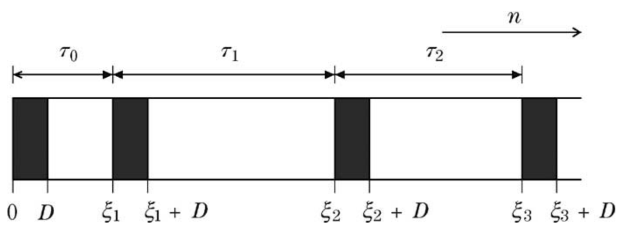

Fig. 7. Service intervals of the AB algorithm.

E. *Additive Bias-Based Algorithm*

In this section, we introduce Additive Bias-Based Algorithm (AB), based on MWM, which provides asymptotic throughput optimality for any $D > 0$. Here, we assume that the dissemination of control information across the network is sufficiently fast such that every node is aware of the backlog matrix at each slot. Thus, this algorithm is also well suited for scheduling crossbar switches with reconfiguration overhead.

Additive Bias-Based Algorithm (AB)
Given: an integer $b \geq 0$.
At time $n \geq 0$, if the system is not performing reconfiguration, then the matrix (logical topology) $v(n)$ is chosen to maximize

$$\langle v(n), X(n)\rangle_+ \triangleq b1_{\{v(n)=v(n-1)\}} + \sum_{i,j} v_{i,j}(n)X_{i,j}(n) \quad (12)$$

subject to the constraints (6)–(8). If $v(n)$ is different from $v(n-1)$, then the network idles for D slots while reconfiguration occurs.

AB is given above. The intuition behind the algorithm is that every decision to reconfigure should be followed by some opportunity to service packets under the logical topology selected (in essence, the algorithm has a built-in hysteresis). Under AB, WDM-reconfiguration decisions are made at each time slot, using maximum weighted matchings as in algorithm MWM. The only difference is that the weight associated with the existing logical topology prior to the decision instant is biased additively by the constant number b. This bias is chosen in such a way as to increase the expected time interval between WDM-reconfiguration decisions sufficiently to ensure stability of the system for $D > 0$.

Fig. 7 illustrates the intervals associated with service and reconfiguration phases of AB. As opposed to the frame-based scheduling policies, the service intervals are of variable duration. We denote by ξ_n the nth reconfiguration decision instant, with $\xi_0 \triangleq 0$, and $\tau_n \triangleq \xi_{n+1} - \xi_n$.

We now formulate a necessary condition for the stability of the bias-based algorithm. The result is based on the fluid-limits technique (see e.g., [7]). We begin by characterizing the dynamics for the system. For $v \in \mathcal{V}$, let $Q_v(n)$ be the cumulative time spent servicing logical topology v up to time

n, and $Q_R(n)$ the cumulative time spent idle reconfiguring the system up to time n. The system dynamics are then given by

$$X_{i,j}(n) = A_{i,j}(n) - U_{i,j}(n) \tag{13}$$

$$U_{i,j}(n) = \sum_{v\in\mathcal{V}}\sum_{l=1}^{n} v_{i,j}1_{\{X_{i,j}(l)>0\}}\left(Q_v(l) - Q_v(l-1)\right) \tag{14}$$

$$Q_v(\cdot) \text{ is nondecreasing} \tag{15}$$

$$Q_R(n) + \sum_{v\in\mathcal{V}} Q_v(n) = n. \tag{16}$$

In (13), we modify the definition of the arrival variable $A_{i,j}(n)$ so that $A_{i,j}(0)$ is the initial backlog matrix at time 0 [i.e., $A_{i,j}(0) = X_{i,j}(0)$]. We allow the above system dynamics to hold over the domain of positive real numbers $\mathbb{R}_+$ by letting $X_{i,j}(t) = X_{i,j}(\lfloor t\rfloor), \forall t \geq 0$, and similarly for A. For variables U, Q_v, and Q_R, we retain continuity in continuous time by linearly interpolating between values of the variables at the nearest integer time slots: for example, for $t \in (n, n+1)$, $U_{i,j}(t) = U_{i,j}(n) + (t-n)(U_{i,j}(n+1) - U_{i,j}(n))$.

Since the above queue dynamics depend on the queue occupancy at time 0, we may introduce a sequence of systems identical to above, indexed by integer $r \geq 0$, where r equals the initial summed backlog over all queues in the system at time 0. For each $r \geq 0$, the system dynamics are as above, with the variables denoted by $X_{i,j}^{(r)}$, $A_{i,j}^{(r)}$, $D_{i,j}^{(r)}$, $Q_v^{(r)}$, and $Q_R^{(r)}$. For any $t \geq 0$, denote the scaled variable $x_{i,j}^{(r)}(t) = X_{i,j}^{(r)}(rt)/r$, and similarly for the scaled variables $d_{i,j}^{(r)}(t)$, $a_{i,j}^{(r)}(t)$, $q_v^{(r)}(t)$, and $q_R^{(r)}(t)$. It can be shown (similar to [14]) that the sequences of scaled variables (indexed by r) converge to the fluid limits $\bar{x}_{i,j}(t)$, $\bar{a}_{i,j}(t)$, $\bar{u}_{i,j}(t)$, $\bar{q}_v(t)$, and $\bar{q}_R(t)$, almost surely. These fluid-limit processes satisfy the following fluid equations, for $t \in \mathbb{R}_+$:

$$\bar{x}_{i,j}(t) = \bar{a}_{i,j}(t) - \bar{u}_{i,j}(t) \tag{17}$$

$$\bar{a}_{i,j}(t) - \bar{a}_{i,j}(0) = \lambda_{i,j}t \tag{18}$$

$$\bar{u}_{i,j}(0) = 0 \tag{19}$$

$$\bar{q}_R(t) + \sum_{v\in\mathcal{V}} \bar{q}_v(t) = t \tag{20}$$

$$\dot{\bar{u}}_{i,j}(t) = \sum_{v\in\mathcal{V}} v_{i,j}\dot{\bar{q}}_v(t), \quad \text{if } \bar{x}_{i,j}(t) > 0. \tag{21}$$

For the following results, we redefine $\delta > 0$ as a positive number satisfying

$$\delta < 1 - \max\left\{\max_i \sum_j \lambda_{i,j}, \max_j \sum_i \lambda_{i,j}\right\}. \tag{22}$$

Lemma 3.1: If the fluid-limit process $\bar{q}_R(t)$ satisfies $\dot{\bar{q}}_R(t) \leq \delta$ for all $t \geq 0$, then AB stabilizes the network.

Proof: See Appendix A. ■

Note that for $D = 0$, Lemma 3.1 immediately implies that AB is stable, since zero time is lost to reconfiguration and thus, $\bar{q}_R(t) = 0$ for all t. For $D > 0$ we now use Lemma 3.1 to prove the stability of the network under any joint Bernoulli-arrival process.

Theorem 3.4: Under Bernoulli arrivals (not necessarily independent or identically distributed in time or across VOQs) with $\delta > 0$, if b is chosen to satisfy $b/N > 2D/\delta - D$, then AB stabilizes the reconfigurable queueing network.

