Dynamic reconfiguration and routing algorithms for IP-over-WDM networks with stochastic traffic

Andrew Brzezinski and Eytan Modiano
{brzezin,modiano}@mit.edu
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

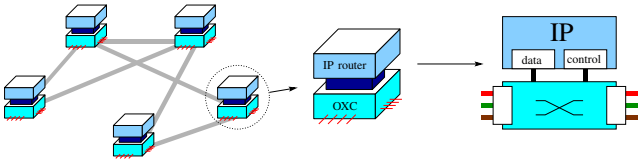
***Abstract*— We develop algorithms for joint IP layer routing and WDM logical topology reconfiguration in IP-over-WDM networks experiencing stochastic traffic. At the WDM layer, we associate a non-negligible tuning latency with WDM reconfiguration, during which time tuned transceivers cannot service backlogged data. The IP layer is modeled as a queueing system. We demonstrate that our algorithms achieve asymptotic throughput optimality by using frame-based maximum weight scheduling decisions. We study both deterministic and random 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. Our algorithms remain valid under a variety of optical layer constraints. We 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 trade-off between WDM reconfiguration and IP layer routing. We find that multi-hop 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 multi-hop IP layer routes.

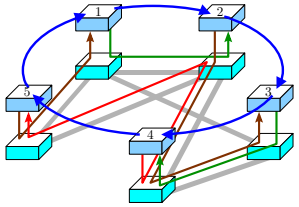
I. Introduction

We consider an optical network architecture consisting of nodes having IP routers overlaying optical cross-connect, with the nodes inteconnected by optical fiber, as in Fig. 1(a). This constitutes the *physical topology* of the network. Optical add/drop multiplexers (ADMs) and optical cross-connects (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.

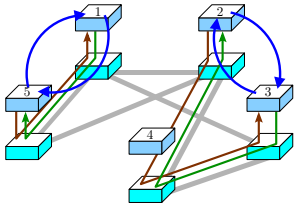
By enabling the transceivers[1] at the nodes to be *tunable*, the network allows for changes in the logical topology configuration. This capability is attractive, because it allows for dynamic reconfiguration algorithms to be employed in order to improve the throughput and delay properties of the network, as well as recover from network failures. In essence, a trade-off emerges between lightpath reconfiguration at the WDM layer and routing at the electronic layer. Fig. 1 depicts the architecture of interest, for a particular 5-node physical topology. Figures 1(b) and 1(c) show the cross-layer connections corresponding to two feasible logical topologies on the physical topology of Fig. 1(a).



(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 5-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. 1. Sample physical topology and feasible logical topologies for 3 wavelengths per fiber, one transceiver per node.

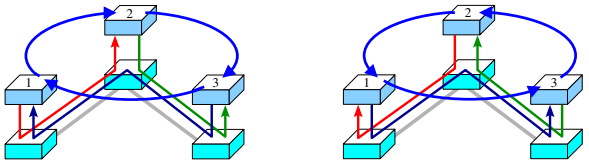
The ability to reconfigure the logical topology requires tunable transceivers and optical cross-connects. The effectiveness of an algorithm employing reconfiguration will depend on the speed with which reconfiguration takes place. In this work, we do not require that the transceivers be fast tunable.

A. Performance trade-off 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 3-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

This work was supported in part by the Defense Advanced Research Projects Agency under grant MDA972-02-1-0021 and by NSF under grants ANI-0073730 and ANI-0335217.

[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.



(a) $\mathcal{C}_1$: Ring $1 \rightarrow 2 \rightarrow 3$. (b) $\mathcal{C}_2$: Ring $1 \rightarrow 3 \rightarrow 2$.

Fig. 2. Lightpath interconnections for 3-node rings on a line physical topology.

(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 tuning latency, packet service delays are affected 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 i-th row and j-th 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 trade-off between incurring reconfiguration overhead associated with tuning latency and additional load from IP layer routing, consider arrival rate matrices Λ_1 and Λ_2 under the 3-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 long term arrival rate 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 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 TDM schedule allocating at least 40% of its time to $\mathcal{C}_1$ and at least 50% of its time to $\mathcal{C}_2$ is sufficient for stability, so long as the contiguous service time allocated to each logical ring is adequately long to make the tuning latency overhead negligible.

