Design of Logical Topologies for Wavelength-Routed Optical Networks

Rajiv Ramaswami, *Senior Member, IEEE*, and Kumar N. Sivarajan, *Member, IEEE*

***Abstract*—This paper studies the problem of designing a *logical topology* over a wavelength-routed all-optical network (AON) *physical topology*. The physical topology consists of the nodes and fiber links in the network. On an AON physical topology, we can set up lightpaths between pairs of nodes, where a lightpath represents a direct optical connection without any intermediate electronics. The set of lightpaths along with the nodes constitutes the logical topology. For a given network physical topology and traffic pattern (relative traffic distribution among the source–destination pairs), our objective is to design the logical topology and the routing algorithm on that topology so as to minimize the network congestion while constraining the average delay seen by a source–destination pair and the amount of processing required at the nodes (degree of the logical topology). We will see that ignoring the delay constraints can result in fairly convoluted logical topologies with very long delays. On the other hand, in all our examples, imposing it results in a minimal increase in congestion. While the number of wavelengths required to imbed the resulting logical topology on the physical all-optical topology is also a constraint in general, we find that in many cases of interest this number can be quite small.**

We formulate the combined logical topology design and routing problem described above (ignoring the constraint on the number of available wavelengths) as a mixed integer linear programming problem which we then solve for a number of cases of a six-node network. Since this programming problem is computationally intractable for larger networks, we split it into two subproblems: logical topology design, which is computationally hard and will probably require heuristic algorithms, and routing, which can be solved by a linear program. We then compare the performance of several heuristic topology design algorithms (that do take wavelength assignment constraints into account) against that of randomly generated topologies, as well as lower bounds derived in the paper.

I. Introduction

All-OPTICAL wavelength-division multiplexed (WDM) networks [1] using *wavelength routing* are considered to be potential candidates for the next generation of wide-area backbone networks. A WDM all-optical network (AON) can use the large bandwidth available in optical fiber to realize many channels, each at a different optical wavelength, and each of these channels can be operated at moderate bit rates (1–2.4 Gb/s). Networks using 20–100 wavelengths will be feasible in the next few years. The *physical topology* of the network consists of optical wavelength routers interconnected by pairs of point-to-point fiber links in an arbitrary mesh topology as shown in Fig. 1. Each pair of links is represented by an undirected edge between routing nodes in this figure. End-nodes are attached to the routers. Each end node has a limited number of optical transmitters and receivers. Each link is capable of carrying a certain number of wavelengths. A routing node, shown in Fig. 2, takes in a signal at a given wavelength at one of its inputs and routes it to a particular output, independent of the other wavelengths. A router with Δ_p inputs and Δ_p outputs capable of handling Λ wavelengths can be thought of as Λ independent $\Delta_p \times \Delta_p$ reconfigurable switches (preceded and followed by wavelength (demux) and (mux) elements respectively).

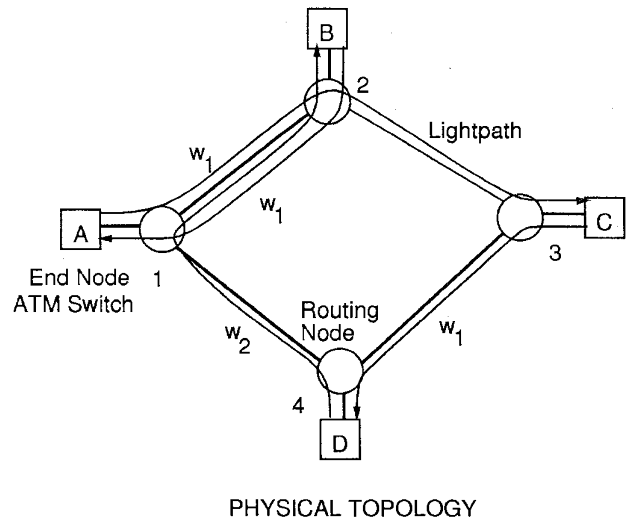


Fig. 1. A WDM network consisting of routing nodes interconnected by pairs of point-to-point fiber-optic links. The routing nodes have end-nodes attached to them that form the sources and destinations for network traffic.

A. Physical and Logical Topologies

The physical topology of the network is the physical set of routing/end-nodes and the fiber-optic links connecting them upon which one sets up *lightpaths* between end nodes. A lightpath consists of a path through the network between end nodes and a wavelength on that path. Lightpaths are set up by configuring the routing nodes in the network. Two lightpaths that share a link in the network must use different

Manuscript received March 31, 1995; revised September 25, 1995. This work was supported in part by ARPA under Grants MDA 972-92-C-0075 and MDA 972-95-C-0001. This work was presented in part at INFOCOM'95, April 4–6, 1995.

R. Ramaswami is with the IBM T.J. Watson Research Center, Yorktowne Heights, NY 10598 USA.

K. N. Sivarajan is with the Indian Institute of Science, Bangalore, India and also with the IBM T.J. Watson Research Center, Yorktowne Heights, NY 10598 USA.

Publisher Item Identifier S 0733-8716(96)03670-0.

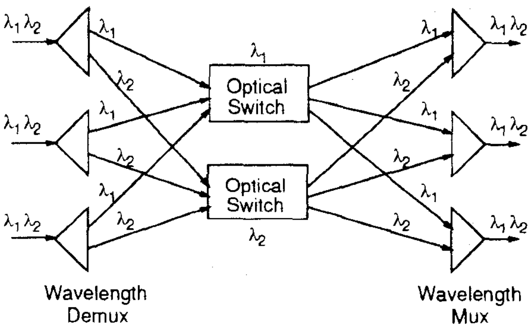


Fig. 2. Structure of a reconfigurable router. The router can switch each wavelength at its input ports independent of the other wavelengths.

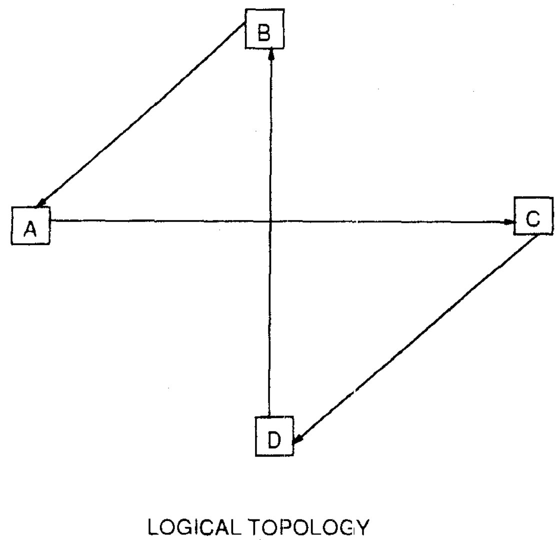


Fig. 3. A possible logical topology for the WDM network of Fig. 1. The directed edges in this topology represent lightpaths between the corresponding end-nodes in Fig. 1.

wavelengths.[1] A lightpath provides a pipe between end nodes with a bandwidth equal to that of one channel, typically 1–2.4 Gb/s. The set of all lightpaths that have been set up between end nodes constitutes the *logical topology*. An example logical topology for the physical topology of Fig. 1 is shown in Fig. 3. This logical topology corresponds to the set of lightpaths shown in Fig. 1. The logical topology is a graph with the nodes corresponding to the end-nodes in the network with a directed edge from node A to node B if a lightpath has been set up from node A to node B. The *physical degree* of a (routing) node is the number of other (routing) nodes that it is directly connected to by fiber-optic links. For example, the physical degree of all the (routing) nodes in Fig. 1 is two. The *logical out-degree* of an end-node is the number of lightpaths that originate from that end-node and the *logical in-degree* of an end-node is the number of lightpaths that terminate in that end-node. For example, in Fig. 3, the logical in-degree and out-degree of every end-node is one. We assume that each routing node is associated with a single end node and vice versa and thus we will simply speak of the physical and logical degree of a node.