Proof: Recall that $v(\xi_n)$ is the maximum weighted logical topology at time ξ_n. We will characterize the minimum time needed for another logical topology $v' \neq v(\xi_n)$ to become the maximum weighted logical topology and thus, trigger a WDM reconfiguration. At time ξ_n, v' satisfies

$$\langle v', X(\xi_n)\rangle \leq \langle v(\xi_n), X(\xi_n)\rangle. \tag{23}$$

After time ξ_n, logical topology $v(\xi_n)$ will be effectively biased with b additional dummy packets over v'. Since the arrival process is Bernoulli, no more than a single packet may arrive to any VOQ at each time slot. Suppose that a single packet arrives to each of the VOQs corresponding to logical topology v' at every slot, and v' does not have any lightpaths in common with $v(\xi_n)$. Further suppose that there are no arrivals to VOQs corresponding to $v(\xi_n)$, and that at each slot, at most one packet is removed from each of the VOQs corresponding to $v(\xi_n)$. Then, in order to have a decision to reconfigure the logical topology, the inter-reconfiguration interval τ_n must satisfy

$$\langle v', X(\xi_n)\rangle + \tau_n N > b + \langle v(\xi_n), X(\xi_n)\rangle - (\tau_n - D)N. \tag{24}$$

Combining (23) and (24), we obtain

$$\tau_n > \frac{b}{2N} + \frac{D}{2}. \tag{25}$$

Suppose $b/N \geq 2D/\delta - D$. Then, using (25), we have that $\tau_n > D/\delta$ for all n, which means that irrespective of the backlog process, at least D/δ slots pass before a reconfiguration decision. Thus, for $\varepsilon > 0$

$$Q_R^{(r)}\left(r(t+\varepsilon)\right) - Q_R^{(r)}(rt) < D\left\lceil \frac{r\varepsilon}{\frac{D}{\delta}}\right\rceil \tag{26}$$

$$\leq r\delta\varepsilon + D. \tag{27}$$

Dividing both sides of (27) by r, the right-hand side of the inequality can be made arbitrarily close to $\delta\varepsilon$ for a sufficiently large integer r. This immediately implies that $\dot{\bar{q}}_R(t) < \delta$. ■

F. Imposing Additional Optical-Layer Constraints

Though we have cast the theorems of this paper in the context of networks with a single port per node and no wavelength constraints, the theorems are valid more generally. In fact, the theorems hold true if the set of allowed logical topologies $\mathcal{V}$ in a network is given. Thus, our frame- and bias-based schemes may be easily generalized to more complex network scenarios, such as networks with multiple ports per node, and with wavelength constraints and associated routing- and wavelength-assignment algorithms, to guarantee asymptotic throughput optimality. In general, so long as there exists a convex combination of allowed logical topologies $v \in \mathcal{V}$ whose entries all strictly exceed those of the arrival rate matrix λ, then frame- and bias-based schemes may be constructed to stabilize the network. For additional details on stability issues, consult [8].

To demonstrate how particular optical networking constraints affect the set of stabilizable arrival rates, we consider the general scenario where node i has P_i ports for $i = 1, \ldots, N$. We again assume sufficiently many wavelengths such that the port constraint is the only active constraint affecting the system.

Theorem 3.5: For a WDM network with port distribution $\{P_i\}_{i=1}^N$, any arrival rate matrix λ satisfying

$$\sum_i \lambda_{i,j} \le P_j \qquad \forall j, \quad \sum_j \lambda_{i,j} \le P_i \qquad \forall i \tag{28}$$

may be expressed as a convex combination of valid logical topology matrices.

Proof: See Appendix B. A different proof of this result may be found in [15]. However, our proof is a novel natural extension of the well-known Birkhoff–von Neumann decomposition for substochastic matrices (see e.g., [16]). ■

Given Theorem 3.5, it may be shown that any arrival rate matrix satisfying (28) with strict inequalities is stable when $D = 0$. Similarly, the stability of the frame- and bias-based algorithms must then follow for appropriately chosen frame/bias sizes. In particular, it can be shown that the proof of Theorem 3.3 remains valid under the general port constraint so long as

$$F > D \max\left\{\max_i \frac{P_i}{P_i - \sum_j \lambda_{i,j}}, \max_j \frac{P_j}{P_j - \sum_i \lambda_{i,j}}\right\}.$$

For the bias-based algorithm, if we redefine δ as any positive number satisfying

$$\delta < \max\left\{\max_i \frac{P_i - \sum_j \lambda_{i,j}}{P_i}, \max_j \frac{P_j - \sum_i \lambda_{i,j}}{P_j}\right\}$$

then the proof of Lemma 3.1 can be shown to follow. Consequently, Theorem 3.4 can be extended to state that the bias-based algorithm is stable so long as

$$\frac{b}{N} \ge 2D \max_i \frac{P_i}{\delta} - D \max_i P_i.$$

IV. Algorithm Performance

In this section, we compare the performance of algorithms under different traffic conditions, reconfiguration overheads, and physical topologies. Our simulations demonstrate that there exists a tremendous advantage to employing multihop routing at the IP layer under certain conditions. In particular, when there is a single transceiver per node, multihop routing is advantageous at low-throughput levels. Also, we observe the tremendous advantage of employing mutlihop routing in an access-network scenario, where a single hub node has N transcievers and each of the other local nodes is equipped with a single transceiver.

When considering the system at the packet level, a relevant performance metric is the average service delay experienced by packets in the system. Through a straightforward application of Little's formula, the average service delay is tied to the time average aggregate queue backlog. For initial queue-occupancy matrix $X(0) = \hat{X}$, under algorithm π and arrival rate matrix λ, the time average delay is given by

$$\frac{1}{\sum_{i,j} \lambda_{i,j}} \limsup_{N \to \infty} \frac{1}{T} E_{\hat{X}} \left[\sum_{n=0}^{T-1} \sum_{i,j} X_{i,j}^{\pi}(n)\right]$$

where $X^{\pi}(n)$ is the queue-backlog matrix at time n under algorithm π. It turns out that quantifying the average delay is difficult, because of the widely varying collection of allowable traffics that have the same arrival rates. Using the theory of Lyapunov functions, the authors of [11] derive bounds on average queue occupancy (and consequently on average delay), which achieve varying degrees of tightness, depending on how correlated different arrival streams are. For this reason, this section makes use of both theory and extensive simulation results to arrive at our conclusions.

In gigabit networks, reconfiguration overheads in the order of $D = 1000$ to $D = 50\,000$ time slots are reasonable values. We only provide data for the case $D = 1000$, though our tests for larger D values yield identical conclusions.

A. *Zero Reconfiguration Overhead* $(D = 0)$

For $D = 0$ it is unknown whether in fact there exists any benefit to IP-layer routing. We begin by showing that for $N = 3$, each algorithm employing packet forwarding is no better than an associated algorithm that never forwards packets.

Theorem 4.1: For $N = 3$, any algorithm employing packet forwarding has an associated algorithm that does not forward packets with an equal or lower average aggregate backlog when $D = 0$, for any joint arrival distribution.

Proof: See Appendix C. ■

Essentially, we may conclude definitively that for $N = 3$, when there is no reconfiguration overhead, there is no benefit from treating the system as more than a switch. For $N > 3$, it is not possible to generalize Theorem 4.1 directly to conclude that packet forwarding is not beneficial with respect to average delay. We leave this as an interesting open problem for future study.

B. *Overview of Algorithms Tested*

We compare several algorithms for joint WDM topology reconfiguration and IP-layer routing. The algorithms are frame- or bias-based versions of the following:

1) MWM;
2) DB;
3) Prioritized DB—DB for reconfiguration and routing decisions, with priority given to single-hop packets;
4) MWM Minhop—MWM for logical topology decisions, with minhop routing at the IP layer.

