Virtual-Topology Adaptation for WDM Mesh Networks Under Dynamic Traffic

Ayşegül Gençata, *Student Member, IEEE*, and Biswanath Mukherjee, *Member, IEEE*

***Abstract*—We present a new approach to the virtual-topology reconfiguration problem for a wavelength-division-multiplexing-based optical wide-area mesh network under dynamic traffic demand. By utilizing the measured Internet backbone traffic characteristics, we propose an adaptation mechanism to follow the changes in traffic without *a priori* knowledge of the future traffic pattern. Our work differs from most previous studies on this subject which redesign the virtual topology according to an expected (or known) traffic pattern, and then modify the connectivity to reach the target topology. The key idea of our approach is to adapt the underlying optical connectivity by measuring the actual traffic load on lightpaths continuously (periodically based on a measurement period) and reacting promptly to the load imbalances caused by fluctuations on the traffic, by either adding or deleting one or more lightpath at a time. When a load imbalance is encountered, it is corrected either by tearing down a lightpath that is lightly loaded or by setting up a new lightpath when congestion occurs. We introduce high and low watermark parameters on lightpath loads to detect any over- or underutilized lightpath, and to trigger an adaptation step. We formulate an optimization problem which determines whether or not to add or delete lightpaths at the end of a measurement period, one lightpath at a time, as well as which lightpath to add or delete. This optimization problem turns out to be a mixed-integer linear program. Simulation experiments employing the adaptation algorithm on realistic network scenarios reveal interesting effects of the various system parameters (high and low watermarks, length of the measurement period, etc.). Specifically, we find that this method adapts very well to the changes in the offered traffic.**

***Index Terms*—Dynamic traffic, mesh network, mixed-integer linear program (MILP), optical network, virtual-topology reconfiguration, WDM.**

I. Introduction

As TODAY'S networks evolve toward the IP-over-wavelength-division-multiplexing (WDM) networking paradigm, we need to examine new methods to design and efficiently operate them. The primary reason for using optical technology in communication networks is its large bandwidth (50 THz) [1], a capacity that other physical-layer technologies cannot match. This capacity can be "chopped up," e.g., by using WDM technology, into hundreds of channels (wavelengths) which can be operated at the speed of electronic equipment. A WDM network consists of the WDM fiber-optic links and WDM-aware nodes connecting them, possibly in an arbitrary mesh topology. The optical nodes in such a network include optical crossconnects (OXC), transmitters, and receivers, and these nodes are capable of separately routing the various wavelength channels in the optical domain. These optical nodes can be configured to set up lightpaths [2] to provide single "electronic" hop communication channels between any two nodes, which may be geographically far apart in the physical network, and to eliminate extra signal processing at intermediate nodes along that path. (For a detailed survey, refer to [3]).

A virtual topology is defined to be the set of all such lightpaths in a network. Such a virtual topology can be employed by an Internet service provider (ISP) or a large institutional user of bandwidth (hereafter referred to as an ISP) to connect its end equipment (e.g., IP routers) by leasing bandwidth (wavelength channels) from the network operator who owns the fiber plant and OXCs. In fact, multiple virtual topologies, possibly belonging to different ISPs, may coexist on the same fiber plant. However, we will focus on only one such virtual topology and its adaptation in the rest of this paper.

This virtual-topology property is a powerful tool because, with the setup of a lightpath, two nodes become virtually neighbors, regardless of the physical connectivity between them. However, it may not be possible to establish a lightpath for every node pair, because of scalability and economic concerns. Hence, some traffic may need to be switched electronically from one lightpath to another at intermediate nodes until it reaches its destination; this approach is called multihopping [3]. The processes of setting up individual lightpaths are clearly related to one another since a lightpath may carry multihop traffic besides the single-hop traffic between the two nodes it directly connects. For this reason, the virtual-topology design is a combined problem of optimizing the use of network resources, for a given traffic demand. This problem has been addressed by several previous studies [4]–[8]. The different aspects of the problem and a literature survey can be found in [9].

In real networks, however, the traffic rates between node pairs fluctuate distinguishably over time [10], [11], which is an important obstacle in virtual-topology design. (An exemplary traffic measurement can be seen in Fig. 1 [10]. In this example, the measurements for both directions of a link on the Abilene network are displayed as two profiles over a 33-h period, starting at 9:00 a.m. on one day and ending a little after 6:00 p.m. the next day). A virtual topology which is optimized

Manuscript received January 29, 2002; revised June 13, 2002; approved by IEEE/ACM Transactions on Networking Editor G. Rouskas. This work was supported in part by the National Science Foundation under Grant ANI-98-05285 and by Istanbul Technical University under a Grant for Long Term Research Activities. An earlier version of this paper was presented at the IEEE INFOCOM, New York, NY, June 2002.

A. Gençata was with the Department of Computer Science, University of California, Davis, CA 95616 USA. She is now with the Department of Computer Engineering, Istanbul Technical University, Istanbul 80626, Turkey (e-mail: gencata@cs.itu.edu.tr).

B. Mukherjee is with the Department of Computer Science, University of California, Davis, CA 95616 USA (e-mail: mukherje@cs.ucdavis.edu).

Digital Object Identifier 10.1109/TNET.2003.810319

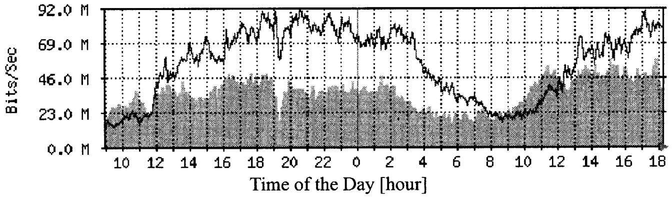

Fig. 1. Traffic measurements on a link in the Abilene network during a 33-h period from 9:00 a.m. on Day 1 to 6:00 p.m. on Day 2. The two profiles correspond to the two directions of traffic on a link.

for a specific traffic demand may not be able to respond with equal efficiency to a different traffic demand. Thus, in such a situation, a reconfiguration of the virtual topology may be needed to match it with the changing traffic.

In this paper, we investigate the problem of on-line reconfiguration of the virtual topology in WDM mesh networks when the traffic load changes dynamically over time. Reconfiguration of optical networks has been studied, both for broadcast optical networks [12], [13], and for wavelength-routed networks [14]–[17]. In these studies, the problem is generally treated as a two-phase operation where the first phase is virtual-topology design for the new traffic conditions and the second phase is the transition operation from the old virtual topology to the newly designed one. A major difficulty with this view is that the future traffic demand is assumed to be known. With this information in hand, the design of a new virtual topology is practicable, and in many studies considerable effort was spent for the design of the next virtual topology so as to reduce the topology changes during the transition phase [15], [17]. In practice, the assumption on the future traffic may decrease the value of the newly designed virtual topology if the traffic changes are predicted inaccurately. What we need is a reconfiguration method that can update the virtual topology without relying on traffic forecasts.

Another problem originates from the transition process. During the second phase, lightpaths involved in transition cannot be used by the ongoing traffic. Most earlier studies developed techniques to minimize the disruption to the ongoing traffic [12], [18], [19]. These studies dealt with this problem either by performing the reconfiguration on all network elements concurrently [18], or by applying step-by-step changes until the new virtual topology is settled upon [12], [19]. In both cases, the traffic disruption because of the transition phase cannot be eliminated completely.

In another earlier study, hitless reconfiguration was defined as a reconfiguration process without the loss of any data [14]. According to the methodology proposed in [14], the transition between topologies was achieved by first establishing all new links without removing any link. The links of the old topology were removed only when the traffic was rerouted through the links of the new topology.

In this study, our problem definition is different. As traffic fluctuates over time, it will be monitored systematically, and the virtual topology will be changed accordingly, but we do not make any other assumptions on future traffic pattern. The reconfiguration process is seen as a continuous measurement-adaptation system where small adjustments are made, instead of waiting for a noticeable drop in system efficiency and changing the entire topology. With this new definition of the problem, we expect to solve the issue of unpredictability of the traffic and to obtain a scheme which is practical for real backbone networks.

The traffic-disruption problem due to the unavailability of network resources can also be solved by our new approach if the step adjustments on the virtual topology are chosen carefully. In our study, we design the step adjustments in a hitless manner, so that a change can be either a lightpath addition or a lightpath deletion (after rerouting the traffic using this virtual link).

Observations on several backbone networks show that the amount of traffic between nodes changes in a smooth and continuous manner [10], [11] (see Fig. 1). Long-term variations have time-of-the-day characteristics where traffic intensities change in terms of hours. Therefore, the adaptation mechanism we propose would be beneficial especially for backbone networks of ISPs since the traffic characteristics in these environments favor the slow adaptation.