Ideally in a network with N nodes, we would like to set up lightpaths between all the $N(N-1)$ pairs. However this is usually not possible because of two reasons. First, the number of wavelengths available imposes a limit on how many lightpaths can be set up. (This is also a function of the traffic distribution.) For example, in [2], it is shown that averaged over a number of randomly chosen 128-node networks with average physical degree four, one can set up an average of 640 full-duplex lightpaths using 32 wavelengths, or only about 10 full-duplex lightpaths per node (much less than the 127 required to have lightpaths to all other nodes). Secondly, each node can be the source and sink of only a limited number of lightpaths, say Δ_1. This is determined by the amount of optical hardware that can be provided (transmitters and receivers) and by the amount of information the node can handle in total.

When it is not possible to establish lightpaths between all pairs of nodes, node pairs that are not directly connected via lightpaths must use a sequence of lightpaths through intermediate nodes to communicate. At each intermediate node, packets coming in on a lightpath must be converted to electronic form, switched electronically and then converted back to optical form and sent out on a different lightpath enroute to their destinations. In other words, packets may have to take *multihop* paths to reach their destination. In addition to the constraints above, each node can electronically switch only a limited amount of information, determined by the number of ports the electronic switch at that node can handle, say Δ_2. These constraints impose a restriction on the maximum degree of the logical topology: the degree (or the number of lightpaths originating or terminating in a node) must be at most $\Delta_l = \min(\Delta_1, \Delta_2)$.

Note that even if the number of available wavelengths were sufficiently large so that lightpaths could be established between all $N(N-1)$ possible source–destination (s-d) pairs, if $\Delta_1 < N-1$, we have to solve the problem of coordinating the use of the lightpaths among the s-d pairs so that collisions and contentions are avoided, which is hard to do in the wide-area environment. If this were the model, all packets could be routed directly on all-optical paths and no forwarding of packets at intermediate nodes would be required. Instead we assume that we set up a logical topology based on the traffic matrix and that those s-d pairs with no direct lightpath must use multihop paths, which avoids the coordination problem altogether.

B. Logical Topology Design

This paper studies the problem of designing such a logical topology and routing on the designed logical topology. We assume that the physical topology is already given and that a traffic matrix representing long-term average flows between end nodes is also given. It is reasonable to minimize both the network *congestion* (defined below) and the average packet *delay*. The average delay consists of a component due to

[1] We assume in this paper that this restriction only applies to the links between routing nodes, i.e., there may be more than one lightpath to/from the same end-node on the same wavelength provided they are routed on different links to/from the attached routing node. A routing/end-node architecture that is capable of implementing this is described in [2].

the queueing delays at the intermediate nodes and the link propagation delays. In high-speed wide-area networks, the propagation delay dominates over the queueing delay as long as link utilizations are not too close to the link capacity. We neglect queueing delays in the rest of this paper.

As we increase the degree of a node or the number of wavelengths in the network, the congestion and delay will decrease. For a given degree and number of wavelengths, if we seek to minimize any metric that involves only propagation delay and ignores the congestion, the best solution is as follows (if the given requirement on the degree of the logical topology is at least equal to the maximum degree of the physical topology): Make the logical topology the same as the physical topology. Then use shortest path routing on the logical topology with the link metric being the propagation delay on the link.

We prefer to formulate the problem in terms of minimizing congestion subject to the restriction that the delay for a source–destination pair be no more than some multiple of the minimum possible delay. This formulation is appropriate when we are given the relative (average) traffic distribution in the network or the "traffic pattern" (but not the actual or absolute value of the (average) traffic) and must maximize the total (average) traffic that the network can support. This is in contrast to other work that tries to minimize average delay but ignores the congestion [3], [4].

Let $T = (\lambda^{sd})$ be the traffic matrix, i.e., λ^{sd} is the arrival rate of packets at s that are destined for d. We seek to create a logical topology G_l and a routing on G_l that minimizes $\lambda_{\max} = \max_{ij} \lambda_{ij}$ where λ_{ij} denotes the offered load on link (i, j) of the logical topology. $\lambda_{\max}$ is the maximum offered load to a logical link and is called the *congestion*. Let G_p be the given physical topology of the network, Δ_l the degree of the logical topology, and W the number of wavelengths available. An informal description of the logical topology design problem is as follows (a precise definition as a mixed-integer linear program (MILP) is given in Section III):

$$\min \lambda_{\max}$$

such that:

- each logical link in G_l corresponds to a lightpath and two lightpaths that share an edge in the physical topology are assigned different wavelengths;
- the total number of wavelengths used is at most W;
- every node in G_l has Δ_l incoming edges and Δ_l outgoing edges;
- traffic is routed so that the traffic flows between each source–destination pair are conserved at each node; and
- for each source–destination pair, the propagation delay is at most α times the worst-case (shortest-path) propagation delay in G_p.

Note that the topology design problem includes routing as a subproblem.

C. Previous Work

The logical topology optimization problem has been studied earlier for the case where the physical topology is a broadcast star and the number of wavelengths is not constrained. In other words, the only constraint is on the logical degree of the nodes. For this case, earlier papers have presented heuristic algorithms to design the logical topology to minimize congestion [5] or minimize the average propagation delay weighted by the traffic [3]. These include heuristics based on evolving successive topologies by exchanging links with low utilizations [5], simulated annealing and genetic algorithms [3], and a stochastic ruler algorithm [6]. The computational aspects of solving the problem of minimizing congestion were studied in [7]. The usual MILP formulation was enhanced by adding a number of inequalities and suitable relaxations of the MILP were used to obtain good lower bounds. Computational results were presented for eight-node networks.

Recently, the problem of designing a logical topology over a wavelength-routed physical topology was considered in [8], [4], and [9]. Chlamtac *et al.* [8] proposed algorithms for embedding regular torus or hypercube logical topologies with a focus on the assignment of wavelengths to lightpaths rather than the optimization of the overall performance of the logical topology. Mukherjee *et al.* [4] proposed a heuristic for embedding a hypercube logical topology with the objective of minimizing the average weighted propagation delay, but did not take into account capacity constraints on lightpaths, and did not consider the problem of explicitly assigning wavelengths to lightpaths. Thus, throughput was neglected and if the required degree of the logical topology is at least equal to the maximum degree of the physical topology, the solution to the optimization problem stated in [4] is to make the logical topology identical to the physical topology. Zhang and Acampora [9] proposed a heuristic based on sequentially assigning a single wavelength to all possible lightpaths in order of decreasing traffic before proceeding to the next wavelength. The objective was to to maximize the amount of traffic carried in one hop from its source to its destination, but degree and delay constraints on the logical topology were ignored. Other related work [10]–[12], [2] addresses specifically the problem of dynamically routing lightpaths, which may be thought of one of the components of the overall logical topology design problem.

D. Outline of the Paper

In Section II, we give a precise formulation of the logical topology design and routing problem as a mixed integer programming problem (ignoring the constraint on the number of available wavelengths) and solve it for various values of the degree and delay bounds in a six-node example. Since this problem is computationally difficult for larger networks, we split the problem into the logical topology design and routing subproblems. In Section III, we derive a lower bound on the congestion of any logical topology, given the traffic distribution matrix and the logical degree, but ignoring the constraints imposed by the physical topology and the limited number of wavelengths. This bound was stated in [13] and [7], and proved in [14], but we present an alternate proof of the same bound. This generalizes a similar bound derived in [15] for uniform traffic. We then state an iterative linear programming lower bound that was derived in [7] for a similar problem. This is particularly useful for large problems. Some lower bounds on the number of wavelengths required to realize the logical topology are derived in Section IV. In Section V, we propose several heuristics for designing logical topologies. In Section VI, we study their performance relative to the bounds derived on the congestion and the number of wavelengths for a 14-node NSFNET backbone network. We also study how the number of wavelengths scales with the size of the network by considering a number of randomly chosen networks. Section VII concludes the paper.

