

High-Capacity Time-Domain Wavelength Interleaved Networks

Tony K. C. Chan, Eric W. M. Wong, *Senior Member, IEEE*, and Yiu-Wing Leung, *Senior Member, IEEE*

Abstract—Time-domain wavelength interleaved network (TWIN) is an elegant and cost-effective all-optical network designed by a group of researchers in Bell Labs. It emulates fast optical switching via fast tunable lasers at the network edge, so it does not need optical switching and buffering in the network core. TWIN can be upgraded to provide larger capacity by using more receivers at the nodes, where capacity is the aggregate data rate supported by the network. In this paper, we focus on making this upgrade resource-effective. Specifically, we exploit and optimize wavelength reuse so that the resulting network, called *high-capacity TWIN* (HC-TWIN), can better utilize its available resources to provide larger capacity while retaining the appealing advantages of TWIN. We formulate the problem of optimizing HC-TWIN, prove its NP-hardness, and design an efficient three-stage algorithm to solve it. Simulation results demonstrate that 1) HC-TWIN can provide larger capacity by realizing larger degree of wavelength reuse and 2) the three-stage algorithm can find optimal or close-to-optimal solutions.

Index Terms—All-optical networks, network architectures, network optimization.

I. INTRODUCTION

A group of researchers in Bell Labs designed an elegant and cost-effective all-optical network called *time-domain wavelength interleaved network* (TWIN) [1]. TWIN pushes the processing functions (e.g., optical switching and traffic grooming) to the network edge so that its network core has low complexity and cost. In TWIN, each node has a fast tunable laser for transmission and is assigned a multipoint-to-point tree for reception. These trees, called *destination trees* [2], are pre-provisioned at distinct wavelengths and overlaid on the physical network. Fig. 1 shows an example of a four-node TWIN. To transmit data to a destination node which uses destination tree t at wavelength λ for reception, a source node tunes its laser to λ and then transmits the data through the destination tree t . Other source nodes share this destination tree t via simple and efficient scheduling [2], [3]. TWIN has two salient features:

- 1) TWIN emulates fast optical switching via fast tunable lasers at the network edge [1]. A source node performs

this emulation by transmitting data bursts at different wavelengths to different destination nodes. As a result, TWIN does not need optical switching and buffering in the network core, and hence it has low complexity and cost.

- 2) TWIN performs traffic grooming optically [1]. The source nodes groom their traffic into the destination trees via simple scheduling and optical merging.

Traffic demands have been increasing because of more users with high-speed access and more bandwidth-demanding applications. In fact, the Internet traffic was doubling every six months [4], and it was envisioned that the bandwidth demand will grow exponentially in the next two decades [5]. It is desirable that a network design can provide large enough capacity to fulfill the ever-increasing traffic demands, where capacity refers to the aggregate data rate supported by the network. TWIN has a network capacity of WC [6], where W is the number of wavelengths in a fiber and C is the data rate of each wavelength channel (called *channel rate*). In principle, TWIN can provide larger network capacity by adopting more wavelength channels per fiber W or a higher channel rate C . In practice, both alternatives involve technological difficulties and higher cost (e.g., channel spacing is limited by nonlinear crosstalk [7], expansion of the usable wavelength band involves costly broadband optical amplification [5], [8], channel rates are limited by dispersion and nonlinear effects [7]–[9], higher channel rates involve faster and more expensive optical/electronic components, very fast components are not mature in terms of performance and/or cost [10], etc.).

TWIN can be upgraded to provide larger capacity by using more receivers at the nodes. In this paper, we focus on making this upgrade resource-effective and our contributions include the following.

- 1) We exploit and optimize wavelength reuse so that the resulting network, called *high-capacity TWIN* (HC-TWIN), can better utilize the available network resources to provide larger capacity for wider applicability while retaining the appealing advantages of TWIN. HC-TWIN includes the original TWIN design at one extreme and the light-path-based design at the other extreme.
- 2) We formulate and solve the problem of optimizing HC-TWIN on any given physical network. In this problem, we minimize the total number of destination trees required while fulfilling the given traffic demands and all the network constraints. This problem is called *destination tree construction* (DTC). We prove that the DTC problem is NP-hard and design an efficient three-stage heuristic algorithm to solve it.

The rest of the paper is organized as follows. In Section II, we describe HC-TWIN and discuss its characteristics. In

Manuscript received June 25, 2008; revised March 31, 2009. First version published April 21, 2009; current version published August 18, 2009. This work was supported by a grant from the HKBU Faculty Research Grant and a grant from the Research Grants Council of Hong Kong [Project CityU 121507].

T. K. C. Chan and Y.-W. Leung are with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong (e-mail: kctchan@gmail.com; ywleung@comp.hkbu.edu.hk).

E. W. M. Wong is with the Department of Electronic Engineering, City University of Hong Kong, Kowloon Tong, Hong Kong (e-mail: ewong@ee.cityu.edu.hk).