In this paper, we develop a reconfiguration algorithm which applies simple adjustments to the virtual topology when it is necessary. Network traffic is measured periodically, and the information on the virtual-link loads (i.e., lightpath loads) is used to decide whether or not an adjustment is needed. Basically, a new lightpath is added when a congestion is encountered, and a lightpath is deleted if it is being underutilized. Thus, an ISP can optimize the operational cost of its virtual topology by leasing only the appropriate amount of lightpaths as it considers to be really necessary. We introduce two system parameters to detect the link-usage (in)efficiencies: high watermark W_H and low watermark W_L. At the end of an observation period (typically hundreds of seconds), if the load of one or more lightpaths is higher than W_H, a new lightpath is established to decrease that load. When the load of a link drops below W_L, that link will be torn down if alternate paths exist for diverting the traffic using that lightpath.

Our scheme is computationally simple when compared with previous studies, e.g., [19], and hitless since the traffic is not directly interrupted. The system parameters W_H and W_L provide flexibility and control over the adaptation process, specifically the frequency of the virtual-topology adjustments, average hop distance, and network resource usage. We elaborate on the effects of the system parameters through our simulation experiments reported in Section V.

In previous studies on virtual-topology design, the methods established as many lightpaths as possible and reconfiguration did not change the number of lightpaths [12], [19]. Our method, on the contrary, keeps only the necessary number of lightpaths that can grow during peak traffic hours and fall when the overall network traffic decreases. This type of operation is more cost effective for ISPs. Also, we show that the adaptation method can perform reconfiguration using far fewer lightpath changes compared with an earlier reconfiguration study [15].

The rest of the paper is organized as follows. Section II explains the network and the traffic models and provides an informal statement of the problem. Selection of the best candidate lightpath to add or to drop at each step is formulated as an optimization problem, and solutions obtained by solving this formulation for a simple network are used to compare our method to an earlier study in Section III. The key ideas of our adaptation algorithm are introduced, and an outline is given in Section IV. In Section V, representative numerical examples employing our adaptation algorithms are discussed. Section VI concludes the paper.

II. Problem Definition

A. Network Model

We consider a network of N nodes connected by bidirectional optical links forming an arbitrary physical topology. Each optical link supports W wavelengths, and any node i is assumed to have T_i transmitters and R_i receivers. We assume that each node is equipped with an OXC with full wavelength-conversion capability, so that a lightpath can be established between any node pair if the resources (an optical transmitter at source, an optical receiver at destination, and at least a wavelength on each fiber link) are available along the path. Mechanisms to accommodate no wavelength conversion and different numbers of wavelengths on different links are straightforward. We consider unidirectional lightpaths, since the traffic between two nodes is not necessarily symmetric (as can be seen in Fig. 1).

Each OXC is connected to an edge device, e.g., an IP router, which can be a source or a destination of a traffic flow and which can provide routing for multihop traffic passing by that node. We assume that each router is capable of processing all packet traffic flowing through it and of observing the amount of traffic on its outgoing lightpaths. In this paper, for ease of explanation, we consider a centralized approach to the virtual-topology reconfiguration problem. A central manager will collect the virtual-link usage information from routers at the end of every observation period. Specifically, the link-usage information needed to make a reconfiguration decision consists of which links are overloaded, which links are underloaded, and what are the end-to-end packet-traffic intensities flowing through the overloaded links. The decision for a topology change will then be made by the central manager, and a signaling mechanism will be started if a lightpath addition or deletion is required as a result of the decision algorithm (in this paper, for simplicity, we ignore the details of the signaling protocol). An implicit assumption here is that the observation period is much longer (typically hundreds of seconds or longer) than the time it takes for control signals to propagate from various nodes to the central manager. We expect that it is possible to design a decentralized protocol to do this job as well, but this is outside the scope of our present investigation.

In the optical layer, we use shortest path routing for routing lightpaths on the physical topology and the first-fit scheme for wavelength assignment [20]. For packet routing, we consider a shortest path (minimum-hop) routing scheme, since it provides better usage of network links and is frequently used by existing routing protocols.

B. Problem Statement

Essentially, our aim is to provide a very good virtual topology under dynamic traffic conditions, by keeping the lightpath loads balanced, and by changing the virtual connectivity only when it is necessary. The adaptation process should be quick enough to match the long-term traffic fluctuations, but should not disturb the traffic unnecessarily. We state the problem we want to solve as follows.

Given:

- A network graph $P(\mathcal{V}, E_P)$ where $\mathcal{V}$ is the set of nodes and E_P is the set of links connecting the nodes. Graph nodes correspond to network nodes with OXCs and links correspond to the fibers between nodes.
- Number of wavelength channels carried by each fiber.
- Number of transmitters and receivers at each node.
- Current virtual topology $V(\mathcal{V}, E_V)$ as another graph where the nodes correspond to the nodes in the physical topology. Each link in E_V corresponds to a direct optical lightpath between the nodes.
- Current traffic load carried by each lightpath.

Determine:

- Whether the current virtual topology of the network is efficient for the current traffic.
- Whether a change in the virtual topology should be made.
- If a change is necessary, which lightpaths should be added and/or deleted.

One can identify the following important steps in solving such a problem:

- Traffic should be monitored continuously to provide adequate information to the reconfiguration system.
- A decision mechanism is needed to trigger a virtual-topology change if the current topology is not convenient.
- Finally, the exact modification to the topology should be determined.

III. Local Optimization

We state our approach to solve this problem formally as a mixed-integer linear program (MILP). This formulation is defined for one step of the adaptation, so it defines a local optimization.

A. Formulation of Adaptation as a MILP

We assume full wavelength-conversion capability at each node in this formulation. We use the following notations:

- s and d denote *source* and *destination* of a traffic flow when used as a superscript or subscript.
- i and j denote *originating* and *terminating* nodes of a lightpath, respectively.
- m and n denote the end points of a physical link.

At any step of the adaptation, one of these three decisions can be made: addition of a lightpath, deletion of a lightpath, or no change to the virtual topology. The choice of the decision is related to the highest and the lowest lightpath loads and the watermark values. Selecting the proper action and the best lightpath which keeps the maximum link load as low as possible is a local optimization problem ("local" with respect to time). This problem turns out to be a MILP. In the remainder of this section, we give the MILP formulation for one step of the adaptation method.

Given:

- Number of nodes in the network $= N$.
- Physical topology of the network $P = \{P_{mn}\}$, where P_{mn} indicates the number of fibers between nodes m and n, and $P_{mn} = P_{nm}$ for $m = 1, 2, \ldots, N$ and $n = 1, 2, \ldots, N$.
- Current traffic matrix $\Lambda = \{\Lambda_{sd}\}$ denotes the average traffic rate (in bits/s) measured during the last observation period between every node pair, with $\Lambda_{ss} = 0$ for $s = 1, 2, \ldots, N$.
- Current virtual topology $V = \{V_{ij,q}\}$, where $V_{ij,q}$ is a binary value denoting the qth lightpath between nodes i and j, and $V_{ii,q} = 0$. $V_{ij,0} = 0$ if there is no lightpath from node i to node j. $V_{ij,0} = V_{ij,1} = \cdots = V_{ij,k-1} = 1$ and $V_{ij,k} = 0$ if there are k lightpaths from node i to node j. Since lightpaths are not necessarily assumed to be bidirectional, $V_{ij,q} = 0 \nRightarrow V_{ji,q} = 0$.
- Number of wavelengths on each fiber $= W$.
- Capacity of each wavelength channel $= C$ bits/s.
- Number of transmitters and receivers at node i: T_i and R_i respectively.
- High watermark value $= W_H$, where $W_H \epsilon (0, 1)$, e.g., $W_H = 0.8$ implies that a lightpath is considered to be overloaded when its load exceeds $0.8 * C$.
- Low watermark value $= W_L$, where $W_L \epsilon (0, 1)$.
- Highest and lowest lightpath loads measured during the observation period: L^P_{Max} bits/s and L^P_{Min} bits/s, respectively.

Variables:

- Physical routing binary variable $p^{ij,q}_{mn} = 1$ if the qth lightpath from node i to node j is routed through the physical link (m, n).
- New virtual topology: $V' = \{V'_{ij,q}\}$, where $V'_{ij,q}$ is defined similar to $V_{ij,q}$. Note that V' is at most one lightpath different from V.
- Traffic routing: The binary variable $\Upsilon^{sd}_{ij,q}$ is 1 when the traffic flowing from node s to node d traverses lightpath $V'_{ij,q}$, and 0 otherwise. $\Upsilon^{sd}_{ii,q} = 0$ by definition. The traffic from s to d is not bifurcated, i.e., all traffic between s and d will flow through the same path.
- Load of maximally loaded lightpath in the network: L_{Max}.