II. Logical Topology Design and Routing Problem

Let $b_{ij} \in \{0, 1\}$ be binary variables, one for each possible link such that $b_{ij} = 1$ if there is a logical link from node i to node j in the logical topology and $b_{ij} = 0$ otherwise. Let d_{ij} denote the propagation delay on logical link (i, j), $d^{\max}$ the maximum propagation delay on the physical topology between any s-d pair and $\alpha d^{\max}$ the maximum permissible average delay between any s-d pair. Note that d_{ij} is the sum of the propagation delays on the physical links over which the logical link is established, and thus is determined by the actual routing of the logical links on the physical topology. In our examples we assume that logical links are established on the shortest propagation-delay routes in the physical topology. Let λ_{ij}^{sd} be the arrival rate of packets from s-d pair (s, d) on link (i, j), λ_{ij} the arrival rate of packets on link (i, j) from all s-d pairs, and $\lambda_{\max}$ the maximum load on any link, viz. the *congestion*, which we seek to minimize. Then the logical topology design and routing problem which was stated informally in Section I can be formulated as the following mixed-integer linear programming problem (MILP), assuming enough wavelengths are available so that wavelength assignment constraints can be ignored: (we will take the wavelength assignment constraints into account in the heuristics later)

$$\min \lambda_{\max}$$

subject to the following.

Flow conservation at each node:

$$\sum_j \lambda_{ij}^{sd} - \sum_j \lambda_{ji}^{sd} = \begin{cases} \lambda^{sd}, & \text{if } s = i \\ -\lambda^{sd}, & \text{if } d = i, \\ 0, & \text{otherwise} \end{cases} \quad \text{for all } s, d, i$$

Total flow on a logical link:

$$\lambda_{ij} = \sum_{s,d} \lambda_{ij}^{sd}, \quad \text{for all } i, j$$

$$\lambda_{ij} \leq \lambda_{\max}, \quad \text{for all } i, j$$

$$\lambda_{ij}^{sd} \leq b_{ij} \lambda^{sd}, \quad \text{for all } i, j, s, d$$

Average delay constraint for each s-d pair:

$$\sum_{i,j} \lambda_{ij}^{sd} d_{ij} \leq \lambda^{sd} \alpha d^{\max}$$

Degree constraints:

$$\sum_i b_{ij} = \Delta_l, \quad \text{for all } j$$

$$\sum_j b_{ij} = \Delta_l, \quad \text{for all } i$$

$$\lambda_{ij}^{sd}, \lambda_{ij}, \lambda_{\max} \geq 0, \quad \text{for all } i, j, s, d$$

$$b_{ij} \in \{0, 1\} \quad \text{for all } i, j.$$

Remarks:

1) Note that the physical topology does not appear in the formulation except by way of the input variables d_{ij}, which are determined by the routing of logical links on the physical topology.
2) As formulated, the above MILP does not allow multiple links between the same pair of nodes in the logical topology. But if multiple links should be allowed (for $\Delta_l \geq 2$), using the fact that the multiplicity can be at most Δ_l, this can be done by replacing the variables, b_{ij} by $b_{ijk}, k = 0, 1, \cdots, \Delta_l - 1$ (thus, we again have one variable for each possible link) and modifying the constraints accordingly.
3) The MILP formulation allows traffic between each source–destination pair to be split across multiple possible routes.
4) Note that the logical link capacity C does not appear in the formulation. We must have $C > \lambda_{\max}$.
5) The delay bound is formulated as a uniform bound on the the average delay between every s-d pair, i.e., the average delay between any s-d pair is restricted to at most α times the worst-case propagation delay between any s-d pair in the network. This can be replaced by possibly different bounds on the average delay for each s-d pair, e.g., $\alpha d^{\max}$ could be replaced by αd^{sd} thus restricting the average delay seen by s-d pair (s, d) to α times the propagation delay between s and d in the physical topology.
6) The delay bounds as formulated above are on the average delay between an s-d pair (over all paths in the logical topology, weighted by the traffic routed on those paths). It may be more desirable to upper-bound the delay on *every* path over which traffic is routed between an s-d pair but we do not know of a way to do this that is not significantly more complex.

In general, we expect that the tighter the delay constraint (smaller α), the larger the minimum achievable congestion. Moreover, with tighter delay constraints, we expect that there will be fewer logical links that span multiple physical links since such logical links have larger delays.

We solve the above MILP for the six-node network shown in Fig. 4 for various values of Δ_l and α using a branch-and-bound routine available in the IBM Optimization Subroutine Library [16]. The traffic matrix used is shown in Table I. Each entry is chosen at random from a uniform distribution in (0, 1). For $\Delta_l = 1$, there are three possible topologies, depending on the value of α as shown in Fig. 5. For $\alpha \geq 2.8$ (weak or no delay constraint), the achievable congestion is 7.078. However the links in this logical topology bear little resemblance to

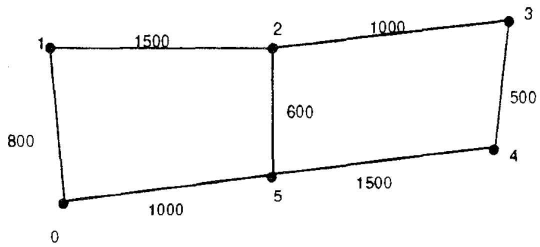


Fig. 4. A sample six-node wide-area network. The numbers on the links represent distances between nodes, or equivalently, relative propagation delays.

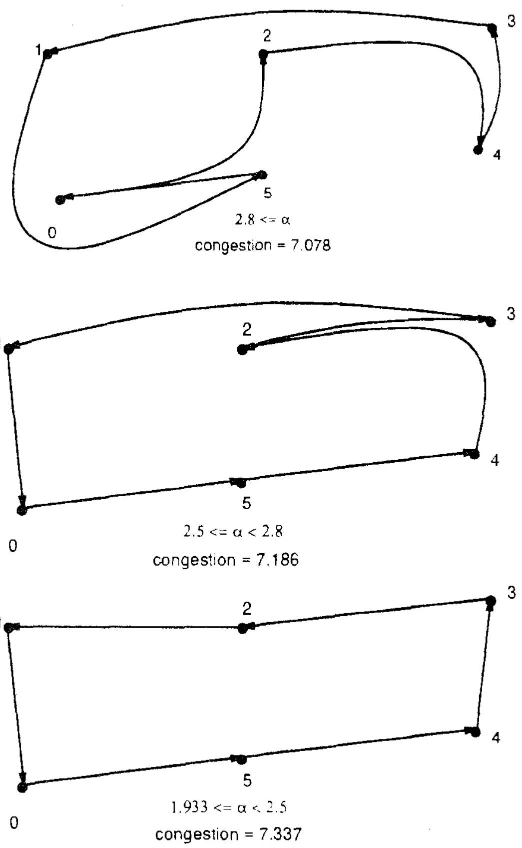


Fig. 5. Optimal degree-one logical topologies for the six-node network for different choices of α (which is proportional to the maximum permitted propagation delay).

the links in the physical topology with many of the logical links spanning multiple physical links. For $2.5 \leq \alpha < 2.8$, the resulting congestion is 7.186. For $58/30(\approx 1.933) \leq \alpha < 2.5$, the resulting congestion is 7.337 and all the links in this logical topology are links in the physical topology as well. For $\alpha < 1.933$, there is no feasible logical topology. It is interesting to observe that while the achievable congestion does increase with tighter delay constraints, this increase is not very much in this example. For $\Delta_l = 2, 3, 4, 5$, the achievable values of the congestion for various values of α are shown in Table II. Again we observe that tight delay constraints (small α) do not lead to significantly reduced values of the achievable congestion. Moreover, the higher the degree, the smaller the

TABLE I
TRAFFIC MATRIX FOR THE SIX-NODE NETWORK

0.000	0.537	0.524	0.710	0.803	0.974
0.391	0.000	0.203	0.234	0.141	0.831
0.060	0.453	0.000	0.645	0.204	0.106
0.508	0.660	0.494	0.000	0.426	0.682
0.480	0.174	0.522	0.879	0.000	0.241
0.950	0.406	0.175	0.656	0.193	0.000

TABLE II
MINIMUM ACHIEVABLE CONGESTION VALUES OBTAINED BY SOLVING THE MILP FOR THE SIX-NODE NETWORK FOR VARIOUS VALUES OF THE DEGREE Δ_l AND THE DELAY PARAMETER α. MFT IS THE MINIMUM FLOW TREE LOWER BOUND FROM SECTION III. AN "X" INDICATES THAT THE MILP IS INFEASIBLE

$\Delta_l \rightarrow$	1	2	3	4	5
MFT	5.692	1.673	0.974	0.657	0.475
MILP ($\alpha = 1$)	X	X	1.210	0.887	0.710
MILP ($\alpha = 2$)	7.337	2.042	1.183	0.887	0.710
MILP ($\alpha = \infty$)	7.078	2.042	1.183	0.887	0.710

value of α that we can impose without significant increase in congestion. We conjecture that this is probably true for most of the interesting cases and are studying this phenomenon further. Finally, we note that in this example, allowing multiple links between the same pair of nodes by modifying the MILP as indicated in the remark above, does not alter the solutions we obtained.