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 [2] considers reconfigurable multi-hop networks, and studies the problem of finding a logical topology to minimize maximum link load for a particular traffic demand. In [1], [3], *branch-exchange* algorithms are introduced to *incrementally* adjust the logical topology towards a desired configuration (here, [3] approaches the problem in a static scenario, while [1] approaches the problem under changing traffic). The work of [4] associates for each time 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. Lastly, the TWIN network architecture of [5], [6] looks at the network in a more granular manner, at the packet level, and reduces the optical transport network to a modified switch scheduling problem. TWIN relies 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 non-negligible link transmission delay.

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 work, 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 trade-offs between reconfiguration at the optical layer and packet routing at the IP layer. Our major contributions are:

1) We develop mechanisms for dynamically triggering WDM reconfiguration under stochastic traffic. Our algorithms are based on maximum weight scheduling decisions, and precisely specify 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 algorithms in the presence of reconfiguration overhead.
3) For multiple transceivers per node, we provide a novel method to determine the stability region in this setting.
4) Using delay as a performance metric, we employ simulations to demonstrate the important trade-off 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.

5) 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, labelled $1, 2, \ldots, N$, 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 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 sufficiently many wavelengths to allow any arbitrary logical interconnection of nodes. Each node is equipped with $(N-1)$ virtual output queues (VOQ) in which data are held prior to transmission across the network, with $\text{VOQ}_{i,j}$ containing the backlogged data at node i destined for node j. Time is assumed to be slotted, 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 re-established to form a new logical topology. Transceivers that are retuned 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

$$X(n+1) = X(n) - u(n+1) + a(n+1), \qquad (1)$$

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,

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

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*[2] 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 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

$$X(n+1) = X(0) - U(n+1) + A(n+1). \qquad (3)$$

Throughout this work, 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 logical links exist from source node i to destination node j. The diagonal entries of this matrix have no meaning under our model. 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, each feasible logical topology is represented by a permutation matrix, and $\mathcal{V}$ is the set of permutation matrices of size N.

When we allow multi-hop 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. 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 j-th 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

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

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

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

$n \geq 0$.

A. Synchronization, propagation delay, and distributed implementation

For much of the analysis in this work, it is necessary that the network nodes are synchronized at the slot level. If we restrict the physical topology of the network to be *linear* (*e.g.* a line or ring), then synchronization is easily implemented by making use of a single node as a point of reference. In a slotted ring, when the propagation delay cannot be ignored as negligible, then it is also necessary to ensure that a packet that propagates around the ring will arrive at its source at a slot boundary. For propagation delay t_p seconds, and slot duration t_s seconds, we simply require that t_p be an integer multiple of t_s. In practice however, this is not always possible, but is relatively easily overcome. For example, in SONET rings, adding a small delay at a node in the ring by using IP buffering may be used to effectively satisfy the integer constraint. In the optical domain, fiber delay lines may be employed to achieve the same buffering effect.

Propagation delay arises as an issue in implementing distributed network control. The algorithms of this paper are described as centralized algorithms. However, different levels of propagation delay may be accommodated by different schemes in this work. For example, under non-negligible propagation delay, the *frame-based* algorithms of Section III-D may easily accommodate an additional idleness associated with the distribution of control information prior to reconfiguration decisions, by including this idleness in the reconfiguration overhead time D. Under negligible propagation delay, the more finely time-slotted *bias-based* algorithm of Section III-E is feasible.

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, that when adapted to our model, jointly perform WDM reconfiguration and IP layer routing. These algorithms are based on *maximum weight matchings* and are known to stabilize the system for the special case of zero tuning latency ($D = 0$). 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 $D > 0$.

MWM Maximum weight matching algorithm

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

$$\sum_j v_{i,j}(n) \leq 1, \ \forall i \tag{5}$$

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

$$v_{i,j}(n) \in \{0, 1\}, \ \forall i, j. \tag{7}$$

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

$$u_{i,j}(n+1) = \begin{cases} v_{i,j}(n), & \text{if } X_{i,j}(n) > 0, \\ 0, & \text{if } X_{i,j}(n) = 0. \end{cases} \tag{8}$$

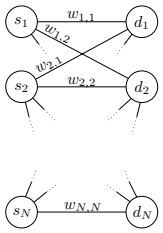


Fig. 3. Weighted complete bipartite graph for maximum weight scheduling.