Objective:

$$\text{Minimize} \quad L_{\text{Max}}.$$

The objective function minimizes the load of the maximally loaded lightpath in the network. This objective allows us to balance the network load in the new virtual topology, by addition or deletion of the best possible lightpath.

Constraints:

- On physical topology:

$$\forall\, i, j, q,\ \sum_n p^{ij,q}_{in} = V'_{ij,q} \tag{1}$$

$$\forall\, i, j, q,\ \sum_n p^{ij,q}_{nj} = V'_{ij,q} \tag{2}$$

$$\forall\, i, j, q, l,\ \sum_n p^{ij,q}_{nl} - \sum_n p^{ij,q}_{ln} = 0, \quad i \neq l \text{ and } j \neq l \tag{3}$$

$$\forall\, m, n,\ \sum_i \sum_j \sum_q p^{ij,q}_{mn} \leq W \cdot P_{mn} \tag{4}$$

$$\forall\, m, n, i, j, q,\ p^{ij,q}_{mn} \leq V'_{ij,q}. \tag{5}$$

Equation (1) ensures that only one outgoing physical link of the source node will be assigned to a lightpath. Equation (2) ensures that only one incoming physical link at the destination node will be assigned to a lightpath. Equation (3) guarantees that the number of incoming and outgoing links reserved for a lightpath at any intermediate node will be equal. The total number of wavelengths used between two nodes is limited to (the number of fiber links) $*W$ by (4). Note that we assume wavelength conversion capability on network nodes, and we use the wavelength channels on different fibers as nondistinguishable entities. To capture wavelength continuity in a nonwavelength-conversion network, some additional constraints will be needed. Equation (5) states that a physical link is assigned only if the lightpath exists.

- On virtual-topology connections:

$$\sum_i \sum_j \sum_q V'_{ij,q} = \sum_i \sum_j \sum_q V_{ij,q} + k_H - (1 - k_H) \cdot k_L \tag{6}$$

where

$$k_H = \left\lceil \frac{L^P_{\text{Max}}}{C} - W_H \right\rceil \text{ and } k_L = \left\lceil W_L - \frac{L^P_{\text{Min}}}{C} \right\rceil,$$

$$\forall\, i, j, q, [1 + 2 \cdot (k_H - 1) \cdot k_L] \cdot (V'_{ij,q} - V_{ij,q}) \geq 0. \tag{7}$$

Note that the values of k_H and k_L are binary and they are calculated by using the maximum and the minimum lightpath loads measured in the last observation period, watermark values, and channel capacity. Therefore, these values are constant for the MILP. Note that k_H is unity when one or more lightpaths are experiencing heavy load. This will ensure that a new lightpath will be added to the virtual topology. k_L is unity when one or more lightpaths has a load below the low watermark. In this case if $k_H = 0$ (i.e., none of the lightpaths in the virtual topology is heavily loaded), a lightpath will be deleted. Thus, we are giving a higher priority to a lightpath addition than to a lightpath deletion to better accommodate the traffic. Equation (6) specifies the total number of lightpaths in the new virtual topology, i.e., it decides whether a change should be made, and if the answer is affirmative, whether a lightpath should be added or deleted. Equation (7) guarantees that the new virtual topology consists of the same set of lightpaths of the old virtual topology except that one lightpath is added or deleted.

- On virtual-topology traffic variables:

$$\forall\, s, d, l, q\ \sum_i \Upsilon^{sd}_{il,q} - \sum_i \Upsilon^{sd}_{li,q} = \begin{cases} 1, & l = d \\ 0, & l \neq s \text{ and } l \neq d \\ -1, & l = s \end{cases} \tag{8}$$

$$\forall\, s, d, i, j, q,\ \Upsilon^{sd}_{ij,q} \leq V'_{ij,q} \tag{9}$$

$$L_{\text{Max}} \le C \cdot W_H \tag{10}$$

$$\forall\, i,j,q, \quad \sum_s \sum_d \Lambda_{sd} \cdot \Upsilon_{ij,q}^{sd} \le L_{\text{Max}}. \tag{11}$$

Equation (8) is a multicommodity-flow equation controlling the routing of packet traffic on virtual links. Equation (9) ensures that traffic can only flow through an existing lightpath, and (10) specifies the capacity constraint for any lightpath. Equation (11) constrains the load on any lightpath to be lower than or equal to the maximum load L_{Max}.

- On transceivers:

$$\forall\, i, \sum_j \sum_q V'_{ij,q} \le T_i \tag{12}$$

$$\forall\, j, \sum_i \sum_q V'_{ij,q} \le R_j. \tag{13}$$

Equations (12) and (13) limit the total number of lightpaths originating from and terminated at a node to the total number of transmitters and receivers at that node.

The above formulation gives the best selection for a virtual-topology adjustment of one lightpath. We will now, by solving the MILP, compare our adaptation scheme with an earlier reconfiguration method proposed in [15].

B. Adaptation With Minimal Lightpath Change

In this section, we focus on minimizing the number of lightpaths (i.e., resource usage cost) and the number of changes made for reconfiguring the topology (i.e., operation cost). We show that the new adaptation approach is cost effective in both sense by comparing our results with an earlier reconfiguration study [15]. The comparison is based on solving MILP formulations of both methods by using the standard solver CPLEX [21].

The reconfiguration method proposed in [15] (which we call full reconfiguration) can be summarized as follows.

```
Start with initial virtual topology.
Every  Δ seconds do:
 Find the optimal virtual topology for the new
 traffic pattern.
 Find the virtual topology such that:
  It requires minimum number of changes from
  the previous topology.
```

We solve the MILP given in [15] by substituting the objective function with

$$\text{Minimize} \quad \sum_i \sum_j \sum_q V_{ij,q}$$

to minimize the total number of lightpaths in the network. We also limit the maximum load L_{Max} to W_H. When the optimum virtual topology is obtained from this formulation, this solution gives the minimum number of lightpaths that any virtual topology must contain to carry the given traffic demand. Let this number be η. This MILP can then be modified as follows.

1) The objective function guarantees that the new virtual topology will be as close as possible to the previous one.
2) A new constraint is added to the formulation to guarantee that the new virtual topology will have η lightpaths.

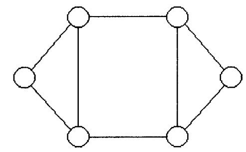

Fig. 2. Six-node network used in experiments.

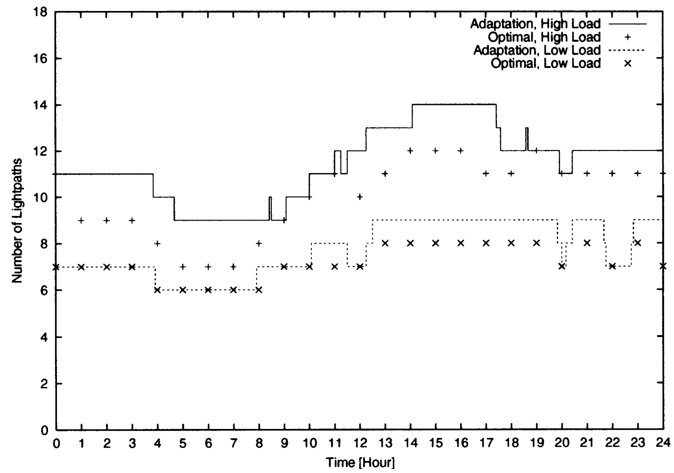

Fig. 3. Number of lightpaths in virtual topologies obtained by solving our proposed adaptation MILP and the optimal reconfiguration MILP.

This second step selects the closest virtual topology among the feasible topologies having exactly η lightpaths and able to carry the given traffic demand. We use two metrics to compare our adaptation method and the full reconfiguration method: the total number of lightpaths in the network and the number of lightpath additions and deletions.

Since finding the optimal topology using the method in [15] is computationally tractable only for small networks, our comparison is based on a six-node topology shown in Fig. 2. (See Section V-A for details of the traffic model. Larger network examples employing our approaches will be shown in Section V). Comparison of the number of lightpaths as a function of time used by our proposed approach and the approach in [15] is shown in Fig. 3 for a 24-h run. Four wavelengths and four transceiver pairs per node are assumed for both cases, and two set of results obtained for two different traffic matrices (one with 33% of the node pairs having nonzero traffic intensities, referred to as *low load*, and the other with 50% node pairs with nonzero traffic, referred to as *high load*). The results of our adaptation scheme are obtained with an observation period of 5 min and are shown as lines. The optimal topology is calculated every hour and is shown as points in Fig. 3. This plot shows that, as traffic load changes over time, the number of lightpaths in the virtual topology for both approaches changes in unison as well. (Compare the profiles in Fig. 1 with Fig. 3, noting that the measurements in Fig. 1 starts at 9:00 a.m.). However, as also expected, the adaptation method establishes a few more lightpaths than those in the optimal topology, and the difference in the number of lightpaths increases slightly with the traffic load.