The algorithms Prioritized DB and MWM Minhop have not been introduced until now. They are heuristic algorithms that we devised in order to test the delay properties of MWM and DB. Prioritized DB operates on the philosophy that once DB

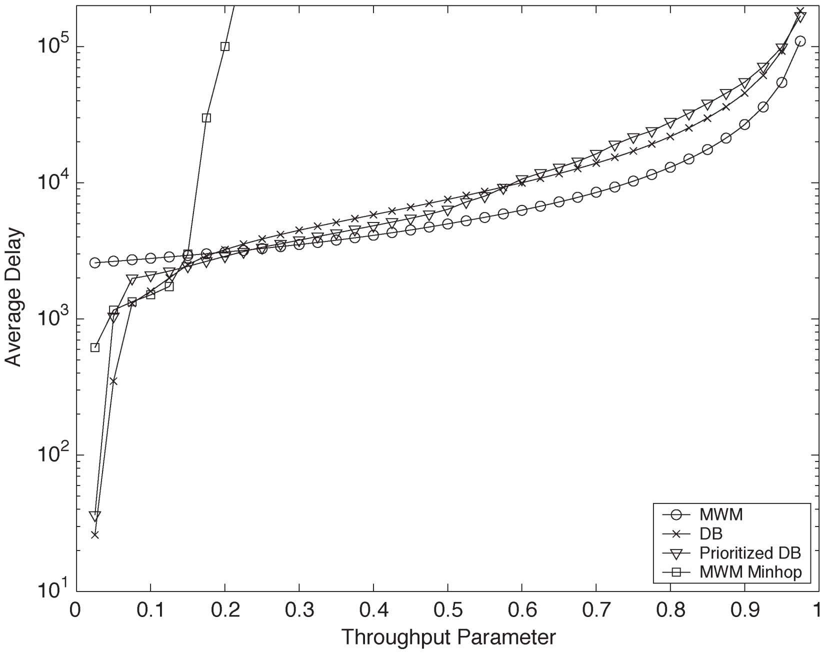

Fig. 8. Average delay for a range of throughput levels.

has chosen a logical topology, it seems reasonable to transmit those packets that are one hop from departure prior to the multihop packets scheduled by DB. Thus, Prioritized DB uses DB for joint logical topology-reconfiguration decisions and IP-layer routing, with the caveat that any nonempty VOQs one hop from departure are serviced with priority.

In general, given D, in our simulations we choose a frame size 10% in excess of the minimum value required for stability, in order to mitigate the probability of large deviations in the queue occupancies.

C. Circuit Versus Packet Switching

It is certainly true that statistical multiplexing from packet switching makes efficient use of link bandwidth. However, the additional link loads from multihopping data across a network experiencing congestion can lead to oscillation and instability of data flows. Circuit switching is an effective solution in this situation, because heavy loads can efficiently be scheduled over the available capacity. Thus, it makes great intuitive sense that different throughput levels are well served by different degrees of circuit and packet switching. In this section, we address this issue by demonstrating that our stabilizing multihop algorithms naturally transition between circuit and packet switching in order to achieve improved delay performance over the range of achievable throughputs.

For our simulation setup, we generate at each throughput level 25 arrival rate matrices with i.i.d. entries selected uniformly from the interval [0, 1], and normalize the maximum row/column sum to the desired throughput level (this is the throughput parameter). Each of these matrices is then simulated for 20×10^6 time slots, with an initial backlog of zero at each VOQ. Each point on the plots of Figs. 8–10 is the mean value over the 25 sample paths generated for each arrival-rate matrix.

Fig. 8 shows the average delay for our algorithms under $D = 1000$. The single-hop routing algorithm (MWM) is outperformed by all other algorithms in the low-throughput regime. However, for increasing throughputs, MWM is the algorithm with best delay performance. MWM Minhop is unstable outside of the low-throughput regime where the plot shows a significant jump in the delay associated with this algorithm. DB and Prioritized DB are stable across all throughputs, though underperforming MWM at moderate to high throughputs.

To understand the apparent performance tradeoff between the circuit-centric approach (WDM reconfiguration with little or no IP-layer routing) and the packet-centric approach (small amount of WDM reconfiguration with IP-layer routing), we show in Fig. 9 the average fraction of departed packets single hopped in each time slot, and in Fig. 10, the fraction of frames in which reconfiguration was triggered, for all algorithms. We have truncated the data in Fig. 10 because for higher throughputs, all algorithms have a fraction of approximately 1. At low-throughput levels, the best performing algorithms employ a large degree of IP-layer routing, with a small fraction of packets single hopped. Also, WDM-layer reconfiguration is not triggered as often by the multihop algorithms, which implies lower delay associated with reconfiguration overhead. At high throughputs, all algorithms tend to depart more packets through single-hop routes, but the multihop algorithms still employ a significant amount of IP-layer routing, which leads to an overall increased load and lack of performance compared to MWM. All algorithms tend to employ WDM-layer reconfiguration at each frame boundary from a relatively low-throughput level and up.

We conclude that DB and Prioritized DB are attractive algorithms, because of their ability to achieve significant gains through the use of packet routing at low throughputs and an increased tendency towards WDM reconfiguration with single-hop routing at the IP layer at high throughputs. These

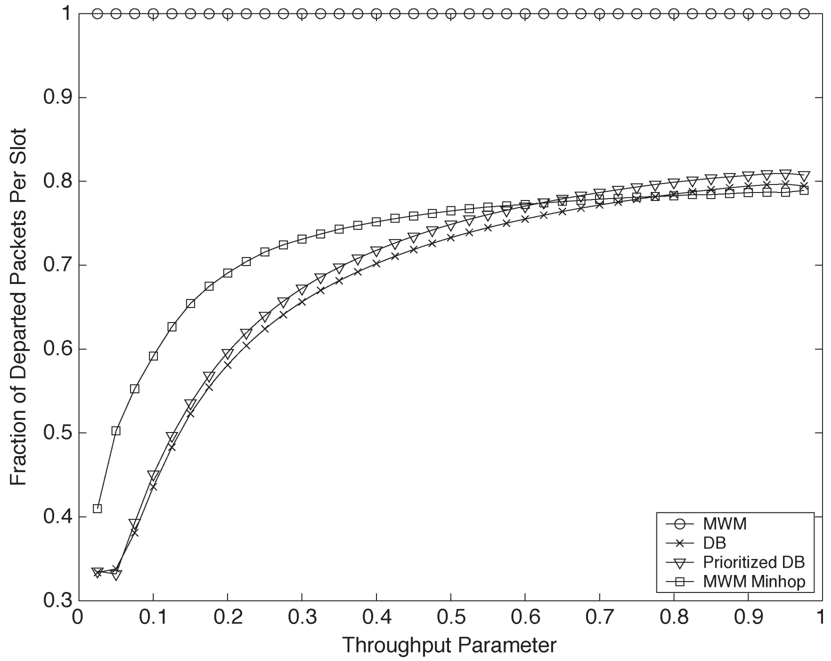

Fig. 9. Fraction of departed packets single hopped per time slot.

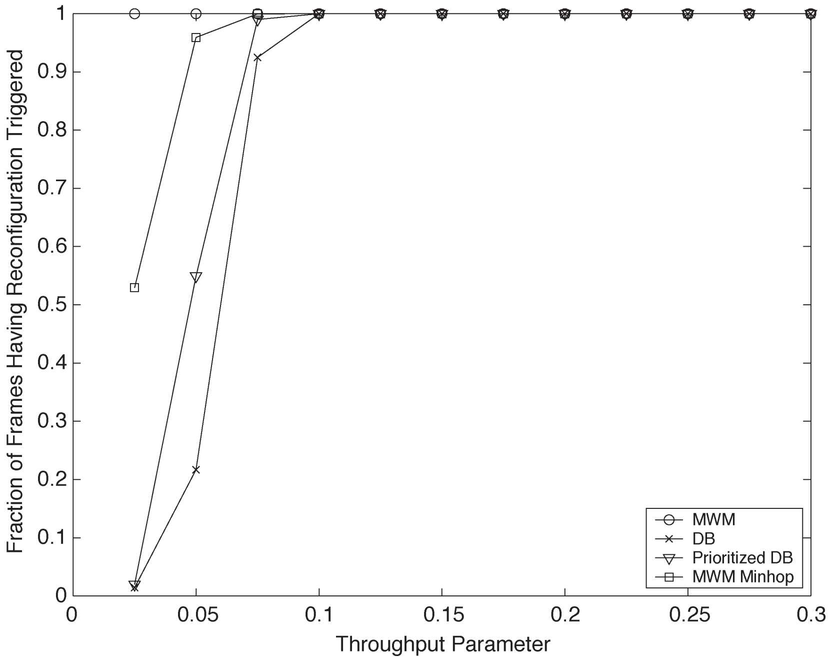

Fig. 10. Fraction of frames in which a reconfiguration was initiated.

algorithms effectively transition between packet switching and circuit switching, and require no knowledge of the traffic arrival process other than the value of δ.