A. Preliminaries

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

$$\sum_i v_{i,j} \leq 1 \, \forall j, \quad \sum_j v_{i,j} \leq 1 \, \forall i. \tag{4}$$

If the inequalities in (4) are all strict inequalities, then V is called *strictly doubly substochastic*. □

Definition 3.2: The system is *stable* if the backlog process $\{X(n)\}_{n=0}^{\infty}$ satisfies $\limsup_{n\to\infty} E[\sum_{i,j} X_{i,j}(n)] < \infty$. □

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

B. Single-hop algorithm using maximum weight matchings

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 Maximum Weight Matching algorithm, MWM (described above). Essentially, MWM constructs a complete weighted bipartite graph, as in Fig. 3, 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 weight matching on this graph, with the edges of the matching established as logical links over the WDM physical topology. Under MWM, IP layer routing is restricted to single-hop paths, which means

DB Differential backlog algorithm

At time slot $n \geq 0$,

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

$$d_{i,j}(n) = \begin{cases} X_{q(i),j}(n) - X_{h(i),j}(n), & \text{if } h(i) \neq j, \\ X_{q(i),j}(n), & \text{else.} \end{cases} \quad (9)$$

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 (5)-(7). Define the maximum weight 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:

$$E_{i,j}(n) = \begin{cases} 1, & \text{if } \tilde{c}_i(n) = 1,\ j = \hat{j}_i,\ X_{q(i),j}(n) > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

that for each logical link i, only $\text{VOQ}_{q(i),h(i)}$ may be serviced by departing packets along that link.

The power of MWM to stabilize the $N \times N$ crossbar switch is particularly well demonstrated in [9], with the following important stability result, adapted to our reconfigurable queueing network model.

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

Proof: This follows immediately from the proof of [9, 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. Multi-hop 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 weight matchings was shown to be throughput optimal in [8]. We refer to this algorithm as Differential Backlog, or DB (described above).

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 equation (9), $d_{i,j}$ 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. 3 is enlisted in step 2, by setting $w_{i,j} = Z_{i,j}(n)$ for all i, j, and selecting a maximum weight matching. Again, the edges of 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 differential backlog for that link (IP 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 *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 Theorem 3.2]. □

F-P Frame stabilizing algorithm

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 tuning latency.
3) Set $u(l) = u^{\text{P}}(X(kF))$ for $l = kF + D, \ldots, (k+1)F - 1$. Here $u^{\text{P}}(X)$ 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 $\text{VOQ}_{i,j}$ 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

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

packets from the batch and place them in $\text{VOQ}_{i,j}$. The left-over packets remain in the batch for the next frame.

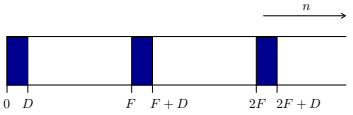


Fig. 4. The regular on-off nature of the frame-based algorithm.

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 $D > 0$.

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. 4. The algorithm operates as follows: at each

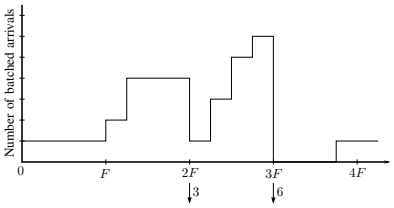


Fig. 5. Illustration of batch size process for a particular VOQ.

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 IP layer routing) $u^{\mathrm{P}}(X)$. Algorithm F-P idles for D slots to allow for tuning latency, 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.

As an example, suppose that $D = 1$ and $F = 4$. Fig. 5 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 3 packets released for service at time $2F$ and 6 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 3 slots are allocated to servicing VOQs within each frame, this ensures that each VOQ backlog changes by an integer multiple of 3 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 class 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

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

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

$$\delta = 1 - \max\{\max_i \textstyle\sum_i \Lambda_{i,j}, \max_j \sum_j \Lambda_{i,j}\}.$$

Corollary 3.1: The frame-based version of MWM, which we refer to as F-MWM, is stable under any arrival process satisfying a 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 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 weight 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 [10] under either F-MWM or F-DB is greater than under the corresponding refined algorithm.