The MILP stated above becomes computationally intractable for larger networks. Therefore, we decompose the problem into the subproblems of logical topology design and routing. For a given logical topology, the routing subproblem is the linear program resulting by correspondingly fixing the values of the b_{ij} in the above MILP and is computationally quite tractable, at least for moderate-sized networks (tens of nodes).

For the topology design subproblem, we will probably need to use heuristic algorithms. However, in order to evaluate the goodness of these heuristic algorithms (which provide upper bounds on congestion) we will need good lower bounds on the achievable congestion values. In the next section, we derive such lower bounds. Since we have not taken the number of wavelengths available as a constraint in formulating the logical topology design problem so far, we then derive lower bounds on the number of wavelengths required to realize a logical topology. After that, we consider several topology design heuristics.

III. LOWER BOUNDS ON CONGESTION

Let $P = (p_{sd})$ be the average traffic distribution matrix, with $\Sigma_s \Sigma_d p_{sd} = 1$, i.e., p_{sd} is the probability that a new packet is from s to d. Let $\lambda^{sd} = p_{sd} r$ be the arrival rate of packets for source–destination pair (s, d), i.e., r is the total arrival rate of packets to the network and let $T = (\lambda^{sd})$ denote the traffic matrix. Let λ_{ij} be the arrival rate of packets for logical link (i,j). Our objective is to minimize the congestion $\lambda_{\max} = \max_{ij} \lambda_{ij}$. In this section, given the traffic distribution P and the maximum degree Δ_l, we derive a lower bound on $\lambda_{\max}$ that must be satisified by *any* logical topology with maximum degree Δ_l and any routing scheme on that logical topology.

For a specific logical topology, routing scheme, and source–destination pair (s,d), let $a_{ij}^{sd} \in [0,1]$ denote the fraction of packets that are routed on paths using logical link (i,j). Then

$$\lambda_{ij} = \sum_{sd} \lambda^{sd} a_{ij}^{sd}.$$

The traffic-weighted average number of hops between an s-d pair is then

$$\overline{H} = \sum_{sd} p_{sd} \sum_{ij} a_{ij}^{sd} = (1/r) \sum_{ij} \lambda_{ij}.$$

In a directed network with E_l links, we then have

$$\lambda_{\max} \geq (1/E_l) \sum_{ij} \lambda_{ij} = r\overline{H}/E_l. \tag{1}$$

For a given traffic distribution and maximum degree Δ_l, a lower bound on $\overline{H}$ for any logical topology and routing scheme was derived in [5] as follows: Note that in any logical topology with N nodes and maximum degree Δ_l, there can at most $N\Delta_l$ s-d pairs one hop apart, $N\Delta_l^2$ s-d pairs two hops apart, $N\Delta_l^3$ s-d pairs three hops apart, etc. In the idealized topology (which may not exist), the $N\Delta_l$ s-d pairs with the largest traffic would be connected by one hop paths, the next $N\Delta_l^2$ s-d pairs in descending order of traffic by two hop paths, and so on. Let h_{ij} denote the number of hops in the shortest hop path from node i to node j in this idealized logical topology. Then $S_k = \Sigma_{h_{ij}=k}\, p_{ij}$ denotes the traffic between s-d pairs for which the destination is at a minimum distance of k hops from the source in this idealized topology. Using this it can be shown that for any logical topology with maximum degree Δ_l and the given traffic distribution p_{ij}

$$\overline{H} \geq \sum_k k S_k.$$

We now obtain a stronger lower bound on $\overline{H}$. Observe that for each source, there can be at most Δ_l destinations one hop away, Δ_l^2 destinations two hops away, etc. Now consider an idealized topology in which for each source the Δ_l destinations with the largest traffic are connected by one hop paths, the next Δ_l^2 destinations in descending order of traffic are connected by two hop paths and so on. We show that $\overline{H}$ for this idealized topology is a lower bound on $\overline{H}$ for any topology with maximum out-degree Δ_l and any routing scheme on that topology.

For $1 \leq i \leq N$, let π_i be a permutation of $(1,2,\cdots,N)$, such that

$$p_{i\pi_i(j)} \geq p_{i\pi_i(j')} \quad \text{if } j \leq j'. \tag{2}$$

We assume that $p_{ii} = 0$ so that w.l.o.g. we can set $\pi_i(N) = i$. Let $m = m(N, \Delta_l)$ be the largest integer such that

$$N > 1 + \Delta_l + \cdots + \Delta_l^{m-1} = \frac{\Delta_l^m - 1}{\Delta_l - 1}.$$

Let $n_k = \Sigma_{i=1}^k \Delta_l^i$ for $1 \leq k \leq m-1, n_m = N-1$ and $n_0 = 0$.

Theorem 1: Let

$$\overline{H}_{\min} = \sum_{i=1}^{N} \sum_{k=1}^{m} \sum_{j=n_{k-1}+1}^{N-1} p_{i\pi_i(j)}.$$

Then, for all logical topologies with maximum degree Δ_l and all routing schemes on those topologies

$$\overline{H} \geq \overline{H}_{\min}.$$

Proof: Let h_{ij} denote the number of hops in the shortest hop path from node i to node j in a particular logical topology. Let $H_i = \max_j h_{ij}$, let

$$P_{ik} = \sum_{j:h_{ij}\geq k} p_{ij}, \quad \text{for } 1 \leq k \leq H_i$$

and let

$$Q_{ik} = \begin{cases} \sum_{j=n_{k-1}+1}^{N-1} p_{i\pi_i(j)}, & 1 \leq k \leq m \\ 0, & \text{otherwise.} \end{cases}$$

Since there can be at most Δ_l^k nodes that are k hops away from node i, the number of nodes (other than i) at $k-1$ or fewer hops from node i is at most $\Delta_l + \Delta_l^2 + \cdots + \Delta_l^{k-1} = n_{k-1}$. Hence

$$|\{j\colon h_{ij} \geq k\}| \geq N - 1 - n_{k-1}.$$

Therefore, $P_{ik} \geq Q_{ik}$ for $1 \leq k \leq H_i$ [using (2)]. Therefore,

$$\begin{aligned} \overline{H} &\geq \sum_{i=1}^{N} \sum_{j=1}^{N} p_{ij} h_{ij} \\ &= \sum_{i=1}^{N} \sum_{k=1}^{H_i} P_{ik} \\ &\geq \sum_{i=1}^{N} \sum_{k=1}^{m} Q_{ik} \\ &= \sum_{i=1}^{N} \sum_{k=1}^{m} \sum_{j=n_{k-1}+1}^{N-1} p_{i\pi_i(j)}. \end{aligned}$$

Q.E.D.

Minimum Flow Tree Bound: Combining (1) and Theorem 1, we get

$$\lambda_{\max} \geq r\overline{H}_{\min}/E_l$$

which we will refer to as the *minimum flow tree* (MFT) bound following [13]. This bound was also stated (for $\Delta_l = 2$) in [7] where some similar stronger bounds are also given.

We next state a much stronger iterative LP-relaxation lower bound on the congestion and we will use the above MFT bound as a starting point for its calculation.