D. Frame- Versus Bias-Based Algorithms

The intuitive motivation for introducing the bias-based algorithm AB in this work is that a reconfiguration algorithm that does not make decisions at fixed intervals may be able to better adapt to actual traffic variations as they happen. Fig. 11 provides simulation results demonstrating the validity of this argument. The simulation scenario has six nodes, a uniform arrival rate matrix of $\lambda_{i,j} = 0.04 \; \forall i \neq j$ (low-throughput scenario), and Bernoulli arrivals, under algorithm DB. Since our algorithms are intended to be implemented at a particular value of frame size F or bias size b, we note that for an appropriately chosen bias size, there is tremendous benefit to using the bias-based algorithm in lieu of the frame-based scheme.

E. Random-Ring Algorithms

In this section, we introduce and analyze a class of randomized algorithms from which the switch-scheduling algorithms of [17] are drawn.

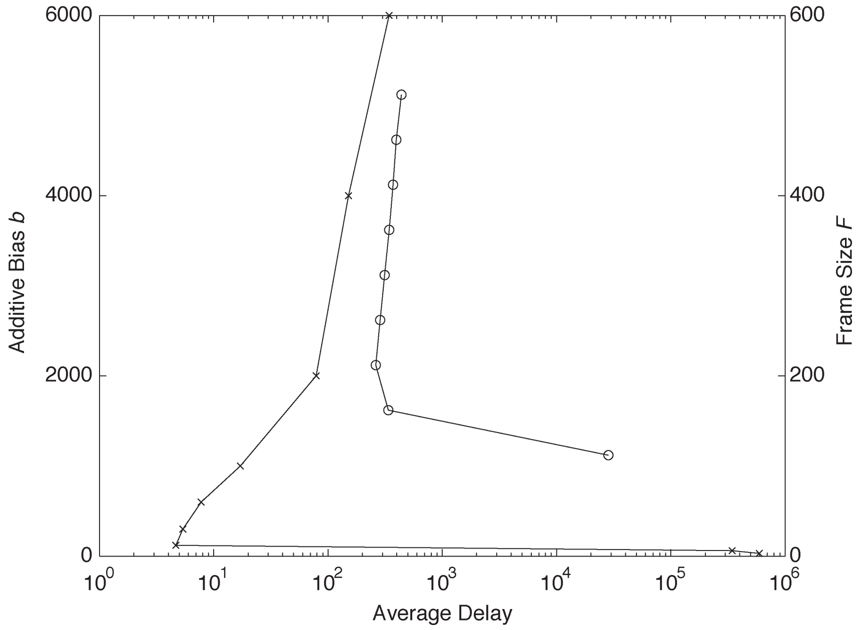

Fig. 11. Frame/bias size versus average simulated delay.

Definition 4.1: The class of random-ring algorithms selects, at each frame boundary, a ring logical topology randomly with equal probability. This class of algorithms includes all possible IP-routing schemes on top of the random logical topology selection.

Clearly, a desirable feature of random-ring algorithms is the low computational complexity associated with choosing a logical topology. Unfortunately, this results in a throughput penalty, as described in the following theorem.

Theorem 4.2: The class of random-ring algorithms is not throughput optimal, in the sense that the stability region of any random-ring algorithm has smaller volume and is a strict subset of the doubly substochastic region.

Proof: See Appendix D. ■

F. Access Network

Consider an access network, where $N-1$ of the nodes (the local nodes) each have a single transceiver, and one node (the hub node) has $P = N-1$ ports. We assume there are N wavelengths so that the only constraints on the allowable logical topologies come from the port constraints. We consider arrival rate matrices λ satisfying

$$\lambda_{i,j} = \begin{cases} 0, & \text{if } i = j \\ \alpha, & \text{if } i = 1 \text{ and } j \neq i \text{, or if } j = 1 \text{ and } i \neq j \\ \beta, & \text{otherwise} \end{cases} \quad (29)$$

where $\alpha > 0$ and $\beta > 0$. From Theorem 3.5, it is easy to see that a stabilizable-rate matrix for $D = 0$ simply must satisfy

$$\alpha + (N-2)\beta < 1. \quad (30)$$

Thus, for F or b chosen appropriately for their respective frame-based algorithms (according to the discussion of Section III-F), we may proceed to investigate the performance tradeoffs of multihop versus single-hop routing for various α and β values.

Fig. 12 plots the data corresponding to the access network under i.i.d. Bernoulli arrivals for a range of α/β values. The plot at the left of Fig. 12 shows that the algorithms based on DB are far superior to MWM for $\alpha/\beta > 1$. We plot the average fraction of frames where reconfiguration was triggered at the right in Fig. 12. It is clear that reconfiguration is in fact unnecessary in this network when the traffic is largely targeted at the hub node. Once the algorithms based on DB choose the logical topology directly connecting each node to the hub node, pure IP-layer routing is employed thereafter. Thus, local traffic among nodes in the access network is easily served by the algorithms based on DB, while MWM suffers from having to reconfigure the logical topology in order to directly service this local traffic. We have omitted the data corresponding to the MWM Minhop algorithm, because of its extremely poor performance (orders of magnitude worse) next to MWM.

V. Conclusion

We have studied algorithms for joint WDM reconfiguration and IP-layer routing in IP-over-WDM networks. The key algorithms (MWM and DB) operate based on maximum weight scheduling, and are asymptotically throughput optimal. We found that optical-layer overhead due to reconfiguration delay is mitigated by frame-based algorithms. We provided fixed-frame- and variable-frame-duration algorithms and proved their stability properties. Our algorithms precisely dictate the control decisions made at each slot at the IP and WDM layers, with DB in general making use of both IP-layer multihop routes and WDM reconfiguration.

In terms of delay performance, there is a great benefit from employing algorithms that tend to use multihop IP-layer routes instead of WDM reconfiguration, when the additional load incurred from these multihop paths is sufficiently small. At high

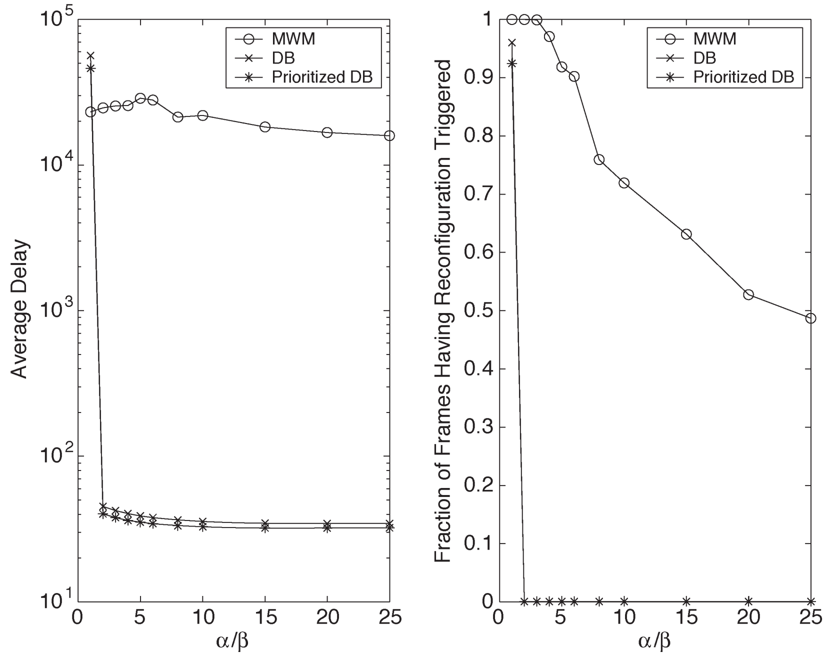

Fig. 12. Average delay (left) and fraction of frames in which a reconfiguration was initiated (right) for a range of α/β values. $N = 6$ nodes, $D = 1000$ time slots. Each nonhub node has an average arrival rate of $\alpha + (N-2)\beta = 0.9$ packets per slot.

system loads, the opposite is true, and WDM reconfiguration is preferable to additional load from multihop IP-layer routing.