E. Additive bias-based algorithm

In this section, we introduce the *additive bias-based algorithm*, based on MWM, which provides asymptotic throughput

AB Additive bias-based algorithm

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

$$\langle v(n), X(n) \rangle_+ \triangleq b_+ 1_{\{v(n)=v(n-1)\}} + \sum_{i,j} v_{i,j}(n) X_{i,j}(n), \quad (11)$$

subject to the constraints (5)-(7). If $v(n)$ is different from $v(n-1)$ then the network idles for D slots while reconfiguration occurs.

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 class of algorithms is also well suited for scheduling crossbar switches with reconfiguration overhead.

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). The additive bias-based algorithm is given above, and is referred to as AB. Under AB, WDM reconfiguration decisions are made at each time slot, using maximum weight 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. 6 illustrates the intervals associated with service and reconfiguration phases of AB. As opposed to the frame-based scheduling policies, the service intervals are of random duration. We denote by ξ_n the n-th 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 limit* technique. We begin by characterizing the dynamics for the system. Denote by $D_{i,j}(n)$ the cumulative number of departed packets from $\text{VOQ}_{i,j}$ up to time n. 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

$$X_{i,j}(n) = A_{i,j}(n) - D_{i,j}(n), \quad (12)$$

$$D_{i,j}(n) = \sum_{v \in \mathcal{V}} \sum_{l=1}^{n} v_{i,j} 1_{\{X_{i,j}(l)>0\}} \left(Q_v(l) - Q_v(l-1)\right), \quad (13)$$

$$Q_v(\cdot) \text{ is non-decreasing}, \quad (14)$$

$$Q_R(n) + \sum_{v \in \mathcal{V}} Q_v(n) = n. \quad (15)$$

In (12), 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

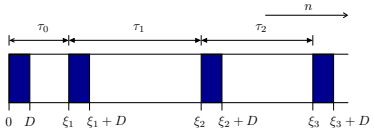


Fig. 6. The service intervals of the additive bias-based algorithm.

$X_{i,j}(t) = X_{i,j}(\lfloor t \rfloor), \forall t \geq 0$, and similarly for A, D, Q_v, and Q_R.

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 (similarly to [11]) that the sequences of scaled variables (indexed by r) converge to the *fluid limits* $x_{i,j}(t)$, $a_{i,j}(t)$, $d_{i,j}(t)$, $q_v(t)$, and $q_R(t)$, almost surely. These fluid limit processes satisfy the following *fluid equations*, for $t \in \mathbb{R}_+$.

$$x_{i,j}(t) = a_{i,j}(t) - d_{i,j}(t), \quad (16)$$

$$a_{i,j}(t) - a_{i,j}(0) = \Lambda_{i,j} t, \quad (17)$$

$$d_{i,j}(0) = 0, \quad (18)$$

$$q_R(t) + \sum_{v \in \mathcal{V}} q_v(t) = t, \quad (19)$$

$$\dot{d}_{i,j}(t) = \sum_{v \in \mathcal{V}} v_{i,j} \dot{q}_v(t), \text{if } x_{i,j}(t) > 0. \quad (20)$$

For the following results, we redefine variable $\delta > 0$ such that

$$1 - \delta > \max\{\max_i \textstyle\sum_j \Lambda_{i,j}, \max_j \sum_i \Lambda_{i,j}\}.$$

This provides a bound analogous to the strict substochasticity required for stability results when $D = 0$.

Lemma 3.1: For any bias-based scheduling algorithm, if the fluid limit process $q_R(t)$ satisfies $\dot{q}_R(t) \leq \delta$ for all $t \geq 0$, then the algorithm stabilizes the network.