Fig. 4 plots the cumulative number of lightpath additions and deletions for both methods during the same 24-h period (results from only the high-traffic-load experiment are shown in this figure). We observe that the adaptation scheme can reconfigure

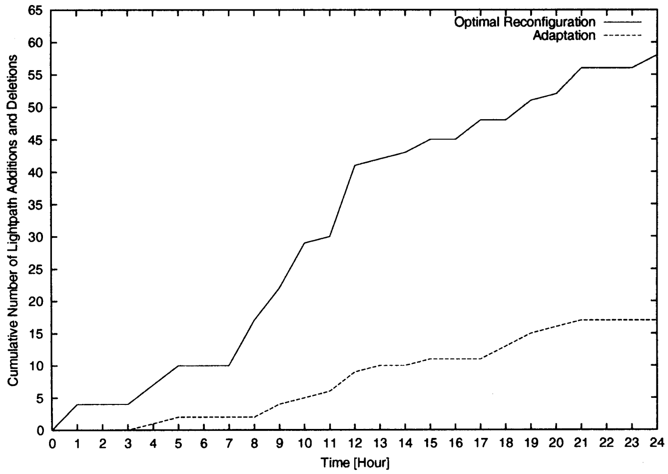

Fig. 4. Comparison of the cumulative number of lightpath changes (sum of lightpath additions and deletions) between our adaptation scheme and optimal reconfiguration.

the topology using far fewer lightpath changes compared with the optimal reconfiguration; thus, the adaptation scheme is a very efficient one.

We remark that the complexity of the MILP limits its use for large networks. Thus, in Section IV we design an efficient heuristic method to adapt the virtual topology in response to traffic changes in large networks.

IV. Heuristic Adaptation Algorithm

A. Key Ideas

We propose a one-phase reconfiguration algorithm where the network traffic flows are continuously observed and simple updates to the virtual topology are made whenever necessary. The key idea of our approach is to set up new lightpaths when congestion occurs or tear down existing lightpaths when they are not used efficiently. Since frequent changes to the virtual topology are not desirable, the length of traffic-observation period and the instant to trigger a virtual-topology change (either a lightpath addition or a lightpath deletion) should be selected carefully.

The length of the observation period is a parameter which can be selected and updated according to the specific properties of the network traffic. It specifies how frequently the adaptation algorithm is activated. We use link loads of the lightpaths in the current topology as input to the triggering function of the adaptation algorithm. Since having perfectly balanced loads on every link may not be practical or may not be possible, we aim to maintain quasi-balanced virtual links. We can define a quasi-balanced topology as follows:

$$\begin{aligned} V =& \{V_1, V_2, \ldots, V_k\} \\ W_L \leq& \delta(V_i) \leq W_H, \qquad i = 1, 2, \ldots, k \end{aligned}$$

where k is the number of lightpaths in the virtual topology whose lightpaths are denoted by V, $\delta(V_i)$ is the load of the ith lightpath, and W_L and W_H are the low watermark and the high watermark values, respectively. The objective of the adaptation algorithm is to keep the load of every lightpath between these two thresholds. At the end of an observation period, if the load of one or more lightpaths is higher than W_H, the lightpath $V_{\max}$ having the maximum load $\delta_{\max} = \delta(V_{\max})$ will be considered, and a new lightpath will be established to decrease $\delta_{\max}$. The new lightpath V_{new} should carry at least one of the traffic flows that $V_{\max}$ was carrying, so that, after the new lightpath is established, the load of $V_{\max}$ can decrease. As an extreme case, one may set $W_H = 100$ and $W_L = 0$, minimizing the changes on the virtual topology but also decreasing the quick reaction ability of the network to dynamic changes in the traffic. The characteristic of the traffic should determine the proper choice for the watermark values.

There are several ways to select the source and the destination nodes of the new lightpath to be added. We choose to establish a lightpath between the end nodes of the multihop traffic with the highest load using lightpath $V_{\max}$. By doing so, we expect to have a noticeable decrease in $\delta_{\max}$. If $V_{\max}$ is only carrying single-hop traffic, then we simply add a new lightpath between the end nodes of $V_{\max}$.

When the load of a virtual link decreases to a value below W_L, either because of changes in the dynamic traffic or because of traffic rerouting due to previous topology changes, the algorithm will tear down that link, unless it is a part of the only path for a traffic flow. The main idea behind the deletion of underloaded lightpaths is to release the network resources whenever they are not used efficiently and to have them available when new lightpaths are to be established later. In some cases, it is possible for the deletion of a lightpath to lead to congestion on another lightpath and, consequently, to an addition of a new lightpath. However, the traffic rates are changing dynamically, and in such an environment this behavior should not be interpreted as instability. This dynamism is necessary for the method to find a very good set of lightpaths for the current traffic. Whenever required, this ping-pong effect can be controlled by adjusting the watermark values.

Whenever a lightpath is added or deleted, the routes for all traffic flows are recalculated. Occasionally, new overloaded or underloaded lightpaths may occur after rerouting due to change of the routes of some traffic, but this is a part of the adaptation process and indicates that it may be beneficial to add or delete more lightpaths. So, a few more steps may be helpful for the virtual topology to reach a load-balanced state.

We should also note that variations to our approach exist, such as deletion or addition of multiple lightpaths in one step when the load of more than one lightpath is outside the region defined by watermarks. In this paper, we study the basic case where only one lightpath is allowed to be added or deleted, and we also study a variation of the basic case where unlimited number of additions or deletions are allowed at every observation period.

B. Outline of the Algorithm

We give a pseudocode of the heuristic adaptation algorithm in Fig. 5. We use λ_{sd} to denote the traffic rate from node s to node d normalized to the capacity of a virtual link. The notation (i, j) is used to express the originating and the terminating nodes of a lightpath, and G to represent the virtual-topology graph. $G - V_{ij}$ is a subgraph of G with one edge from node i to node j extracted.

The algorithm first checks for any lost end-to-end traffic because of a lack of connectivity in the virtual topology. Under normal operation, the adaptation process expects a connected

Input:

- Virtual topology, $V = \{V_{ij}\}$
- Current traffic rates, λ_{sd}
- Load of virtual links, δ_{ij}
- Available number of transmitters and receivers at node i, T_i and R_i

Output:

A decision which can be one of the following:

- Addition of a lightpath between nodes i_{new} and j_{new}
- Deletion of an existing lightpath between i_d and j_d
- No change to the virtual topology

Algorithm:

At the end of every observation period:

if $\lambda_{sd} > 0$ and there is no path from s to d then
  Establish V_{sd} for $\max_{sd}\{\lambda_{sd}\}$
else
  Find the maximum link load: $\delta_{max} = \delta_{i_{max},j_{max}} = \max_{i,j}(\delta_{ij})$
  if $\delta_{max} > W_H$ then
    max_load $\leftarrow 0$
    for every multi-hop traffic flow λ_{sd} using $V_{i_{max},j_{max}}$ do
      if ($\lambda_{sd} >$ max_load) and ($T_s > 0$) and ($R_d > 0$) then
        max_load $\leftarrow \lambda_{sd}$
        $s_{mm} \leftarrow s, \quad d_{mm} \leftarrow d$
      endif
    endfor
    if max_load > 0 then
      if $V_{s_{mm}d_{mm}}$ can be established then
        Establish $V_{s_{mm}d_{mm}}$
        $T_{s_{mm}} \leftarrow T_{s_{mm}} - 1, \quad R_{d_{mm}} \leftarrow R_{d_{mm}} - 1$
      endif
    else
      if $V_{i_{max}j_{max}}$ can be established then
        Establish $V_{i_{max}j_{max}}$
        $T_{i_{max}} \leftarrow T_{i_{max}} - 1, \quad R_{j_{max}} \leftarrow R_{j_{max}} - 1$
      endif
    endif
  endif
  if no lightpath is added then
    Find the minimum link load: $\delta_{min} = \delta_{i_{min},j_{min}} = \min_{i,j}(\delta_{ij})$
    complete $\leftarrow$ false
    while ($\delta_{min} < W_L$) and not complete do
      if (i_{min}, j_{min} are connected in $G - V_{i_{min}j_{min}}$)
        or ($\delta_{min} = 0$)
      then
        Tear down $V_{i_{min}j_{min}}$
        $T_{i_{min}} \leftarrow T_{i_{min}} + 1, \quad R_{j_{min}} \leftarrow R_{j_{min}} + 1$
        complete $\leftarrow$ true
      else
        Find the next lowest link load: $\delta_{nextlow}$
        $\delta_{min} \leftarrow \delta_{nextlow}$
        $(i_{min}, j_{min}) \leftarrow (i_{nextlow}, j_{nextlow})$
      endif
    endwhile
  endif
endif

Fig. 5. Pseudocode of heuristic adaptation algorithm.

network of already established lightpaths and it preserves the connectivity of node pairs as long as these nodes have a nonzero rate of traffic flowing between them. In some cases, traffic rate from a node s to a node d may cease, and if this silent period is longer than an observation period, a lightpath providing the connectivity may be torn down.