In the MILP stated in Section II, we have considered each s-d pair as a commodity. This is usually referred to as the *disaggregate* formulation [7]. We can get a more tractable *aggregate* MILP formulation by identifying a commodity with each source, rather than each s-d pair as follows. (The aggregate problem is more tractable because it has fewer variables and constraints.)

Let $\lambda^s = \Sigma_d \lambda^{sd}$ be the total traffic from source s, λ_{ij}^s be the arrival rate of packets from source s on logical link (i, j), λ_{ij} the arrival rate of packets on logical link (i, j) from all sources, and $\lambda_{\max}$ the maximum load on any logical link, viz. the *congestion*, which we seek to minimize. Then the aggregate MILP formulation is as follows:

$$\min \lambda_{\max}$$

subject to the following.

Flow conservation at each node:

$$\sum_j \lambda_{ij}^s - \sum_j \lambda_{ji}^s = \begin{cases} \lambda^s, & \text{if } s = i, \\ -\lambda^{si}, & \text{if } s \neq i, \end{cases} \quad \text{for all } s, i$$

Total flow on a logical link:

$$\lambda_{ij} = \sum_s \lambda_{ij}^s, \quad \text{for all } i, j$$

$$\lambda_{ij} \le \lambda_{\max} \quad \text{for all } i, j$$

$$\lambda_{ij}^s \le b_{ij} \lambda^s, \quad \text{for all } i, j, s$$

Average delay constraint for each source:

$$\sum_{i,j} \lambda_{ij}^s d_{ij} \le \lambda^s \alpha d^{\max}$$

Degree constraints:

$$\sum_i b_{ij} = \Delta_l, \quad \text{for all } j$$

$$\sum_j b_{ij} = \Delta_l, \quad \text{for all } i$$

$$\lambda_{ij}^s, \lambda_{ij}, \lambda_{\max} \ge 0, \quad \text{for all } i, j, s$$

$$b_{ij} \in \{0, 1\} \quad \text{for all } i, j.$$

Following [7], we add the following additional constraint to the above MILP:

$$\lambda_{\max} \ge \sum_s \lambda_{ij}^s + \lambda_{\max}^L (1 - b_{ij}), \quad \text{for all } i, j.$$

Here $\lambda_{\max}^L$ is any *a priori* lower bound on $\lambda_{\max}$, e.g., the MFT lower bound derived earlier.

Remarks:

1) The LP-relaxation of the above MILP is the linear program obtained by replacing the constraints "$b_{ij} \in 0, 1$ for all i, j," by "$0 \le b_{ij} \le 1$ for all i, j." Note that the additional constraint added following [7] is superfluous in the MILP but (usually) becomes active in the LP-relaxation. If we set $\lambda_{\max}^L$ to the MFT lower bound and solve the LP-relaxation of the above problem, we get another lower bound on $\lambda_{\max}$ which we will denote as $\lambda_{\max}^L(1)$. Iteratively, we can set $\lambda_{\max}^L = \lambda_{\max}^L(i), i \ge 1$, and solve the LP-relaxation to get an improved lower bound $\lambda_{\max}^L(i+1)$. We will refer to these bounds as the iterative LP-relaxation bounds. We will use $i = 25$ in the NSFNET examples to be considered below and call it the *LP lower bound*. Iterating further (beyond $i = 25$) leads to very little improvement in the lower bound.
2) The aggregate MILP is much more tractable than the disaggregate one; however one shortcoming of the aggregate formulation is that the delay constraints are on the average delay experienced by packets from each source rather than between each s-d pair. If the aggregate MILP is solvable but the disaggregate MILP is not, we can solve the (disaggregate) routing subproblem on the topology produced by the aggregate MILP, to check whether the solution satisfies the more stringent delay constraints of the disaggregate formulation. If it does, then the optimal solution to the disaggregate MILP is the same as that of the aggregate MILP; otherwise we have a possibly suboptimal solution which may be still be adequate for practical purposes.

IV. Lower Bounds on Number of Wavelengths

Given an undirected physical topology, $G_p(N, E_p)$, with N nodes and E_p undirected edges, how many wavelengths, $\Lambda(G_p, \Delta_l)$ do we need to set up a regular, directed logical topology with degree Δ_l, i.e., Δ_l lightpaths to and from each node?

If the minimum degree of the physical topology is δ_p, then,

$$\Lambda \ge \lceil \Delta_l / \delta_p \rceil.$$

To prove this, consider a node with physical degree δ_p, say node A. The Δ_l lightpaths *from* node A must be routed over one of these δ_p edges so that the average number of lightpaths routed over each one of these edges is Δ_l/δ_p. Since the number of lightpaths traversing an edge is an integer, at least one of these δ_p edges, say α, must have $\lceil \Delta_l/\delta_p \rceil$ lightpaths routed over it. The bound follows by noting that all these lightpaths traverse the edge α in the same direction, i.e., from node A, and thus must use distinct wavelengths.

In some cases, the following argument leads to a better bound on Λ. We first replace each undirected edge in G_p by a pair of edges, directed in opposite directions. Let h_{ij} denote the number of hops in the shortest path from node i to node j. For each node i, let $l_i(\Delta_l)$ denote the sum of the Δ_l smallest values of h_{ij} for different j. The total number of physical edges traversed, including repetitions, by the Δ_l lightpaths from node i must be at least $l_i(\Delta_l)$. Therefore the total number of physical edges traversed by the $N\Delta_l$ lightpaths in the logical topology is at least $\Sigma_i \, l_i(\Delta_l)$. Since the average number of lightpaths per physical (directed) edge is a lower bound on Λ, we have

$$\Lambda \ge (1/2E_p) \sum_i l_i(\Delta_l).$$

Note that we divide by $2E_p$, the number of directed edges in the physical topology, and not by E_p, the number of undirected edges, in computing a lower bound on the number of wavelengths since it is only the lightpaths that traverse a physical topology edge in the same direction that are constrained to use different wavelengths.

We will use the better among the two lower bounds in the examples to be considered later.

V. Topology Design Algorithms

HLDA: We first consider a simple logical topology design algorithm, which we call HLDA (for heuristic topology design algorithm), that attempts to place logical links between nodes in order of descending traffic. The idea behind this heuristic is that routing most of the traffic in one hop may lower the congestion. The HLDA does not take delay constraints into account when designing the logical topology but these constraints can be imposed in the routing phase. A pseudo code description of the algorithm is given below.

Step 1: Given the traffic distribution matrix $P = (p_{ij})$, make a copy $Q = (q_{ij}) = P$.

Step 2: Select the source destination pair $(i_{\max}, j_{\max})$ with largest traffic, i.e., $q_{i_{\max}j_{\max}} = \max_{ij} q_{ij}$.

If we have tried all source–destination pairs with nonzero traffic already, then go to Step 4.

Step 3: If node $i_{\max}$ has fewer than Δ_l outgoing edges, and node $j_{\max}$ fewer than Δ_l incoming edges, then

> find lowest available wavelength on the shortest propagation-delay path between $i_{\max}$ and $j_{\max}$ in G_p. (If there is more than one shortest path, scan them sequentially)
> If wavelength available then
>> Create logical edge $i_{\max}, j_{\max}$
>> Find source destination pair i', j' with next highest traffic, i.e.,
>> $q_{i'j'} = \max_{i \neq i_{\max}, j \neq j_{\max}} q_{ij}$
>> Set $q_{i_{\max}j_{\max}} = q_{i_{\max}j_{\max}} - q_{i'j'}$
>> Go to Step 2.
>
> else
>> $q_{i_{\max}j_{\max}} = 0$; go to Step 2.

else $q_{i_{\max}j_{\max}} = 0$; go to Step 2.

Step 4: If we do not yet have $N\Delta_l$ edges, place as many remaining logical edges as possible at random so that degree constraints are not violated and a wavelength can be found on the shortest path for the logical edge.

Note that HLDA places multiple logical links between nodes with very high traffic, if the "residual" traffic (see Step 3) between such a node pair is larger than the traffic between the next highest node pair.