Proof: See Appendix A. □

Note that for $D = 0$, Lemma 3.1 immediately implies that the additive bias-based algorithm is stable, since zero time is lost to reconfiguration and thus $q_R(t) = 0$ 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

$$\langle v', X(\xi_n)\rangle \leq \langle v(\xi_n), X(\xi_n)\rangle. \quad (21)$$

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

$$\langle v', X(\xi_n)\rangle + \tau_n N > b_+ + \langle v(\xi_n), X(\xi_n)\rangle - (\tau_n - D)N. \quad (22)$$

Combining (21) and (22), we obtain

$$\tau_n > \frac{b_+}{2N} + \frac{D}{2}. \quad (23)$$

Suppose $b_+/N \geq 2D/\delta - D$. Then, using (23), 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 $\varepsilon > 0$

$$\begin{aligned} Q_R^{(r)}(r(t+\varepsilon)) - Q_R^{(r)}(rt) &< D\left\lceil \frac{r\varepsilon}{D/\delta} \right\rceil, & (24)\\ &\leq r\delta\varepsilon + D. & (25) \end{aligned}$$

Dividing both sides of (25) by r, the right hand side of the inequality can be made arbitrarily close to $\delta\varepsilon$ for sufficiently large integer r. This immediately implies that $\dot{q}_R(t) < \delta$. □

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 in a network, $\mathcal{V}$, 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

$$\sum_i \Lambda_{i,j} \leq P_j \,\forall j, \quad \sum_j \Lambda_{i,j} \leq P_i \,\forall i, \quad (26)$$

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 [12]. However, our proof is a novel natural extension of the well-known Birkhoff-von Neumann decomposition for substochastic matrices (see *e.g.* [13]). □

Given Theorem 3.5, it may be shown that any arrival rate matrix satisfying (26) 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, note that the proof of Theorem 3.3 remains valid under the general port constraint. It can be shown that Theorem 3.4 requires the modification that $b_+/N \geq D(\bar{P}+1)/\delta - D\bar{P}$ for stability to hold, where $\bar{P} = \max_i\{P_i\}$. We will consider the example of an access network in Section IV-D.

IV. Algorithm performance

In this section, we compare the performance of algorithms under different traffic conditions, tuning latencies, and physical topologies. Our simulations demonstrate that there exists a tremendous advantage to employing multi-hop routing at the IP layer under certain conditions. In particular, when there is a single transceiver per node, multi-hop routing is advantageous at low throughput levels. Also, in an access network scenario, where the hub node has N transcievers, multi-hop routing with no WDM reconfiguration is extremely effective when the amount of traffic directed at the hub node becomes large relative to the local inter-node traffic.

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

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

where $X^{\pi}(n)$ is the queue backlog matrix at time n under algorithm π.

In gigabit networks, tuning latencies on the order of $D = 1,000$ to $D = 50,000$ time slots are reasonable values. We only provide data for the case $D = 1,000$, though our tests for larger D values yield identical conclusions.

A. 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:

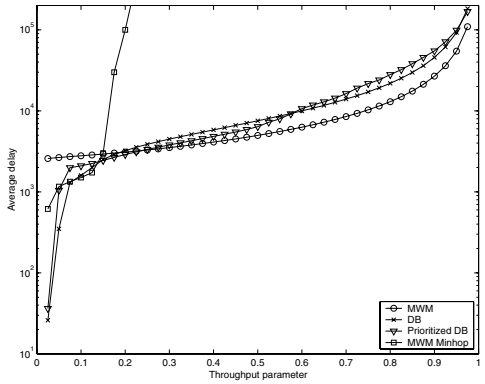


Fig. 7. Average delay for a range of throughput levels.

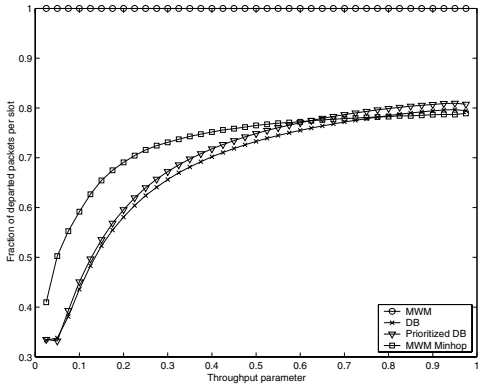


Fig. 8. Fraction of departed packets single-hopped per time slot.