We demonstrated theoretically that multihop routing is of no use when reconfiguration delay is negligible in the three-node scenario. Further, we showed that simple algorithms employing random-ring selection at the WDM layer are not capable of achieving throughput optimality.

An important direction for future research is to gain some traction on analytically establishing performance tradeoffs between algorithms employing different degrees of reconfiguration/routing. Switching theory has provided bounds on performance of scheduling algorithms (e.g., [11]), but much work remains before algorithm performance can be compared under various arrival processes. In terms of scheduling, WANs cannot easily accommodate the burden of passing full state information to all nodes in the network, because of problems with scalability and large delays. Thus, distributed scheduling algorithms for networks with large delays are an important design objective.

Appendix A
Proof of Lemma 3.1

Under the bias-based scheduling algorithm, (12) implies the following additional property of the system dynamics:

$$\langle v, X(n)\rangle < \max_{v'}\left\{\langle v', X(n)\rangle + b1_{\{v'=v(n-1)\}}\right\}$$

implies that Q_v is not increasing at time n:

The fluid-limit version of this property is then given by

$$\langle v, \bar{x}(t)\rangle < \max_{v'}\left\{\langle v', \bar{x}(t)\rangle\right\}$$

implies that $\bar{q}_v$ is not increasing at time t.

The remainder of the proof follows closely with the proof of [14, Lemma 3]. Denote the quadratic Lyapunov function L by $L(X) = (1/2)\sum_{i,j} X_{i,j}^2$. Then, for any $t \geq 0$ such that $L(\bar{x}(t)) > 0$

$$\frac{\mathrm{d}}{\mathrm{d}t} L\left(\bar{x}(t)\right) = \sum_{i,j} \bar{x}_{i,j}(t)\left(\lambda_{i,j} - \dot{\bar{u}}_{i,j}(t)\right) \tag{31}$$

$$= \sum_{i,j} \bar{x}_{i,j}(t)\left(\lambda_{i,j} - \sum_{v\in\mathcal{V}} v_{i,j}\dot{\bar{q}}_v(t)\right) \tag{32}$$

$$= \sum_{i,j} \bar{x}_{i,j}(t)\left(\lambda_{i,j} - v_{i,j}^{\mathrm{dom}}\right) + \sum_{i,j} \bar{x}_{i,j}(t) v_{i,j}^{\mathrm{dom}} - \left(1 - \dot{\bar{q}}_R(t)\right) \times \max_{v\in\mathcal{V}} \sum_{i,j} \bar{x}_{i,j}(t) v_{i,j}. \tag{33}$$

Here, (31) and (32) follow from the fluid equations for the system. Setting $\mathcal{V}'$ at time t to be the set of logical topologies v satisfying $\langle v, \bar{x}(t)\rangle = \max_{v'}\langle v', \bar{x}(t)\rangle$, we have that $\sum_{v\in\mathcal{V}'} \dot{\bar{q}}_v(t) + \dot{\bar{q}}_R(t) = 1$. Since λ is chosen to be doubly substochastic with all row/column sums strictly less than $1-\delta$, there exists another doubly substochastic matrix v^{dom}, with maximum row or column sum equal to $1-\delta$, and whose entries are all greater than the entries of λ. Thus, (33) follows. Now, we have

$$\sum_{i,j} \bar{x}_{i,j}(t)\left(\lambda_{i,j} - v_{i,j}^{\mathrm{dom}}\right) \leq \left(\min_{i,j}\left(v_{i,j}^{\mathrm{dom}} - \lambda_{i,j}\right)\right)\sum_{i,j} \bar{x}_{i,j}(t)$$

$$= -\varepsilon \sum_{i,j} \bar{x}_{i,j}(t) \tag{34}$$

where $\varepsilon > 0$. Also, noting that matrix $v^{\mathrm{dom}}/(1-\delta)$ is a doubly substochastic matrix, and supposing $\dot{\bar{q}}_R(t) \le \delta$ for all $t \ge 0$, we have

$$\sum_{i,j}\bar{x}_{i,j}(t)v^{\mathrm{dom}}_{i,j} - (1-\dot{\bar{q}}_R(t))\max_{v\in\mathcal{V}}\sum_{i,j}\bar{x}_{i,j}(t)v_{i,j} \tag{35}$$

$$\le (1-\delta)\left(\sum_{i,j}\bar{x}_{i,j}\frac{v^{\mathrm{dom}}_{i,j}}{1-\delta} - \max_{v\in\mathcal{V}}\sum_{i,j}\bar{x}_{i,j}v_{i,j}\right) \tag{36}$$

$$\le 0. \tag{37}$$

Here, (37) follows by well-known properties of the convex doubly substochastic region (for instance, see [18, Lemma 2]).

Combining (33), (34), and (37), we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}L\left(\bar{x}(t)\right) \le -\varepsilon\sum_{i,j}\bar{x}_{i,j}(t). \tag{38}$$

Since $\varepsilon > 0$, it can be shown (similar to [11]) that this is a sufficient condition to guarantee stability.

Appendix B
Proof of Theorem 3.5

Definition B.1: Matrix $\lambda = (\lambda_{i,j}, i,j = 1,\ldots,N)$ is called doubly underloaded if it satisfies (28). Furthermore, if all inequalities in (28) are satisfied with equality, λ is called doubly loaded, while if all inequalities in (28) are strict, λ is called strictly doubly underloaded.

The theorem proof is accomplished in several steps, similar to [16]. First, von Neumann's result for finding a doubly stochastic matrix that dominates (entry-by-entry) a doubly substochastic matrix is extended to find a doubly loaded matrix that dominates a doubly underloaded matrix. Secondly, an algorithm is derived for constructing a bipartite graph based on a doubly loaded matrix, with the property that the graph has a maximum matching that includes all nodes. Finally, an algorithm for expressing any doubly loaded matrix as a convex combination of allowed logical topologies is provided.

A. Extending von Neumann's Result

Given doubly underloaded matrix λ, if the summation over the elements of λ is less than $\sum_i P_i$, then there must exist k, l such that $\sum_j \lambda_{k,j} < P_k$ and $\sum_i \lambda_{i,l} < P_l$. This follows easily: Suppose that no such k can be found. Then, $\sum_j \lambda_{k,j} \ge P_k, \forall k$, and since the matrix is doubly underloaded, $\sum_j \lambda_{k,j} = P_k, \forall k$. This implies that $\sum_k\sum_j \lambda_{k,j} = \sum_k P_k$, which violates our initial assumption. An identical argument applies to the value of l. Thus, k, l must exist, and the entry $\lambda_{k,l}$ should be increased to $\lambda + \min\{P_k - \sum_j \lambda_{k,j}, P_l - \sum_i \lambda_{i,l}\}$. Repeating this process at most $2N-1$ times (once for each row/column with the final entry loading both a row and a column simultaneously), a doubly loaded matrix is achieved. The following lemma summarizes this result.

Lemma B.1: Given a doubly underloaded matrix λ, there exists a doubly loaded matrix $\tilde{\lambda} = (\tilde{\lambda}_{i,j}, i,j = 1,\ldots,N)$ that dominates λ entry-by-entry: $\tilde{\lambda}_{i,j} \ge \lambda_{i,j}, \forall i,j$.