MLDA: We next consider another heuristic, which we call MLDA (for minimum-delay logical topology design algorithm) which is only defined if Δ_l is larger than the degree of the physical topology. If this is the case, the MLDA creates a pair of directed logical edges for each physical edge and the remaining edges are added according to the HLDA. Thus the logical topologies created by the MLDA are capable of routing all packets on the shortest physical path between every pair of nodes and therefore, capable of satisfying the tightest delay constraints that are physically realizable; hence the term "minimum-delay."

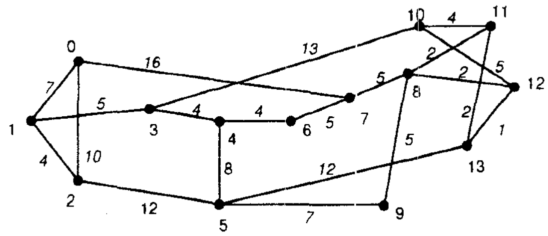


Fig. 6. The 14-node NSFNET network.

TILDA: We also consider another algorithm TILDA (for traffic independent logical topology design algorithm). TILDA designs logical topologies regardless of the traffic. It first places logical edges between all one-hop neighbors in the physical topology, then between all two-hop neighbors (provided that there are no logical edges between them already), then between all three-hop neighbors (provided that there are no logical edges between them already), etc., provided the degree constraints are not violated. Since lightpaths that use as few physical topology edges as possible will tend to keep the number of lightpaths that use a physical edge small, TILDA attempts to minimize the number of wavelengths required, and may be an appropriate choice if the traffic is unknown or known to be uniform.

LPLDA: Consider the solution produced by the iterative LP-relaxation bounds of Section II at some suitable iteration (iteration # 25 in the NSFNET examples to be considered below). Denote the values of b_{ij} in this solution by b'_{ij}. $b'_{ij} \in [0, 1]$. In the LP logical topology design algorithm (LPLDA) we round the values of b'_{ij} to zero or one, according to the following procedure, to obtain a logical topology: We order the b'_{ij} in decreasing order and starting with the largest b'_{ij}, we round each successive value of b'_{ij} to one, provided the degree constraints are not violated, and to zero, otherwise.

RLDA: For comparison purposes, we use another algorithm (RLDA) that places logical edges entirely at random, subject to finding a lightpath for each edge and not violating degree constraints, but ignoring the traffic matrix altogether.

VI. Examples

A. NSFNET

Figure 6 shows the 14-node NSFNET backbone network. For this network, we will use two different traffic patterns, P_1 and P_2, shown in Table III and Table IV respectively. P_1 is a traffic pattern created by picking 42 (an average of three per node) s-d pairs at random and allocating a random amount of traffic chosen from a uniform distribution in (0, 100) for each s-d pair. Each remaining s-d pair is then allocated a random amount of traffic chosen from a uniform distribution in (0,1). This captures a situation where most of the network traffic is concentrated among 42 pairs, with little traffic among the remaining ones. P_2 corresponds to a measured traffic distribution taken from [4] with traffic distributed more evenly over a large number of s-d pairs.

In Fig. 7, we plot the minimum achievable congestion as a function of the degree for logical topologies designed by

TABLE III
THE TRAFFIC MATRIX P_1 FOR THE 14-NODE NSFNET

0.000	33.029	32.103	26.008	0.525	0.383	82.633	31.992	37.147	0.568	0.358	0.544	0.651	0.160
0.546	0.000	0.984	0.902	0.866	0.840	0.013	62.464	0.475	0.001	0.342	0.925	0.656	0.501
35.377	0.459	0.000	0.732	0.272	0.413	28.242	0.648	0.909	0.991	56.150	23.617	1.584	0.935
0.739	0.225	0.296	0.000	0.896	0.344	0.012	84.644	0.293	0.208	0.755	0.106	0.902	0.715
0.482	96.806	0.672	51.204	0.000	0.451	0.979	0.814	0.225	0.694	0.504	0.704	0.431	0.333
0.456	0.707	0.626	0.152	0.109	0.000	0.804	0.476	0.429	0.853	0.280	0.322	90.503	0.212
0.042	0.067	0.683	0.862	0.197	0.831	0.000	0.585	67.649	56.138	0.896	0.858	73.721	0.582
0.616	0.640	0.096	97.431	0.308	0.441	0.299	0.000	0.161	0.490	0.321	0.638	82.231	0.376
0.786	0.323	0.676	0.359	0.019	50.127	12.129	0.650	0.000	0.483	45.223	58.164	0.894	0.613
0.037	0.318	0.367	2.981	0.976	0.629	0.525	0.293	0.641	0.000	33.922	0.228	0.995	71.905
12.609	0.479	0.146	0.174	0.181	0.072	23.080	0.671	0.634	0.759	0.000	0.725	0.592	0.445
0.887	0.004	1.614	0.471	0.120	0.263	0.585	0.086	0.157	95.633	42.828	0.000	0.527	0.021
9.019	0.569	0.936	0.975	81.779	0.573	0.738	0.410	0.490	0.948	0.154	0.145	0.000	0.436
20.442	0.515	0.719	0.089	39.269	49.984	0.720	0.863	0.858	0.490	0.106	0.765	0.059	0.000

TABLE IV
THE TRAFFIC MATRIX P_2 FOR THE 14-NODE NSFNET

0.000	1.090	2.060	0.140	0.450	0.040	0.430	1.450	0.510	0.100	0.070	0.030	0.000	0.330
11.710	0.000	8.560	0.620	11.120	7.770	3.620	15.790	3.660	16.610	2.030	37.810	4.830	13.190
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.310	3.410	13.640	0.000	1.900	0.600	0.700	2.880	2.000	3.260	3.070	6.690	0.080	4.010
0.280	67.510	19.020	3.430	0.000	4.030	10.770	62.220	24.020	17.920	0.450	79.030	9.970	5.290
0.000	5.810	3.420	5.520	3.400	0.000	2.610	2.680	0.870	3.870	0.040	0.840	0.060	2.480
1.750	22.020	102.310	4.470	22.030	7.900	0.000	114.100	19.820	21.950	0.780	71.400	0.330	32.840
2.390	63.840	210.300	8.520	28.210	2.660	97.080	0.000	43.950	33.000	11.370	48.630	5.530	13.850
6.450	18.930	37.350	6.000	24.990	6.810	25.060	61.020	0.000	39.620	14.520	127.500	23.340	0.760
0.050	35.290	10.260	3.730	22.340	9.480	4.980	57.080	6.840	0.000	6.300	17.640	5.910	0.760
0.100	1.020	3.130	1.690	0.240	0.060	0.810	1.450	0.580	7.120	0.000	0.840	0.060	0.500
1.280	26.150	1.000	5.940	24.860	1.320	5.490	40.570	29.530	22.370	10.500	0.000	1.010	0.540
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.730	29.090	13.630	9.890	35.610	12.070	6.440	28.790	4.670	0.000	3.990	0.000	10.750	0.000

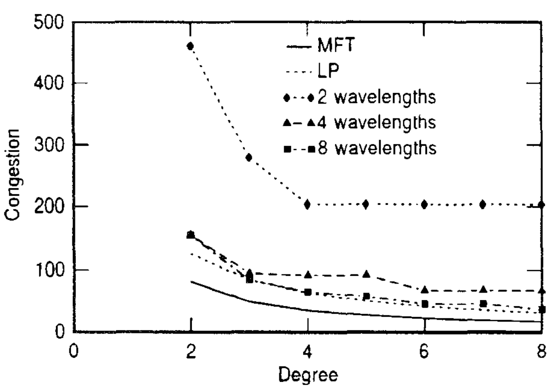


Fig. 7. Minimum achieved congestion (in arbitrary units) versus logical degree for different numbers of the available wavelengths for the traffic matrix P_1 by the topology design algorithm HLDA in the 14-node NSFNET example.