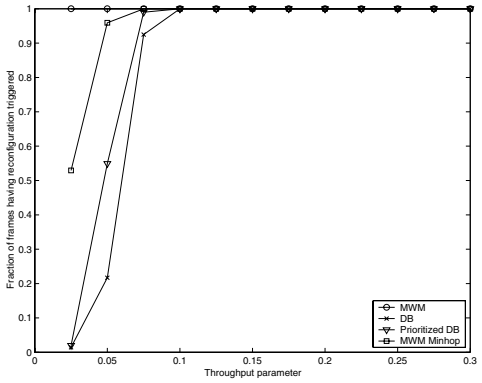


Fig. 9. Fraction of frames in which a reconfiguration was initiated.

1) MWM;
2) DB;
3) Prioritized DB: DB 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 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 VOQ's one hop from departure are serviced with priority.

In general given D, 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.

B. 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 multi-hopping 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 would appear 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 multi-hop 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 Figures 7-9 is the mean value over the 25 sample paths generated for each arrival rate matrix.

Fig. 7 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 trade-off 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. 8 the average fraction of departed packets single-hopped in each time slot, and in Fig. 9 the fraction of frames in which reconfiguration was triggered, for all algorithms. We have truncated the data in Fig. 9 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 multi-hop algorithms, which

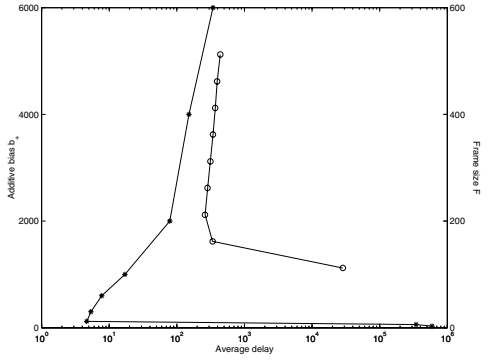


Fig. 10. Frame/bias size versus average simulated delay.

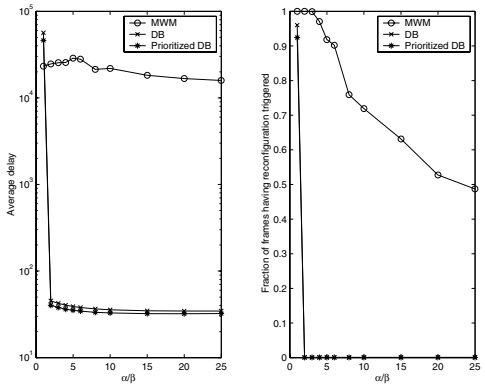


Fig. 11. Average delay (left) and fraction of frames in which a reconfiguration was initiated (right) for a range of α/β values. $N = 6$ nodes, $D = 1000$ time slots. Each non-hub node has an average arrival rate of $\alpha + (N-2)\beta = 0.9$ packets per slot.

implies lower delay associated with reconfiguration overhead. At high throughputs, all algorithms tend to depart more packets through single-hop routes, but the multi-hop 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 algorithms effectively transition between packet switching and circuit switching, and require no knowledge of the traffic arrival process other than the value of δ.

C. Frame vs. bias-based algorithms

The intuitive motivation for introducing additive bias-based algorithms 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. 10 provides simulation results demonstrating the validity of this argument. The simulation scenario has 6 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 appropriately chosen bias size, there is tremendous benefit to using the bias-based algorithm in lieu of the frame-based scheme.

D. Access network

Consider an access network, where $N-1$ of the 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

$$\Lambda_{i,j} = \begin{cases} 0, & \text{if } i = j, \\ \alpha, & \text{if } i = 1 \text{ and } j \neq i, \text{ or if } j = 1 \text{ and } i \neq j, \\ \beta, & \text{else,} \end{cases} \tag{27}$$

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

$$\alpha + (N-2)\beta < 1. \tag{28}$$

Thus, by simply requiring $F > D/\delta$ under the frame-based algorithm, or $b_+/N \geq D(\bar{P}+1)/\delta - D\bar{P}$ under the bias-based algorithm, we may proceed to investigate the performance trade-offs of multi-hop versus single-hop routing for various α, β values.

Fig. 11 plots the data corresponding to the access network under *i.i.d.* Bernoulli arrivals for a range of α/β values. The plot at left of Fig. 11 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 right in Fig. 11. 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. Conclusions

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 random 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 the Differential Backlog (DB) algorithm in general making use of both IP layer multi-hop routes and WDM reconfiguration.