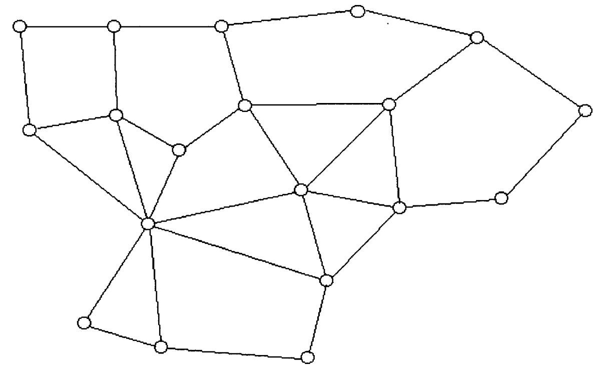

Fig. 6. Example telco network used as the physical topology in our study.

If all of the communicating node pairs have at least one route, then the lightpath loads are considered next. Information on ligthpath loads is collected from routers which are source of a lightpath, by the central manager (which is running the adaptation algorithm), at the end of each observation period (see Section II-A). A change to the virtual topology is made only when a link load is higher than W_H or lower than W_L; otherwise, no change is made until the end of the next observation period. The maximum link load is considered first, since a highly loaded link (and with a potentially increasing load) would cause more serious problems compared with an underloaded link. Lightpath deletion is considered only if no lightpath is added, because we allow only one lightpath change at a time (i.e., in one observation period). Lightpath deletion is performed only if that link is not on the unique path between a node pair i and j for which the traffic rate $\lambda_{ij} > 0$. All lightpaths with a load lower than W_L are considered in increasing order of their loads, until an appropriate lightpath is found.

The second variation that we implement in this study adds or deletes lightpaths until all loads are between the watermarks. In an observation period, only additions or deletions are allowed exclusively, but the number of changes is not limited. Although this method requires more processing, it shows a very interesting aspect of the adaptation method, namely, even though multiple lightpath additions or deletions are allowed, our numerical examples reveal that, in the vast majority of adaptations, just a single lightpath addition or deletion suffices to bring the virtual topology to the quasi-balanced state.

V. Illustrative Numerical Examples

A. Simulation Environment

Simulation experiments were conducted to investigate the fitness of the adaptation approach and to expose the effect of various system parameters. In our experiments, we implemented the heuristic algorithm shown in Fig. 5.

A telco mesh network of 19 nodes interconnected by 31 bidirectional links was used as the physical topology for the numerical examples shown in this section for illustration purposes (Fig. 6). For our examples, we assume that the number of wavelengths is 16 per fiber link and equal for all links. Each node is assumed to have eight transmitters and eight receivers.

Since this paper is focused on backbone networks, a traffic model is derived based on the observations on several backbones' link loads. Backbone traffic is the aggregation of several end systems' traffic, and the aggregation process filters out the short-term variations. On the other hand, as can be seen in Fig. 1, long-term variations (on a scale of hours) remain and repeat their pattern in one-day periods. To obtain a realistic model, we sampled some representative link traffic rates (Fig. 1 shows one of them) from real networks over a 24-h period. The sampling allows us to represent the traffic rate as a function of time. The average traffic rate at a point in time between two sampling points of the rate function was calculated by a linear interpolation to represent the continuous change in traffic. These traffic-rate functions are used to generate the simulated traffic between any node pair in the network. We expect that the effect of aggregation and smoothing is more intense at the edge routers with respect to the smoothing effect of aggregation of traffic from a few node pairs on a network link. Therefore, we believe that it is reasonable to use the link traffic to simulate the traffic between the edge routers for our purposes. This assumption is not vital for the adaptation method we propose, and the method should work for any input traffic having long-term fluctuations, since the reaction time can be adjusted according the time scale of these fluctuations. To create a simulation pattern for each node pair, a traffic rate function is randomly selected among five different patterns. The chosen rate function is then scaled by multiplying it by a random value. The random scaling allows us to create differences between the traffic rates of different node pairs, even when the same pattern is chosen for different node pairs. Each random scaling factor α_{sd} (for a node pair $\langle s, d \rangle$) takes values in the range $[0.2, 1.2]$ in our examples to create appropriate variations in the traffic volumes between different node pairs. In this model, every element of the traffic matrix is a continuous function of time representing the traffic rates during one day, rather than a single value. We should also note that no assumption on future traffic demand is being made in the adaptation scheme except that (without loss of generality) the traffic rates are expected to fluctuate slowly compared with the time interval between adaptation steps. Using this model, the traffic matrix was randomly generated to have nonzero rate functions for 60% of the node pairs (values on the main diagonal remain zero).

In this paper, we employ the following performance metrics to reveal the different aspects of the adaptation scheme: the average time between consecutive reconfiguration steps, the traffic-weighted average hop distance, the number of lightpaths in the virtual topology, and the percentage of links with loads in the balanced region $[W_L, W_H]$. Through these metrics, we examine the effects of changing the system parameters, specifically high watermark, low watermark, and length of observation period.

B. Typical Operation

As a primary experiment, we demonstrate the operation of the system by measuring the maximal and minimal lightpath loads in the network at the end of every observation period. High and low watermarks will define a *balance region* in which the lightpath loads are allowed to fluctuate as long as they do not exceed the high watermark or drop below the low watermark. Every time a link load goes out of the balance region, the system will try to adapt to the new traffic conditions. In most of our

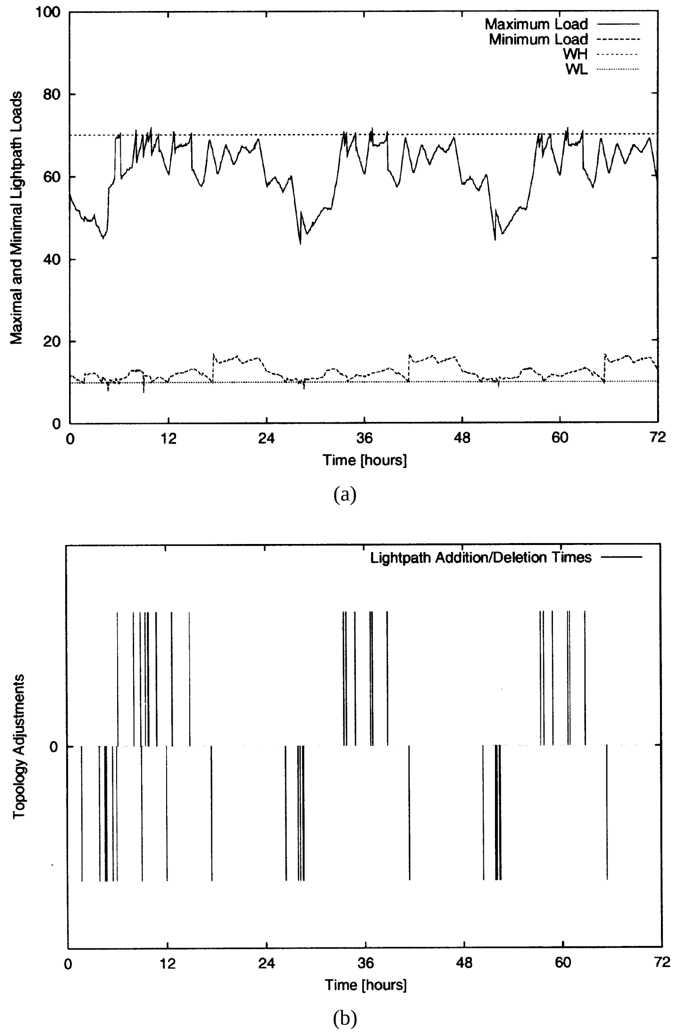

Fig. 7. (a) Maximal and minimal lightpath loads in the network during a three-day run ($W_H = 70, W_L = 10$). (b) Impulse graphic indicating times of lightpath addition or deletion.

examples that follow, we will allow a maximum of only one lightpath addition or deletion; finally, in Section V-G, we will examine the system performance when an unlimited number of lightpath additions and deletions are allowed.