B. Bipartite Graph From a Doubly Loaded Matrix

Given doubly loaded matrix $\tilde{\lambda}$, we now construct a corresponding bipartite graph for which Hall's Theorem guarantees the existence of a maximum matching covering all nodes. This maximum matching may subsequently be translated to a valid logical topology. Designate the nodes of the two bipartitions by

$$S = \left\{s_1^1, s_1^2, \ldots, s_1^{P_1}, s_2^1, \ldots, s_2^{P_2}, \ldots, s_N^1, \ldots, s_N^{P_N}\right\}$$

$$D = \left\{d_1^1, d_1^2, \ldots, d_1^{P_1}, d_2^1, \ldots, d_2^{P_2}, \ldots, d_N^1, \ldots, d_N^{P_N}\right\}.$$

Above, S and D represent source and destination ports, respectively. Algorithm B.1 establishes edges between the nodes of S and D.

Algorithm B.1: Let $\phi = \tilde{\lambda}$. Associate with each node n a bin b_n, initially empty and having maximum capacity 1. Consider in turn each element $\phi_{i,j}$ of matrix ϕ, repeating the following steps until $\phi_{i,j} = 0$:

1) Obtain $k = \min\{m : b_{s_i^m} < 1\}$, and $l = \min\{m : b_{d_j^m} < 1\}$.
2) Add an edge joining s_i^k to d_j^l if no such edge exists.
3) Obtain $y_{i,j} = \min\{\phi_{i,j}, 1 - b_{s_i^k}, 1 - b_{d_j^l}\}$.
4) Set $\phi_{i,j} \leftarrow \phi_{i,j} - y_{i,j}$, $b_{s_i^k} \leftarrow b_{s_i^k} + y_{i,j}$, and $b_{d_j^l} \leftarrow b_{d_j^l} + y_{i,j}$.

For a doubly loaded matrix $\tilde{\lambda}$, upon algorithm completion, it is simple to show that each bin is at capacity: Suppose $b_{s_i^k} < 1$. Then, if there is no j such that $\phi_{i,j} > 0$, it must be true that $\sum_j \tilde{\lambda}_{i,j} \le P_i - (1 - b_{s_i^k}) < P_i$. This follows because each time matrix entry element $\phi_{i,j}$ is decreased, one of the bins at source i (one of $b_{s_i^1}, \ldots, b_{s_i^{P_i}}$) is increased by the same amount. Since we have assumed the entire ith row of ϕ is zero, then the sum over the same bins must equal the initial ith row sum of matrix ϕ, or equivalently $\sum_j \tilde{\lambda}_{i,j}$. This sum must be less than P_i since all source i bins are not full, which provides a contradiction to our assumption that $\tilde{\lambda}$ is doubly loaded. The argument for $b_{d_j^l} < 1$ follows similarly.

Alternatively, if $b_{s_i^k} < 1$, and there exists j such that $\phi_{i,j} > 0$, then there must exist a value l such that $b_{d_j^l} < 1$. This follows because $\phi_{i,j}$ has not been reduced to zero, which implies that the full column sum of P_j has not been distributed over the P_j bins corresponding to ports at destination j. Thus, the algorithm would have discovered source and destination bins with which to reduce $\phi_{i,j}$ further, which contradicts that the algorithm has terminated.

For each (i,j), the algorithm reduces $\phi_{i,j}$ to zero in at most $2\min\{P_i, P_j\} - 1$ steps, because this is the maximum number of times that the minimizing term $y_{i,j}$ does not have to equal $\phi_{i,j}$. Thus, we have shown that the algorithm terminates, and that all bins are full (at unit capacity) upon termination.

We now show that the bipartite graph constructed by the above algorithm satisfies the condition of Hall's Theorem to guarantee the existence of a saturated matching (a matching that covers each node). Take any set of source nodes $\mathcal{S} \subseteq S$. Then, we require that this set connects to at least $|\mathcal{S}|$ destination nodes in D.

A useful way of considering each bin in the algorithm is as a measure of the flow departing (in the case of a source node bin) or arriving (in the case of a destination node bin) at that port. As each link is added in the algorithm, an element of matrix ϕ is reduced by some amount, and the bins associated with the source and destination nodes of that link are increased by the same amount. This captures the amount of flow serviced from the source to the destination along that link.

Upon algorithm termination, each bin is at unit capacity, which equivalently means that one unit of flow departs from each source node and arrives at each destination node. Thus, since $\mathcal{S}$ is the source of $|\mathcal{S}|$ units of flow, at least $|\mathcal{S}|$ units of flow must arrive to the destination nodes. Further, since each destination bin has unit capacity, this flow must arrive along at least $|\mathcal{S}|$ links. Thus, we have that the set of neighbor nodes to $\mathcal{S}$ must have size at least $|\mathcal{S}|$. Applying Hall's Matching Theorem [19], a saturated matching is guaranteed. The following lemma summarizes this result.

Lemma B.2: The bipartite graph generated by Algorithm B.1 has a saturated matching.

C. Translating a Saturated Matching on the Bipartite Graph Into a Logical Topology

Beginning with $N \times N$ matrix $v = 0$, for each edge (s_i^k, d_j^l) in the saturated matching, increment $v_{i,j}$ by 1. Once each edge has been considered, matrix v must have ith row sum P_i and jth column sum P_j. This follows because the matching on the bipartite graph is saturated, and thus, source i is associated with P_i nodes with edges in the matching, and destination j is associated with P_j nodes with edges in the matching. Thus, v is a valid logical topology under the port distribution $\{P_i\}_{i=1}^N$. Finally, by the construction of Algorithm B.1 it is clear that a nonzero element in v implies that the corresponding entry of $\tilde{\lambda}$ is nonzero. The following lemma summarizes this result.

Lemma B.3: For a bipartite graph obtained according to Algorithm B.1, the graph may be translated to a corresponding logical topology whose incidence matrix has ith row sum equal to P_i and jth column sum equal to P_j (we refer to this as a saturated logical topology). Furthermore, the entries at which this incidence matrix is nonzero has corresponding entries in $\tilde{\lambda}$ that are nonzero.

D. Proof of Theorem 3.5

Given a doubly underloaded matrix λ, Lemma B.1 guarantees the existence of a matrix $\tilde{\lambda}$ that is doubly loaded and that is entry-by-entry dominant over λ. Applying Algorithm B.1 to $\tilde{\lambda}$, Lemmas B.2 and B.3 guarantee the existence of a saturated logical topology where each link has nonzero associated rate in the doubly loaded rate matrix $\tilde{\lambda}$. The following algorithm capitalizes on this to decompose $\tilde{\lambda}$ as a convex combination of valid logical topology incidence matrices. This algorithm is the natural generalization of the decomposition presented in [16].

Algorithm B.2: Begin with doubly loaded matrix $\omega = \tilde{\lambda}$. Repeat the following steps until $\omega = 0$. At the nth step of the algorithm, do the following steps:

1) For matrix ω, find a saturated logical topology v^n according to Algorithm B.1 and Lemmas B.2 and B.3.
2) Set $\alpha_n = \min\{\omega_{i,j}/v_{i,j}^n : v_{i,j}^n > 0, \forall i, j\}$.
3) Set $\omega \leftarrow (1/(1-\alpha_n))(\omega - \alpha_n v^n)$.

Since the logical topology found for a doubly loaded matrix is saturating, step n of the algorithm reduces the ith row sum by $\alpha_n P_i$, and the jth column sum by $\alpha_n P_j$. Thus, all row and column sums are reduced by a factor of $1 - \alpha_n$ at each iteration. For this reason, the scale factor of $1 - \alpha_n$ is applied at each iteration to bring the matrix back to a doubly loaded matrix. Finally, since at each iteration, α is chosen to reduce at least one matrix element to 0, with at least N elements reduced to 0 at once at the last step, the decomposition takes at most $N^2 - N + 1$ steps to complete. $\tilde{\lambda}$ may then be expressed as

$$\tilde{\lambda} = \sum_{n=1}^{N^2-N+1} \left(\alpha_n \prod_{k=1}^{n-1} (1 - \alpha_k) \right) v^n.$$

The fact that the weights sum to unity is guaranteed by the property that each logical topology in the decomposition is saturating.