In terms of delay performance, there is a great benefit from employing algorithms that tend to use multi-hop IP layer routes instead of WDM reconfiguration, when the additional load incurred from these multi-hop paths is sufficiently small. At high system loads the opposite is true, and WDM reconfiguration is preferrable to additional load from multi-hop IP layer routing.

An important direction for future research is to gain some traction on *analytically* establishing performance trade-offs between algorithms employing different degrees of reconfiguration/routing. Switching theory has provided bounds on performance of scheduling algorithms (*e.g.* [14]), but much work remains before algorithm performance can be compared under various arrival processes. In terms of scheduling, wide-area networks 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, (11) implies the following additional property of the system dynamics.

$$\langle v, X(n)\rangle < \max_{v'} \left\{\langle v', X(n)\rangle + b_+ 1_{\{v'=v(n-1)\}}\right\} \text{ implies that } Q_v \text{ is not increasing at time } n.$$

The fluid limit version of this property is then given by

$$\langle v, x(t)\rangle < \max_{v'} \{\langle v', x(t)\rangle\} \text{ implies that } q_v \text{ is not increasing at time } t.$$

The remainder of the proof follows closely with the proof of [11, 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 $L(x(t)) > 0$,

$$\begin{aligned}
&\frac{d}{dt}L(x(t)) \\
&= \sum_{i,j} x_{i,j}(t)\left(\Lambda_{i,j} - \dot{d}_{i,j}(t)\right), && (29)\\
&= \sum_{i,j} x_{i,j}(t)\left(\Lambda_{i,j} - \sum_{v\in\mathcal{V}} v_{i,j}\dot{q}_v(t)\right), && (30)\\
&= \sum_{i,j} x_{i,j}(t)\left(\Lambda_{i,j} - v_{i,j}^{\text{dom}}\right) \\
&\quad + \sum_{i,j} x_{i,j}(t)v_{i,j}^{\text{dom}} - (1-\dot{q}_R(t))\max_{v\in\mathcal{V}}\sum_{i,j} x_{i,j}(t)v_{i,j}. && (31)
\end{aligned}$$

Here, (29) and (30) follow from the fluid equations for the system. Setting $\mathcal{V}'$ at time t to be the set of logical topologies v satisfying $\langle v, x(t)\rangle = \max_{v'}\langle v', x(t)\rangle$, we have that $\sum_{v\in\mathcal{V}'} \dot{q}_v(t) + \dot{q}_R(t) = 1$. 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, (31) follows. Setting $\varepsilon = -\min_{i,j}(v_{i,j}^{\text{dom}} - \Lambda_{i,j})$, we have

$$\sum_{i,j} x_{i,j}(t)\left(\Lambda_{i,j} - v_{i,j}^{\text{dom}}\right) \leq -\varepsilon \sum_{i,j} x_{i,j}(t). \quad (32)$$

Also, noting that matrix $v^{\text{dom}}/(1-\delta)$ is a doubly substochastic matrix, and supposing $\dot{q}_R(t) \leq \delta$ for all $t \geq 0$, we have

$$\begin{aligned}
&\sum_{i,j} x_{i,j}(t)v_{i,j}^{\text{dom}} - (1-\dot{q}_R(t))\max_{v\in\mathcal{V}}\sum_{i,j} x_{i,j}(t)v_{i,j}, && (33)\\
&\leq (1-\delta)\left(\sum_{i,j} x_{i,j}\frac{v_{i,j}^{\text{dom}}}{1-\delta} - \max_{v\in\mathcal{V}}\sum_{i,j} x_{i,j}v_{i,j}\right), && (34)\\
&\leq 0. && (35)
\end{aligned}$$

Here, (35) follows by well known properties of the convex doubly substochastic region (for instance, see [15, Lemma 2]).

Combining (31), (32), and (35), we obtain

$$\frac{d}{dt}L(x(t)) \leq -\varepsilon \sum_{i,j} x_{i,j}(t). \quad (36)$$

It can be shown 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 (26). Furthermore, if all inequalities in (26) are satisfied with equality, Λ is called *doubly loaded*, while if all inequalities in (26) are strict, Λ is called *strictly doubly underloaded.* □

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 similarly to [13, Prop. 1]. The following lemma emerges from this result.