An exemplary operation of the system for a three-day period with $W_H = 70$ and $W_L = 10$ is shown in Fig. 7. Note that, in this section we represent the high and low watermark values as $0 \leq W_H, W_L \leq 100$, normalized to 100, so that W_H and W_L can be interpreted as percentage loads. Fig. 7(a) plots the maximal and minimal loads in the network observed at the end of each observation period (300 s for this experiment). Fig. 7(b) shows the times of topology adjustments, where a positive impulse indicates a lightpath addition and a negative impulse indicates a lightpath deletion. The same average traffic rate functions were repeated for every day, but slight differences can be seen for the same time of different days related to randomly generated traffic and to the dynamic nature of the algorithm.

C. Effect of Watermarks

The effect of changing the value of W_H on traffic-weighted average hop distance for a fixed value of $W_L = 10$ is shown in

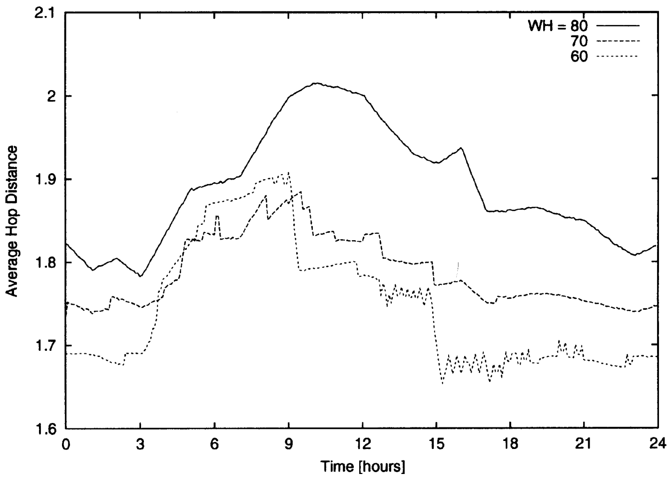

Fig. 8. Traffic-weighted average hop distances during a day, for different values of W_H with $W_L = 10$.

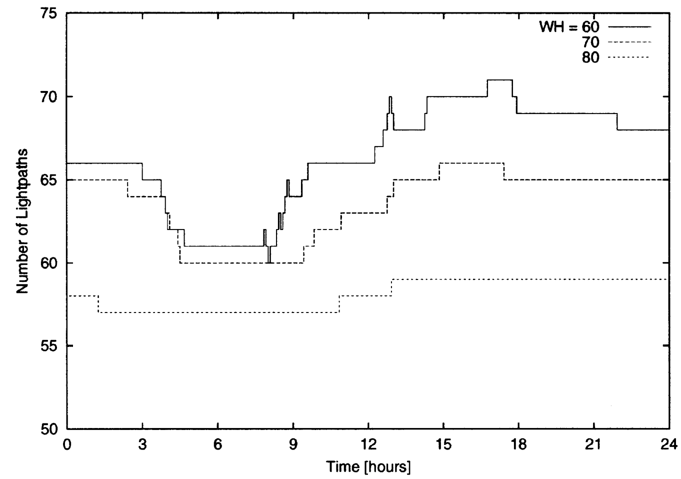

Fig. 10. Number of lightpaths during a day, for different values of W_H with $W_L = 10$.

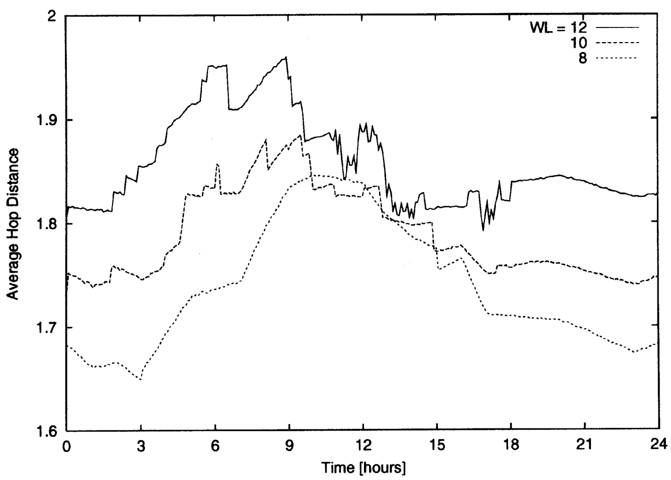

Fig. 9. Traffic-weighted average hop distances during a day, for different values of W_L with $W_H = 70$.

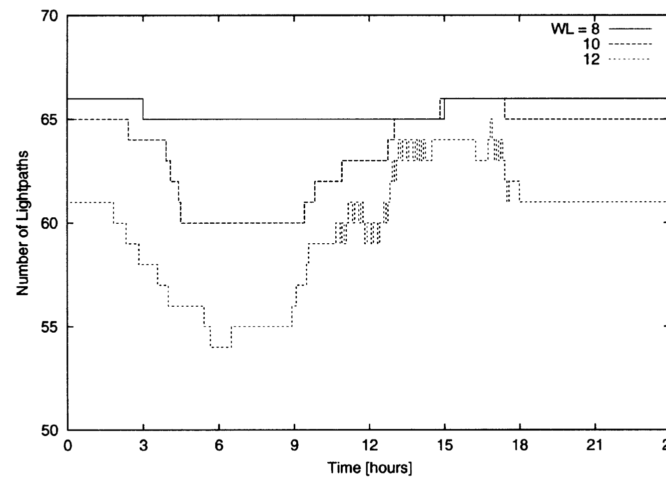

Fig. 11. Number of lightpaths during a day, for different values of W_L with $W_H = 70$.

Fig. 8. The hours on the x axis correspond to the real hours of the day. During the night (until 9 or 10 a.m.), as the traffic load decreases all over the network, the lightpath loads decrease as well and the adaptation scheme deletes the lightly loaded links, causing an increase in average hop distance. During the daytime, the average hop distance decreases with the increasing traffic load. As can be seen in the figure, the average hop distance decreases when W_H takes smaller values, since the lightpaths are allowed to carry less load and the W_H limit can be reached more quickly, resulting in new lightpath additions. For the case where $W_H = 60$, more lightpaths are deleted compared with the case where $W_H = 70$ during nighttime. This effect manifests itself in Fig. 8 as an aggressive climb in the average hop distance, and is related to the tightness of the watermarks: As the values of the watermarks approach each other, more lightpaths have loads closer to the bounds, and small changes in traffic may result in topology changes.

Fig. 9 plots the effect of W_L on the average hop distance for a fixed value of W_H. When W_L is small, lightpaths are deleted less frequently since fewer lightpaths will have their load drop below a small W_L, compared with the case with a higher W_L value. Therefore, the virtual topology keeps more lightpaths alive and has a lower hop-distance value.

The total number of lightpaths in the virtual topology during a one-day period is plotted in Fig. 10 for different values of W_H with $W_L = 10$. Higher values of W_H result in a smaller number of lightpaths and more economic usage of network resources such as transceivers. Consistent with the results obtained for average hop distance, the number of lightpaths is smaller when the network load decreases due to time-of-the-day effect.

Fig. 11 plots the number of lightpaths for different values of W_L with $W_H = 70$. The number of lightpaths in the system decreases when W_L increases because the adaptation scheme tends to delete lightpaths more aggressively. The total number of lightpaths is smaller when W_L takes higher values. Since a larger number of lightpaths requires a larger amount of network resources, it may be preferable to keep this number small for economic use of resources. The two system parameters W_H and W_L can be used to balance the use of resources and the average hop distance.

Figs. 12 and 13 plot the average time between two consecutive adjustments in the virtual topology as a function of W_H

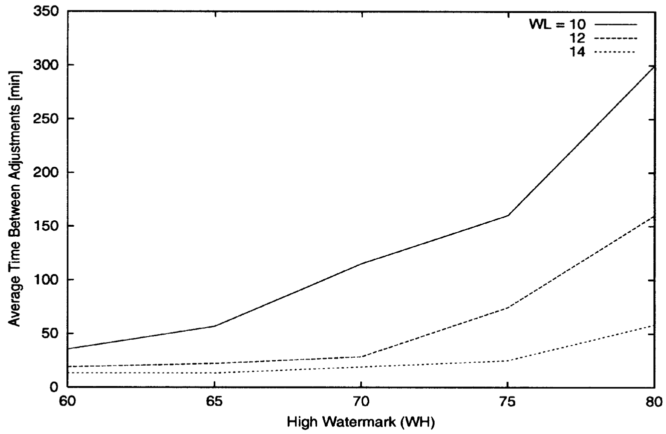

Fig. 12. Average time between two consecutive reconfiguration steps, as a function of W_H.