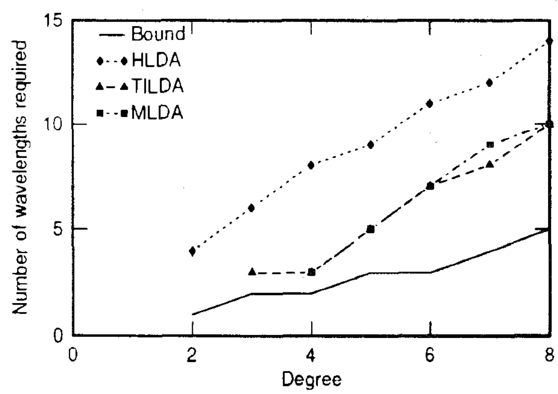


Fig. 8. Number of wavelengths required for traffic pattern P_1 for the 14-node NSFNET example. The lower bound shown is the one derived in Section IV.

HLDA for traffic pattern P_1 and for different numbers of available wavelengths. The delay parameter α is assumed to be ∞. Observe that with eight wavelengths we are close to the LP lower bound, which is also plotted in the figure. Observe also that the LP lower bound is a much better bound than the MFT bound—this is true for almost all the examples considered. It takes about 1 min of computation time to compute each iteration of the LP lower bound on an IBM RS/6000-580 workstation.

Next, we plot the number of wavelengths required as a function of the degree for traffic pattern P_1 in Fig. 8. We define the number of wavelengths required, Λ, to be the minimum number required so that the topology design algorithm never fails to place a logical edge due to the unavailability of a wavelength. For a particular degree, the congestion generally decreases as the number of wavelengths is increased but soon reaches a minimum value after which increasing the number of wavelengths has no effect (since the logical topology does not change) so that Λ is usually also the smallest number of wavelengths for which the minimum congestion is achieved.[2] The lower bound derived in Section IV is also plotted. Note that HLDA requires a larger number of wavelengths than MLDA or TILDA. This is because MLDA and TILDA tend to place logical edges using fewer hops on average in the physical topology. Note that even HLDA requires only a relatively small number of wavelengths, for example, eight for logical degree four.

In Table V, we show the achievable congestion versus degree for the different design algorithms, assuming the number of wavelengths is not a constraint, for the traffic pattern P_1.

[2] In some cases, we may be able to achieve the minimum congestion even with a slightly smaller number of wavelengths than Λ.

TABLE V

CONGESTION (ARBITRARY UNITS) VERSUS LOGICAL DEGREE FOR TRAFFIC PATTERN P_1 (NO CONSTRAINT ON THE NUMBER OF WAVELENGTHS) FOR THE NSFNET EXAMPLE FOR VARIOUS TOPOLOGY DESIGN ALGORITHMS. AN "X" INDICATES THAT EITHER THE TOPOLOGY DESIGN PROBLEM OR THE ROUTING PROBLEM IS INFEASIBLE

Degree	MFT	LP	MILP	LPLDA	HLDA	MLDA	TILDA	RLDA
2	81.93	126.18	209.17	243.43	155.37	X	X	266.49
3	49.18	84.53	103.03	102.82	84.58	X	146.38	156.93
4	35.49	63.43	76.94	82.03	65.16	80.85	139.20	94.17
5	27.78	50.75	59.37	53.49	54.39	60.57	71.74	69.47
6	22.73	42.29	46.27	44.45	42.29	43.88	56.85	55.27
7	19.40	36.25	39.27	36.55	36.25	36.25	44.14	44.16
8	16.90	31.72	33.24	32.27	32.68	32.33	35.97	39.70

TABLE VI

CONGESTION (ARBITRARY UNITS) VERSUS LOGICAL DEGREE FOR TRAFFIC PATTERN P_2 (NO CONSTRAINT ON THE NUMBER OF WAVELENGTHS) FOR THE NSFNET EXAMPLE FOR VARIOUS TOPOLOGY DESIGN ALGORITHMS. AN "X" INDICATES THAT EITHER THE TOPOLOGY DESIGN PROBLEM OR THE ROUTING PROBLEM IS INFEASIBLE

Degree	MFT	LP	MILP	LPLDA	HLDA	MLDA	TILDA	RLDA
2	144.17	282.51	297.98	345.42	544.16	X	X	382.73
3	79.52	189.62	189.78	195.71	261.63	X	233.49	216.22
4	55.60	142.32	142.33	142.33	142.33	155.97	210.85	146.49
5	41.98	113.87	113.87	113.87	113.87	129.98	115.41	113.87
6	33.24	94.89	94.89	94.89	94.89	97.48	94.88	95.40
7	27.24	81.33	81.33	81.33	81.33	81.33	81.33	81.33
8	23.00	71.17	71.17	71.17	71.17	71.17	71.17	72.37

TABLE VII

ACHIEVABLE CONGESTION VALUES OBTAINED BY SOLVING THE ROUTING SUBPROBLEM ON THE DEGREE-4 LOGICAL TOPOLOGIES DESIGNED BY HLDA, MLDA AND RLDA FOR THE 14-NODE NSFNET FOR VARIOUS VALUES OF THE DELAY PARAMETER α FOR TRAFFIC PATTERN P_2. AN "X" INDICATES THAT THE ROUTING SUBPROBLEM IS INFEASIBLE

$\alpha \rightarrow$	1.0	1.5	2.0	∞
LPLDA	X	142.33	142.33	142.33
HLDA	X	X	142.33	142.33
RLDA	X	X	149.80	149.69
MLDA	156.17	155.97	155.97	155.97
TILDA	217.38	210.85	210.85	210.85

The delay parameter α is assumed to be ∞. The MFT and LP lower bounds are indicated. MILP indicates an upper bound obtained by using a branch and bound routine [16] to come up with integer solutions for the aggregated MILP formulation of Section III using less than 10^7 iterations. It takes approximately 50 h of CPU time on an IBM RS/6000-580 workstation to perform this many iterations. The congestion values for RLDA are averaged over four different random topologies. The LP lower bound is very close to the optimal value in all cases. Observe that HLDA achieves lower values of the congestion than the other algorithms in this example except for degree five, when LPLDA is slightly better. MLDA is fairly close to HLDA in performance while using fewer wavelengths as observed earlier. LPLDA also produces good topology designs. TILDA is significantly poorer, showing that good design algorithms should take the traffic into consideration. The MFT bound is also plotted for comparison.

In Table VI, we plot the corresponding data for the traffic pattern P_2. Again the LP lower bound is close to the optimal value, and LPLDA produces the best solutions. Surprisingly, we see that for small logical degrees RLDA performs better than HLDA, MLDA and TILDA. This leads us to conjecture that heuristic topology design algorithms like HLDA and that of [9] are of value only when the traffic is concentrated among a few s-d pairs. If the traffic is much denser, there appears to be little value in "maximize the one-hop traffic" heuristics for logical topology design. This is reasonable because in the latter cases we expect the multihop traffic to be the dominant factor in determining the achievable congestion and these heuristics do not take this into account. MLDA provides a reasonable compromise between using short hop paths (thereby allowing multihop traffic to be efficiently routed) as well as placing direct logical edges for those s-d pairs with a lot of traffic between them.

A better option is probably to use an iterative topology design algorithm that starting with an initial topology, iteratively solves the routing problem, identifies the congested links and tries to reduce congestion by slightly changing the logical topology using techniques similar to those described in [17].

In Table VII and Table VIII, we tabulate the achievable values of the congestion for various values of α for each of the five logical topology design algorithms: LPLDA, HLDA, RLDA, MLDA and TILDA, for degree four and degree six, respectively, for the traffic pattern P_2. As we expect, MLDA, LPLDA and TILDA design feasible topologies for tight delay constraints ($\alpha = 1$ or 1.5). For larger values of α, MLDA produces a congestion that is only slightly worse than HLDA. Overall, MLDA and LPLDA appear to be appropriate heuristics for logical topology design.