Lemma B.1: Given a doubly underloaded matrix Λ, there exists a doubly loaded matrix $\tilde{\Lambda} = (\tilde{\Lambda}_{i,j}, i,j = 1,\ldots,N)$ which dominates Λ pointwise: $\tilde{\Lambda}_{i,j} \geq \Lambda_{i,j}, \forall i,j$. □

B. Bipartite graph from a doubly loaded matrix

Given doubly loaded matrix Ω, we construct a corresponding bipartite graph for which Hall's Theorem guarantees existence of a maximum matching covering all nodes (we call this a saturated matching). Designate the node sets of the two bipartitions by

$$\begin{aligned}
S &= \{s_1^1, s_1^2, \ldots, s_1^{P_1}, s_2^1, \ldots, s_2^{P_2}, \ldots, s_N^1, \ldots, s_N^{P_N}\},\\
D &= \{d_1^1, d_1^2, \ldots, d_1^{P_1}, d_2^1, \ldots, d_2^{P_2}, \ldots, d_N^1, \ldots, d_N^{P_N}\}.
\end{aligned}$$

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

Algorithm B.1: Let $\Phi = \Omega$. 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 $b_{d_j^l} \leftarrow b_{d_j^l} + y_{i,j}$. □

The following lemma follows from the construction of Algorithm B.1, and by Hall's Theorem.

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 one. Once each edge has been considered, matrix v must have i-th row sum P_i and j-th 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 with a saturated matching, the graph may be translated to a corresponding logical topology whose incidence matrix has i-th row sum equal to P_i and j-th 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 [13].

Algorithm B.2: Begin with doubly loaded matrix $\Omega = \tilde{\Lambda}$. Repeat the following steps until $\Omega = 0$. At the n-th step of the algorithm,

1) For matrix Ω, find a saturated logical topology v^n according to Algorithm B.1 and Lemmas B.2-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 saturated, step n of the algorithm reduces the i-th row sum by $\alpha_n P_i$, and the j-th 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 zero, the algorithm terminates in finitely many steps. $\tilde{\Lambda}$ may then be expressed as

$$\tilde{\Lambda} = \sum_{k=1}^{N^2} \left(\alpha_k \prod_{l=1}^{k-1} (1 - \alpha_l) \right) v^k$$

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

REFERENCES

[1] A. Narula-Tam and E. Modiano, "Dynamic load balancing in WDM networks with and without wavelength constraints," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1972–1979, October 2000.

[2] J.-F. P. Labourdette and A. S. Acampora, "Logically rearrangeable multihop lightwave networks," *IEEE Trans. Commun.*, vol. 39, pp. 1223–1230, August 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, pp. 2822–2832, October 1994.

[4] I. Baldine and G. Rouskas, "Traffic adaptive WDM networks: a study of reconfiguration issues," *J. Lightwave Technol.*, vol. 19, pp. 433–455, April 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, 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. Select. Areas Commun.*, vol. 21, pp. 1441–1451, November 2003.

[7] J. G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," in *IEEE Proc. INFOCOM*, 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, December 1992.

[9] A. Stolyar, "Maxweight scheduling in a generalized switch: state space collapse and workload minimization in heavy traffic," *Annals of Applied Probability*, vol. 14, no. 1, pp. 1–53, January 2004.

[10] S. Meyn and R. Tweedie, *Markov Chains and Stochastic Stability*, Springer Verlag, 1996.

[11] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queueing system with asynchronously varying service rates," *Probability in the Engineering and Informational Sciences*, vol. 18, pp. 191–217, 2004.

[12] A. L. Dulmage and N. S. Mendelsohn, "Matrices associated with the hitchcock problem," *Journal of the ACM*, October 1962.

[13] C. Chang, W. Chen, and H. Huang, "Birkhoff-von Neumann input buffered crossbar switches," in *IEEE Proc. INFOCOM*, 2000.

[14] E. Leonardi, M. Mellia, F. Neri, and M. Ajmone Marsan, "Bounds on average delays and queue size averages and variances in input-queued cell-based switches," in *IEEE Proc. INFOCOM*, 2001.

[15] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, pp. 1260–1267, August 1999.