Appendix C
Proof of Theorem 4.1

The proof is by induction, using a stochastic coupling argument [20]. We begin with algorithm $\mathcal{P}_0$, and successively refine it at each time to an algorithm with improved average expected aggregate backlog. The recursion implies that an algorithm with no forwarding produces smaller or equal average aggregate backlog. For this proof, at step $n-1$ of the induction, assume that arrivals under algorithms $\mathcal{P}_{n-1}$ and $\mathcal{P}_n$ are coupled to the same queues for all time. Quantities marked with a tilde symbol, such as $\tilde{X}$, correspond to algorithm $\mathcal{P}_n$, while those without a tilde symbol correspond to algorithm $\mathcal{P}_{n-1}$.

Suppose we have algorithm $\mathcal{P}_{n-1}$ for $n \geq 1$ and consider time $n-1$. By the recursion, up to and including time $n-1$, algorithm $\mathcal{P}_{n-1}$ does not forward any packets. At time n, if $\mathcal{P}_{n-1}$ does not forward any packets, then let $\mathcal{P}_n$ choose the same controls as $\mathcal{P}_{n-1}$ for all time. If $\mathcal{P}_{n-1}$ does forward one or more packets, let $\mathcal{P}_n$ choose the same controls as $\mathcal{P}_{n-1}$ up to time $n-1$. At time n, we must consider three cases. For all time after n, let $\mathcal{P}_n$ attempt to mimic $\mathcal{P}_{n-1}$ in its controls, only deviating from $\mathcal{P}_{n-1}$ if there simply is no packet in a queue under $\mathcal{P}_n$ where, for the corresponding queue under $\mathcal{P}_{n-1}$, a packet is forwarded or departs the system.

Case 1: If $\mathcal{P}_{n-1}$ forwards only a single packet along link (a, b), then note that for any link (a, b), there are only two possible logical topologies containing this link. These configurations are $\{(a, b), (b, c), (c, a)\}$ and $\{(a, b), (b, a)\}$. For either configuration, link (a, b) is being used to forward a packet from $\text{VOQ}_{a,c}$ to $\text{VOQ}_{b,c}$. Let $X(n-1) = (X_{a,b}, X_{b,c}, X_{c,a}, X_{a,c}, X_{c,b}, X_{b,a})$ be the vectorized queue-backlog matrix at time $n-1$. For the first configuration containing link (a,b), algorithm $\mathcal{P}_{n-1}$ results in the following queue occupancy at time n:

$$X(n) = X(n-1) + a(n) + (0, -u_{b,c}(n)+1, -u_{c,a}(n), -1, 0, 0)\,.$$

Since $-u_{b,c}(n)+1 \geq 0$, it is sufficient to let $\mathcal{P}_n$ employ a logical configuration that allows packets to depart from the $\mathrm{VOQ}_{c,a}$ and $\mathrm{VOQ}_{a,c}$. This is clearly an allowable control, and thus, $\mathcal{P}_n$ results in the queue-occupancy distribution

$$\tilde{X}(n) = X(n-1) + a(n) + (0, 0, -u_{c,a}(n), -1, 0, 0)\,.$$

For the second possible configuration containing link (a,b), the queue-occupancy distributions at time n are

$$X(n) = X(n-1) + a(n) + (0, 1, 0, -1, 0, -u_{b,a}(n))$$
$$\tilde{X}(n) = X(n-1) + a(n) + (0, 0, 0, -1, 0, -u_{b,a}(n))\,.$$

Here, $\mathcal{P}_n$ chooses the configuration that allows packets from the $\mathrm{VOQ}_{a,c}$ and $\mathrm{VOQ}_{b,a}$ to exit the system.

For either case, it is clear that $\mathcal{P}_n$ has an improved or equal aggregate queue occupancy at each time after n.

Case 2: If $\mathcal{P}_{n-1}$ forwards two packets, there are three possible sets of links that are used for forwarding: $\{(a,b),(b,c)\}$, $\{(a,b),(b,a)\}$, or $\{(a,b),(c,a)\}$. Note that each of these sets of links determines the switch configuration chosen by the switching algorithm. We consider each of these cases in turn. If $\mathcal{P}_{n-1}$ forwards packets along links (a,b) and (b,c), then $\mathcal{P}_n$ has chosen switch configuration $\{(a,b),(b,c),(c,a)\}$. The queue-occupancy distributions under the policies are then given by

$$X(n) = X(n-1) + a(n) + (0, 1, -u_{c,a}(n)+1, -1, 0, -1)$$
$$\tilde{X}(n) = X(n-1) + a(n) + (0, 0, 0, -1, 0, -1)\,.$$

Here, algorithm $\mathcal{P}_n$ chooses the switch configuration that allows packets from $\mathrm{VOQ}_{a,c}$ and $\mathrm{VOQ}_{b,a}$ to exit the system.

If $\mathcal{P}_{n-1}$ forwards packets along links (a,b) and (c,a), then $\mathcal{P}_{n-1}$ has again chosen switch configuration $\{(a,b),(b,c),(c,a)\}$. The queue-occupancy distributions under the policies are then given by

$$X(n) = X(n-1) + a(n) + (1, -u_{b,c}(n)+1, 0, -1, -1, 0)$$
$$\tilde{X}(n) = X(n-1) + a(n) + (0, 0, 0, -1, -1, 0)\,.$$

Here, algorithm $\mathcal{P}_n$ chooses the switch configuration that allows packets from $\mathrm{VOQ}_{a,c}$ and $\mathrm{VOQ}_{c,b}$ to exit the system.

Finally, if $\mathcal{P}_{n-1}$ forwards packets along links (a,b) and (b,a), then $\mathcal{P}_{n-1}$ has chosen switch configuration $\{(a,b),(b,a)\}$. The queue-occupancy distributions under the policies are then given by

$$X(n) = X(n-1) + a(n) + (0, -1+1, 0, -1+1, 0, 0)$$
$$\tilde{X}(n) = X(n-1) + a(n) + (0, 0, 0, 0, 0, 0)\,.$$

Here, algorithm $\mathcal{P}_n$ does nothing because $\mathcal{P}_{n-1}$ has effectively made no change to its occupancy distribution.

It is clear that in all cases, $\mathcal{P}_n$ has an improved or equal aggregate queue occupancy at each time after $n-1$.

Case 3: If $\mathcal{P}_{n-1}$ forwards three packets, then the switch configuration must be $\{(a,b),(b,c),(c,a)\}$. The queue-occupancy distributions under the policies are then given by

$$X(n) = X(n-1) + a(n) + (1, 1, 1, -1, -1, -1)$$
$$\tilde{X}(n) = X(n-1) + a(n) + (0, 0, 0, -1, -1, -1)\,.$$

Here, algorithm $\mathcal{P}_n$ chooses the switch configuration $\{(a,c),(c,b),(b,a)\}$ to allow packets from $\mathrm{VOQ}_{a,c}$, $\mathrm{VOQ}_{c,b}$, and $\mathrm{VOQ}_{b,a}$ to exit the system. Again, it is clear that $\mathcal{P}_n$ results in an improved aggregate queue occupancy at each time after $n-1$.