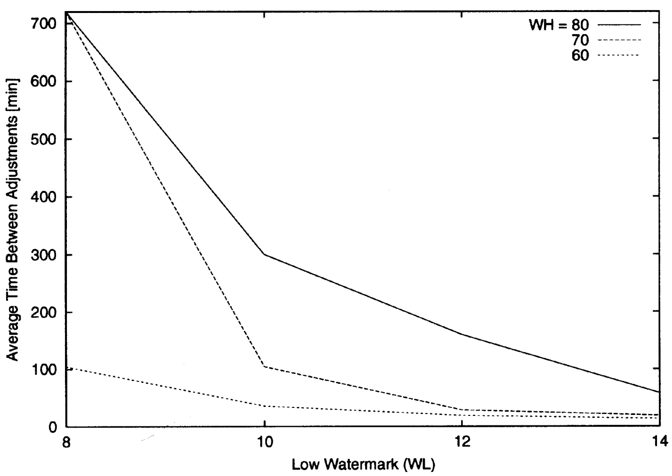

Fig. 13. Average time between two consecutive reconfiguration steps, as a function of W_L.

and W_L, respectively. We observe in Fig. 12 that the virtual topology is adjusted less frequently when W_H takes higher values. Similarly, when W_L takes higher values, the virtual topology is adjusted more frequently as can be seen in Fig. 13. These two figures show that the average number of lightpath additions and deletions can be controlled by appropriately choosing the watermarks.

D. Effect of Observation Period

A third parameter that affects the system's behavior is the length of observation period over which the lightpath loads are measured. The average traffic rate on a lightpath is calculated by measuring the total amount of traffic passing over it and dividing that amount by the length of the period. The lightpath load is the ratio of the measured average traffic rate to the lightpath capacity. Since the traffic rate is taken as an average value during a period, the length of one period will affect the system's ability to track the changes in traffic. Momentary changes could be measured and reacted to by using enough short observation periods, but a smoothing effect on traffic measurements by using longer periods would be more accurate to avoid the instabilities and frequent adjustments. On the other hand, long periods would decrease the efficiency of step-by-step adaptation on following the traffic changes closely if we do not allow more than one lightpath to be changed at the end of each period. Table I shows the effect of the observation-period length on topology-adjustment decisions. In this experiment, $W_H = 70$ and $W_L = 10$, and the number of adjustments in a day is averaged over a five-day run.

TABLE I
PROBABILITY OF ADJUSTMENT DECISION AT THE END OF AN OBSERVATION PERIOD. $W_H = 70, W_L = 10$

Length of Observation Period [s]	Total Number of Periods in One Day	Average Number of Adjustments	Adjustment Probability
100	864	12.33	0.014
200	432	12.17	0.028
400	216	14	0.065
800	108	17.6	0.163

The first column shows the length of observation period in seconds, used during the experiments. The values in the second column are the total number of observation periods in a day; therefore, they also show how many times the adaptation algorithm was activated. The average number of adjustments during one day (number of additions and number of deletions are almost equal) is shown in the third column. The probability of adjustment at the end of an observation period is the ratio of the value in the third column to the one in the second column. As the observation period gets longer, it is more likely that the algorithm will decide to make an adjustment in the virtual topology. This result confirms our claim above, namely, short observation periods keep track of the traffic more accurately.

E. Effect of Window Size

The basic algorithm in Fig. 5 may sometimes react very quickly because a change is made based on the measurement during the last observation period. If sudden jumps occur in traffic load, keeping a longer history information may be more useful to prevent the false alarms in the network from triggering a change. To study the effect of such history information, we store the lightpath loads measured at each period during a history window. The basic algorithm in Fig. 5 is the case where the window size is equal to one. At each new period, the window is slided forward by one unit to include the new measurement and to exclude the oldest one. The decision on lightpath addition/deletion is based on the average load calculated using the entire window. In Fig. 14, we show three plots of maximal lightpath loads interlaced for window sizes, 1, 2, and 4 ($W_H = 70$ and $W_L = 10$). The graphics are shifted vertically for a clear view and a 70% horizontal line is shown for each plot for guidance. As the window size increases, the observed average loads become smoother and some changes are delayed or skipped. When the window size is one, a lightpath is added around 6 a.m., but this addition does not take place for larger window sizes. However, increasing the window size would also delay some necessary changes. Specifically, when many lightpath loads increase at the same time, this delay may cause problems.

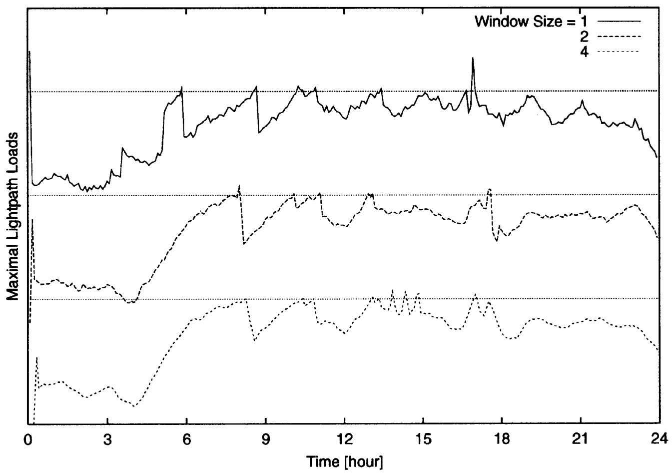

Fig. 14. Maximal lightpath loads during a day for different history window sizes. The plots are shifted for better view.

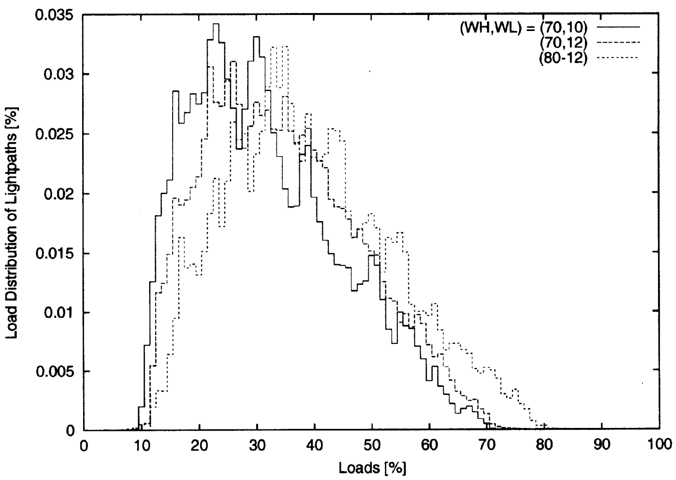

Fig. 15. Distribution of lightpath loads for different values of watermarks over the course of the simulated five days.

F. Load Balancing

The adaptation mechanism works to balance the lightpath loads all over the network to maintain all loads in the balance region. In this section, we show the load-balancing ability of the system. Fig. 15 plots the distribution of lightpaths according to their loads, for three different watermark-value pairs. At the end of each observation period (300 s for this example), the loads of all lightpaths in the network are measured and the occurrence of each load is added to the total occurrences of that load since the beginning of the simulation. This process is repeated at the end of every observation period throughout the simulation. A point in this graphic indicates the percentage of the lightpaths with that load in the course of the experiment (a five-day run). The figure shows that a high percentage of the links are in the balance region. Another result is that most of the lightpaths are gathered in a smaller region toward the middle of the balance region, showing that few lightpaths are critically close to the watermarks, and they can trigger a topology adjustment in the following observation period. The same experiment is repeated for different values of W_H and W_L, and the percentage of the lightpaths in the balance region is tabulated in Table II. As W_H and W_L values are relaxed, the percentage of lightpaths in the balance region

TABLE II
PERCENTAGE OF THE LIGHTPATH LOADS BETWEEN THE WATERMARKS OVER THE COURSE OF SIMULATION

	W_L			
W_H	8	10	12	14
60	99.9976	99.9958	99.9932	99.9882
65	99.9995	99.9986	99.9964	99.9919
70	99.9999	99.9991	99.9971	99.9949
75	99.9999	99.9991	99.9987	99.9962
80	99.9999	99.9997	99.9994	99.9984

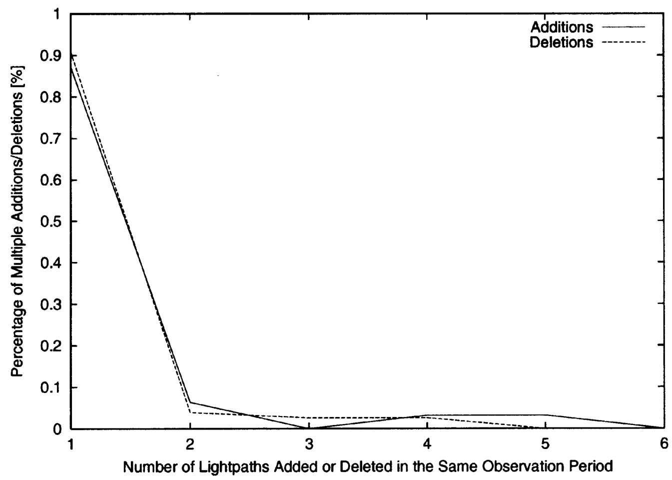

Fig. 16. Distribution of number of lightpath additions and deletions when the number of additions and deletions are not limited.

increases. This table shows that the watermarks can be adjusted to obtain a more or less aggressive topology adaptation.