The behavior of TILDA is slightly surprising. While it is the best topology design algorithm in this example for degree six,

TABLE VIII
ACHIEVABLE CONGESTION VALUES OBTAINED BY SOLVING THE ROUTING SUBPROBLEM ON THE DEGREE-SIX LOGICAL TOPOLOGIES DESIGNED BY HLDA, MLDA AND RLDA FOR THE 14-NODE NSFNET FOR VARIOUS VALUES OF THE DELAY PARAMETER α FOR TRAFFIC PATTERN P_2. AN "X" INDICATES THAT THE ROUTING SUBPROBLEM IS INFEASIBLE

$\alpha \rightarrow$	1.0	1.5	2.0	∞
LPLDA	X	94.89	94.89	94.89
HLDA	X	94.89	94.89	94.89
RLDA	X	94.91	94.89	94.89
MLDA	97.60	97.48	97.48	97.48
TILDA	94.89	94.89	94.89	94.89

TABLE IX
NUMBER OF WAVELENGTHS REQUIRED BY TILDA FOR RANDOM GRAPHS WITH AVERAGE PHYSICAL DEGREE THREE

$\Delta_l \rightarrow$	3	4	5	6
Lower Bound	2	2	3	4
$N = 32$	3	4	5	7
$N = 64$	3	4	5	7
$N = 128$	4	4	7	8
$N = 256$	3	5	8	10

it is far worse than the others for degree four. This suggests that if the objective is to minimize congestion, the use of heuristics that do not take the traffic matrix into account may be a poor idea.

Based on these examples we can make the following conclusions: When the logical degree is small it is more important to set up shorter-hop lightpaths than it is to set up lightpaths for s-d pairs with a large amount of traffic (since that will compromise the number of ports available for connecting to nearby nodes). For larger logical degrees it helps to place logical edges between s-d pairs with a lot of traffic between them. However even in this case, in order to meet tight delay constraints it is desirable first to have logical edges between neighboring nodes and then to have edges between s-d pairs with heavy traffic between them. This is precisely what MLDA does. MLDA and TILDA also tend to use a smaller number of wavelengths than HLDA.

B. Random Graphs and Number of Wavelengths

We next consider a family of random graphs ranging in size N from 32 to 256 nodes, all of average physical degree three. (To generate a random graph with N nodes and average degree Δ_p, we first place N edges to create a cycle and then, for each of the remaining $N(\Delta_p/2 - 1)$ edges we choose, in succession, a pair of nodes randomly from the node pairs that are not connected by an edge.) Table IX shows the number of wavelengths required by TILDA as a function of the logical degree Δ_l and also the lower bound from Section IV. The lower bound here is independent of N for the values of N considered. Observe that the number of wavelengths required by TILDA grows very slowly with the number of nodes and is still small (≤ 10) for a 256-node network.

VII. CONCLUSION

This paper formulated the problem of designing a logical topology over a wavelength-routed physical topology. The formulation takes into account processing constraints at the nodes, constraints on the average-delay for each s-d pair and wavelength assignment on the lightpaths. For the case where the number of wavelengths is not limited, we formulated the problem as an MILP and solved it exactly for a six-node network. The results illustrated that 1) not imposing the delay constraint results in topologies that are "unnatural" in that nodes that are physically located in close proximity may have to use long, convoluted paths to communicate, and 2) imposing the delay constraints did not significantly reduce the congestion in the examples considered.

We then proposed a simple logical topology design heuristic (HLDA) that works well when the traffic is concentrated among a small fraction of the total number of source–destination pairs in the network, but does not work well when the traffic is distributed more evenly among the source–destination pairs, since it is a greedy heuristic based on assigning lightpaths to s-d pairs with large traffic. We also proposed a modification of this heuristic (MLDA) that is capable of realizing tighter delay constraints seemingly without much increase in the congestion, and also provides efficient routes for multihop traffic. We also explored the use of a traffic-independent heuristic (TILDA) and the LPLDA heuristic based on rounding the values of the solution to the relaxed linear programming problem.

We found that degree constraints may play a more significant role in limiting the performance of a logical topology than the number of wavelengths available. In the 14-node NSFNET backbone example considered, 10 wavelengths were sufficient to achieve the best congestion for nodal degrees up to eight and this was observed to be the case for larger random networks as well. Also MLDA and TILDA tend to design logical topologies using a smaller number of wavelengths than HLDA.

Overall, the LPDLA appears to minimize congestion when the traffic is more evenly distributed among a large number of s-d pairs, and MLDA appears to achieve a good compromise between having good throughput, tight delay constraints and using a small number of wavelengths over a range of logical degrees.

ACKNOWLEDGMENT

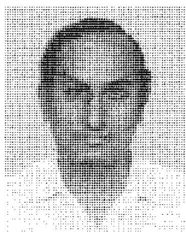
The authors thank R. Cruz and the reviewers for their comments, which helped us to improve our results as well as the presentation.

REFERENCES

[1] P. E. Green, *Fiber-Optic Networks.* Englewood Cliffs, NJ: Prentice-Hall, 1992.
[2] R. Ramaswami and K. N. Sivarajan, "Routing and wavelength assignment in all-optical networks," *IEEE/ACM Trans. Networking*, vol. 3, pp. 489–500, Oct. 1995.
[3] J. A. Bannister, L. Fratta, and M. Berla, "Topological design of the wavelength-division optical network," in *Proc. IEEE INFOCOM '90,* 1990, pp. 1005–1013.

[4] B. Mukherjee, S. Ramamurthy, D. Banerjee, and A. Mukherjee, "Some principles for designing a wide-area optical network," in *Proc. IEEE INFOCOM '94*, 1994.
[5] J.-F. P. Labourdette and A. S. Acampora, "Logically rearrangeable multihop lightwave networks," *IEEE Trans. Commun.*, vol. 39, pp. 1223–1230, Aug. 1991.
[6] A. Ganz and X. Wang, "Efficient algorithm for virtual topology design in multihop lightwave networks," *IEEE/ACM Trans. Networking*, vol. 2, pp. 217–225, June 1994.
[7] D. Bienstock and O. Günlük, "Computational experience with a difficult mixed-integer multicommodity flow problem," *Mathematical Programming*, vol. 68, pp. 213–237, 1995.
[8] I. Chlamtac, A. Ganz, and G. Karmi, "Lightnets: Topologies for high-speed optical networks," *J. Lightwave Technol.*, vol. 11, pp. 951–961, May/June 1993.
[9] Z. Zhang and A. S. Acampora, "A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength reuse," in *Proc. IEEE INFOCOM'94*, 1994, pp. 534–543.
[10] K. Bala, T. E. Stern, and K. Bala, "Algorithms for routing in a linear lightwave network," in *Proc. INFOCOM'91*, 1991, pp. 1–9.
[11] K. Bala, "Routing in linear lightwave networks," Ph.D. dissertation, Columbia Univ., CTR Tech. Rep. 393-93-02, 1993.
[12] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: an approach to high-bandwidth optical WAN's" *IEEE Trans. Commun.*, vol. 40, pp. 1171–1182, July 1992.
[13] B. Yener and T. Boult, "A study of upper and lower bounds for minimum congestion routing in lightwave networks," in *Proc. INFOCOM'94*, 1994, pp. 138–147.
[14] B. Yener and T. Boult, "Flow trees: A lower bound computation tool for network optimization," Columbia Univ., Tech. Rep. CUCS-006-94, 1994.
[15] K. N. Sivarajan and R. Ramaswami, "Lightwave networks based on de Bruijn graphs," *IEEE/ACM Trans. Networking*, vol. 2, pp. 70–79, Feb. 1994.
[16] IBM, *Optimization Subroutine Library: Guide and Reference*, 1991. Release 2.
[17] A. Kershenbaum, *Telecommunications Network Design Algorithms*. New York: McGraw-Hill, 1993.

Rajiv Ramaswami (S'88–M'90–SM'96), for a photograph and biography, see this issue, p. 762.



Kumar N. Sivarajan (S'88–M'90) received the B. Tech. degree in electrical engineering (Electronics) from the Indian Institute of Technology, Madras, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, in 1987 and 1990, respectively.

Since July 1990, he has been with the IBM T.J. Watson Research Center, Yorktowne Heights, NY. Since October 1994, he has also been with the Indian Institute of Science, Bangalore. His research interests are in the areas of optical, cellular and ATM networks where he has worked on network architecture, algorithms, and performance analysis.

Dr. Sivarajan was the recipient of the IEEE Charles LeGeyt Fortescue Fellowship for the academic year 1987/1988.