Appendix D
Proof of Theorem 4.2

For this proof, we invoke the multihop parameters described in Section II. The proof follows for any $D \geq 0$. Denote by $\mathcal{V}^r \subset \mathcal{V}$ the set of logical topology matrices corresponding to logical rings of size N. An arrival rate matrix is stabilizable if there exists a subprobability measure $(\phi_E, E \in \mathcal{E})$ such that

$$\sum_{E\in\mathcal{E}} \phi_E \leq 1 \tag{39}$$

$$\sum_{E\in\mathcal{E}} \phi_E R^j E_{:,j} > \lambda_{:,j}, \qquad j = 1,\ldots,N. \tag{40}$$

The reasoning here is that under some joint reconfiguration and routing algorithm, the variable ϕ_E represents the long-term fraction of time allocated to activation matrix E. Thus, if an arrival rate matrix λ may be dominated as in (40), then there exists a stabilizing control strategy. Indeed, the subprobability measure weights may be used to form a stabilizing TDM schedule over the activation matrices $\mathcal{E}$, so long as the inter-reconfiguration times are made sufficiently large to account for the idleness due to reconfiguration overhead.

Since there are $(N-1)!$ different logical rings having N nodes, it is clear that under any random-ring algorithm, the long-term amount of time allocated to each ring is $1/(N-1)!$. Thus, the subprobability measures $(\phi_E, E \in \mathcal{E})$

achievable under a random-ring algorithm are restricted to the form

$$\phi_E = \sum_{v \in \mathcal{V}^r} \frac{\phi_{E|v}}{(N-1)!}, \qquad E \in \mathcal{E}$$

where $\sum_E \phi_{E|v} = 1$ for all $v \in \mathcal{V}^r$, and $\phi_{E|v} > 0$ only if E is an allowed activation matrix under logical ring v.

For $j = 1, \ldots, N$, we may now express the left-hand side of (40) as

$$\sum_{E \in \mathcal{E}} \sum_{v \in \mathcal{V}^r} \frac{\phi_{E|v}}{(N-1)!} R^j E_{:,j} \tag{41}$$

$$= \frac{1}{(N-1)!} \sum_{v \in \mathcal{V}^r} \sum_{E \in \mathcal{E}} \phi_{E|v} R^j E_{:,j}. \tag{42}$$

Now, $(\phi_{E|v}, E \in \mathcal{E})$ has no restrictions other than to be a subprobability measure restricted to logical ring v. Consider the set of arrival rate matrices that are strictly dominated by the inner summation in (42), as we range over the compact set of feasible subprobability measures $(\phi_{E|v}, E \in \mathcal{E})$. This set of arrival rate matrices must be equal (up to a set of measure zero) to the stability region corresponding to electronic routing over a fixed logical ring. Thus, the set of stabilizable arrival rate matrices for the class of random-ring algorithms has outer bound equal to the average over the $(N-1)!$ fixed-ring stability regions. Since each fixed-ring stability region clearly has smaller volume than the doubly substochastic region, the result follows.

References

[1] A. Narula-Tam and E. Modiano, "Dynamic load balancing in WDM networks with and without wavelength constraints," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 10, pp. 1972–1979, Oct. 2000.

[2] J.-F. P. Labourdette and A. S. Acampora, "Logically rearrangeable multihop lightwave networks," *IEEE Trans. Commun.*, vol. 39, no. 8, pp. 1223–1230, Aug. 1991.

[3] J.-F. P. Labourdette, F. W. Hart, and A. S. Acampora, "Branch-exchange sequences for reconfiguration of lightwave networks," *IEEE Trans. Commun.*, vol. 42, no. 10, pp. 2822–2832, Oct. 1994.

[4] I. Baldine and G. Rouskas, "Traffic adaptive WDM networks: A study of reconfiguration issues," *J. Lightw. Technol.*, vol. 19, no. 4, pp. 433–455, Apr. 2001.

[5] I. Widjaja, I. Saniee, R. Giles, and D. Mitra, "Light core and intelligent edge for a flexible, thin-layered and cost-effective optical transport network," *IEEE Commun. Mag.*, vol. 41, no. 5, pp. S30–S36, May 2003.

[6] K. Ross, N. Bambos, K. Kumaran, I. Saniee, and I. Widjaja, "Scheduling bursts in time-domain wavelength interleaved networks," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 9, pp. 1441–1451, Nov. 2003.

[7] J. G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," in *IEEE Proc. Information Communications (INFOCOM)*, Tel Aviv, Israel, 2000, pp. 556–564.

[8] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Automat. Contr.*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[9] D. L. Mills, "Internet time synchronization: The network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.

[10] I. Olkin and A. W. Marshall, *Inequalities: Theory of Majorization and Its Applications*. New York: Academic, 1979.

[11] E. Leonardi, M. Mellia, F. Neri, and M. A. Marsan, "Bounds on average delays and queue size averages and variances in input-queued cell-based switches," in *IEEE Proc. Information Communications (INFOCOM)*, Anchorage, AK, 2001, pp. 1095–1103.

[12] A. Stolyar, "Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic," *Ann. Appl. Probab.*, vol. 14, no. 1, pp. 1–53, Jan. 2004.

[13] S. Meyn and R. Tweedie, *Markov Chains and Stochastic Stability*. London, U.K.: Springer-Verlag, 1996.

[14] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queueing system with asynchronously varying service rates," *Probab. Eng. Inf. Sci.*, vol. 18, no. 2, pp. 191–217, 2004.

[15] A. L. Dulmage and N. S. Mendelsohn, "Matrices associated with the Hitchcock problem," *J. ACM*, vol. 9, no. 4, pp. 409–418, Oct. 1962.

[16] C. Chang, W. Chen, and H. Huang, "Birkhoff-von Neumann input buffered crossbar switches," in *IEEE Proc. Information Communications (INFOCOM)*, Tel Aviv, Israel, 2000, pp. 1614–1623.

[17] A. Bianco, P. Giaccone, E. Leonardi, F. Neri, and R. Brusin, "Multihop scheduling for optical switches with large reconfiguration overhead," in *IEEE Workshop High Performance Switching and Routing (HPSR)*, Phoenix, AZ, 2004, pp. 193–197.

[18] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260–1267, Aug. 1999.

[19] D. B. West, *Introduction to Graph Theory*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[20] D. Stoyan, *Comparison Methods for Queues and Other Stochastic Models*. New York: Wiley, 1983.

Andrew Brzezinski (S'00) received the B.A.Sc. degree in electrical engineering from the University of Toronto, Canada, in 2000, the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 2002, and is currently working toward the Ph.D. degree in electrical engineering at the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology (MIT), Cambridge, MA.

The major focus of his research is in the area of high-speed communication networks. He is particularly interested in developing and studying new algorithms, architectures, and technologies that enhance network efficiency, reduce start-up and operating costs, and provide the end-user with an improved networking experience. His research pursuits have led to interesting applications and/or results in the areas of switching theory, graph theory, control of stochastic networks, queuing analysis, and information theory.

Eytan Modiano (S'90–M'93–SM'00) received the B.S. degree in electrical engineering and computer science from the University of Connecticut, Storrs, in 1986, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, MD, in 1989 and 1992, respectively.

Between 1987 and 1992, he was a Naval Research Laboratory Fellow, and during 1992–1993 was a National Research Council Post Doctoral Fellow, while conducting research on security and performance issues in distributed network protocols. Between 1993 and 1999, he was with the Communications Division, MIT Lincoln Laboratory, where he designed communication protocols for satellite, wireless, and optical networks, and was the project leader for MIT Lincoln Laboratory's Next Generation Internet (NGI) project. He joined the MIT faculty in 1999, where he is currently an Associate Professor at the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems (LIDS). His research is on communication networks and protocols with emphasis on satellite, wireless, and optical networks.

Dr. Modiano is currently an Associate Editor for Communication Networks for IEEE TRANSACTIONS ON INFORMATION THEORY and for *The International Journal of Satellite Communications*. He had served as a Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC) special issue on wavelength division multiplexing (WDM) network architectures, the *Computer Networks Journal* special issue on Broadband Internet Access, the *Journal of Communications and Networks* special issue on Wireless Ad-Hoc Networks, and for the IEEE JOURNAL OF LIGHTWAVE TECHNOLOGY special issue on Optical Networks. He is the Technical Program Co-Chair for Wiopt 2006 and Vice-Chair for Infocom 2007.