G. Unlimited Additions and Deletions

To show how many lightpath additions or deletions are necessary at every observation period to balance the lightpath loads, we relaxed our algorithm to allow an unlimited number of additions or deletions at each step. When a lightpath is added or deleted, the traffic is redistributed and the lightpath loads are recalculated. If there is still an imbalance in loads, an additional change is made immediately, without waiting for the next observation period. Fig. 16 plots the distribution of the number of additions and deletions in one observation period (with $W_H = 70$ and $W_L = 10$). The percentage is calculated over the total number of periods where at least one change is made. For example, the percentage of observation periods where only one addition was made is 87% of all observation periods where at least one addition was made. This figure shows that one change in a period is sufficient to obtain a good virtual topology in the majority of adaptation cases. Also note that, a virtual-topology change is seldom, i.e., in most of the observation periods no change is made to the virtual topology as can also be observed in Fig. 7. The percentage of such *silent* observation periods is 94% for the specific numerical example in Fig. 16.

VI. CONCLUSION

Based on a new view of the virtual-topology reconfiguration problem, we presented an adaptation scheme for WDM mesh networks under dynamic traffic. We defined the problem as tracking the long-term traffic fluctuations by adapting the topology in a measurement-adaptation cycle with the following constraints: no assumption should be made on future traffic rates and the ongoing traffic should not be interrupted by a transition phase. Our work is motivated by the traffic characteristics of backbone networks, which change slowly and according to the time of the day.

We developed an algorithm which adjusts the virtual topology by adding or deleting one lightpath at the end of a measurement cycle, if it is necessary. We introduced two system parameters, called high watermark W_H and low watermark W_L, on lightpath loads to detect any over- or underutilized lightpath. We showed through simulations that these two parameters can be effectively used to regulate the operation of the proposed topology-adaptation method. We also presented a MILP formulation for the selection of the lightpath to be added or deleted to minimize the maximum link load in the network for an adaptation algorithm which allows one lightpath change at a time. By solving this MILP, we showed that the performance of the adaptation scheme is comparable to the optimal reconfiguration [15] in terms of number of lightpaths and much better in terms of cumulative number of changes. We implemented a variation of the adaptation method where an unlimited number of changes were allowed at every step, and our results showed that one change in one observation period was enough to maintain the loads in balance most of the time.

We believe that this work evolves the virtual-topology reconfiguration problem from a perspective where it is considered as an interrupted process to a new one where it is taken as an evolutionary process. A future direction is to investigate how high and low watermarks can be changed dynamically to satisfy specific performance demands. The algorithm we introduced can be enhanced in different directions. Application of new heuristics to our scheme to select the lightpath to be added or deleted may be possible and the performance of different heuristics can be evaluated using the MILP formulation presented in this paper. The development of a distributed version of the algorithm is also an open problem for further study.

References

[1] B. Mukherjee, *Optical Communication Networks*. New York: McGraw-Hill, 1997.

[2] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high bandwidth optical WANs," *IEEE Trans. Commun.*, vol. 40, pp. 1171–1182, July 1992.

[3] B. Mukherjee, "WDM optical communication networks: Progress and challenges," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1810–1824, Oct 2000.

[4] B. Mukherjee, D. Banerjee, S. Ramamurthy, and A. Mukherjee, "Some principles for designing a wide-area WDM optical network," *IEEE/ACM Trans. Networking*, vol. 4, pp. 684–696, Oct 1996.

[5] R. Ramaswami and K. N. Sivarajan, "Design of logical topologies for wavelength-routed optical networks," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 840–851, June 1996.

[6] R. M. Krishnaswamy and K. N. Sivarajan, "Design of logical topologies: A linear formulation for wavelength-routed optical networks with no wavelength changers," *IEEE/ACM Trans. Networking*, vol. 9, pp. 186–198, Apr. 2001.

[7] R. Dutta and G. N. Rouskas, "A survey of virtual topology design algorithms for wavelength routed optical networks," *Opt. Networks Mag.*, vol. 1, no. 1, pp. 73–89, Jan. 2000.

[8] J.-F. P. Labourdette and A. S. Acampora, "Logically rearrangeable multihop lightwave networks," *IEEE Trans. Commun.*, vol. 39, pp. 1223–1230, Aug. 1991.

[9] R. Dutta and G. N. Rouskas, "Optical WDM networks: Principles and practice," in *Design of Logical Topologies for Wavelength Routed Networks*. Norwell, MA: Kluwer, 2000, pp. 79–102.

[10] Abilene Network Traffic Statistics. [Online]. Available: http://www.abilene.iu.edu/

[11] Cooperative Association for Internet Data Analysis. [Online]. Available: http://www.caida.org

[12] J.-F. P. Labourdette, G. W. Hart, and A. S. Acampora, "Branch-exchange sequences for reconfiguration of lightwave networks," *IEEE Trans. Commun.*, vol. 42, pp. 2822–2832, Oct. 1994.

[13] I. Baldine and G. N. Rouskas, "Traffic adaptive WDM networks: A study of reconfiguration issues," *J. Lightwave Technol.*, vol. 19, pp. 433–455, Apr. 2001.

[14] K. Bala *et al.*, "Toward hitless reconfiguration in WDM optical networks for ATM transport," in *Proc. IEEE GLOBECOM*, 1996, pp. 316–320.

[15] D. Banerjee and B. Mukherjee, "Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study," *IEEE/ACM Trans. Networking*, vol. 8, pp. 598–607, Oct. 2000.

[16] B. Ramamurthy and A. Ramakrishnan, "Virtual topology reconfiguration of wavelength routed optical WDM networks," in *Proc. IEEE GLOBECOM*, 2000, pp. 1269–1275.

[17] N. Sreenath, C. S. R. Murthy, B. H. Gurucharan, and G. Mohan, "A two-stage approach for virtual topology reconfiguration of WDM optical networks," *Opt. Networks Mag.*, pp. 58–71, May 2001.

[18] G. N. Rouskas and M. H. Ammar, "Dynamic reconfiguration in multihop WDM networks," *J. High Speed Networks*, vol. 4, no. 3, pp. 221–238, 1995.

[19] A. Narula-Tam and E. Modiano, "Dynamic load balancing for WDM based packet networks," in *Proc. IEEE INFOCOM*, 2000, pp. 1010–1019.

[20] H. Zang, J. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Opt. Networks Mag.*, vol. 1, no. 1, pp. 47–60, Jan. 2000.

[21] CPLEX. [Online]. Available: http://www.ilog.com

Ayşegül Gençata (S'91) received the B.Sc. and M.Sc. degrees in control and computer engineering from Istanbul Technical University, Istanbul, Turkey, in 1993 and 1995, respectively. She is working toward the Ph.D. degree in computer engineering at the same university.

She is a Research Assistant with Istanbul Technical University. She was a Visiting Research Scholar at the Optical Networks Laboratory, University of California, Davis, from December 2000 to June 2002. Her research interests include architecture and protocol designissues in optical WDM networks.

Biswanath Mukherjee (S'82–M'87) received the B.Tech. (Hons) degree from the Indian Institute of Technology, Kharagpur, India, in 1980 and the Ph.D. degree from the University of Washington, Seattle, in 1987.

In 1987, he joined the University of California, Davis, where he has been a Professor of computer science since 1995. He served as Chairman of the Department of Computer Science from 1997 to 2000. He is a Member of the Board of Directors of IPLocks Inc., a Silicon Valley startup company. He has consulted for and served on the Technical Advisory Board of a number of startup companies in optical networking. He has served on the editorial boards of the *ACM/Baltzer Wireless Information Networks (WINET)*, the *Journal of High-Speed Networks*, *Photonic Network Communications*, and *Optical Network Magazine*. He is the author of the textbook *Optical Communication Networks* (New York: McGraw-Hill, 1997), a book which received the Association of American Publishers, Inc. 1997 Honorable Mention in Computer Science. His research interests include lightwave networks, network security, and wireless networks.

Dr. Mukherjee is a corecipient of paper awards presented at the 1991 and 1994 National Computer Security Conferences. He has served on the editorial boards of the IEEE/ACM TRANSACTIONS ON NETWORKING and *IEEE Network*. He has also served as Editor-at-Large for optical networking and communications for the IEEE Communications Society, and as the Technical Program Chair of the IEEE INFOCOM 1996 conference. While at the University of Washington, he held a GTE Teaching Fellowship and a General Electric Foundation Fellowship.