Digital Object Identifier 10.1109/JLT.2009.2021487

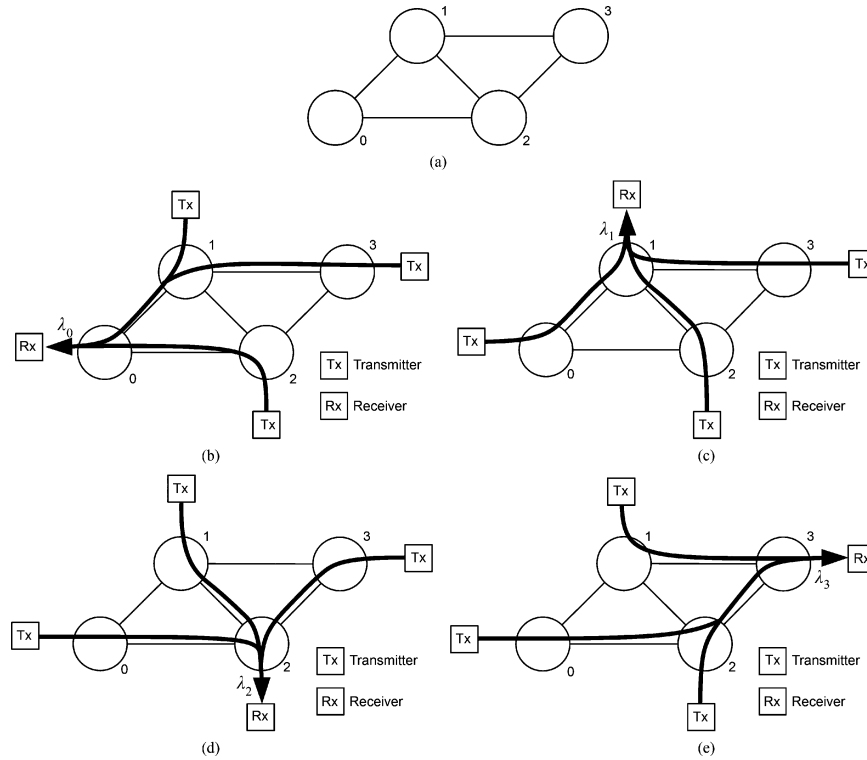


Fig. 1. Example of a four-node TWIN. (a) Physical network. (b) Destination tree for node 0 to receive data bursts from other nodes at wavelength λ_0 . (c) Destination tree for node 1 to receive data bursts from other nodes at wavelength λ_1 . (d) Destination tree for node 2 to receive data bursts from other nodes at wavelength λ_2 . (e) Destination tree for node 3 to receive data bursts from other nodes at wavelength λ_3 .

Section III, we formulate the destination tree construction problem and prove its NP-hardness. In Section IV, we design a three-stage heuristic algorithm to solve this problem. In Section V, we present simulation results to demonstrate the effectiveness of HC-TWIN and the three-stage algorithm. In Section VI, we conclude our research.

II. HIGH-CAPACITY TIME-DOMAIN WAVELENGTH INTERLEAVED NETWORKS

A. Provision of Large Capacity

HC-TWIN provides large network capacity through spatial reuse of wavelengths in different fibers. As a result, it can provide more destination trees on the same physical network to accommodate larger traffic demands. HC-TWIN reuses wavelengths in the following manner:

- 1) HC-TWIN reuses the same wavelength in different links. Fig. 2(a) shows an example in which node 0 receives data through two destination trees at the same wavelength λ_0 . These trees do not use the same link, so they can reuse the wavelength at λ_0 in different links. As a result, the network can provide two destination trees at λ_0 (while TWIN provides only one destination tree at λ_0). In general, a network can provide multiple destination trees at the same wavelength via spatial reuse.
- 2) HC-TWIN reuses the same wavelength in different fibers of the same link. When an optical network is being constructed, it is very cost-effective to install multiple fibers in each link because the fiber cost is much smaller than the fiber installation cost. The advantages of multifiber networks have been widely studied in the literature (e.g., see

[11]–[13]). With multiple fibers per link, we can realize a larger degree of spatial reuse of wavelengths. Fig. 2(b) shows an example in which we reuse the wavelength at λ_0 in different fibers to provide three destination trees at λ_0 [i.e., one more destination tree than the example shown in Fig. 2(a)]. In general, we can provide multiple destination trees at the same wavelength via spatial reuse of wavelengths in different fibers.

Using the above spatial reuse methods, it is necessary to determine the destination trees for every node to satisfy its traffic demand. If the network uses more destination trees, it needs more receivers and may need more transmitters. Specifically, a node needs n receivers if it uses n destination trees for reception, and it may need more transmitters if it uses more destination trees for transmission (the number of transmitters required essentially depends on its outgoing traffic volume). Therefore, it is desirable to use as few destination trees as possible in order to reduce the cost of receivers and transmitters. For this reason, we propose to minimize the total number of destination trees required while fulfilling the given traffic demands and all the network constraints. We formulate this problem in Section III and solve it in Section IV. After determining the destination trees, we realize HC-TWIN using the original node architecture and transmission protocol of TWIN.

B. Characteristics of HC-TWIN

- 1) *Advantages:* HC-TWIN has several advantages:
 - 1.1) HC-TWIN retains the advantages of TWIN. First, HC-TWIN emulates fast optical switching via fast tunable lasers at the network edge, reducing the complexity and cost of the network core. Second, HC-TWIN can

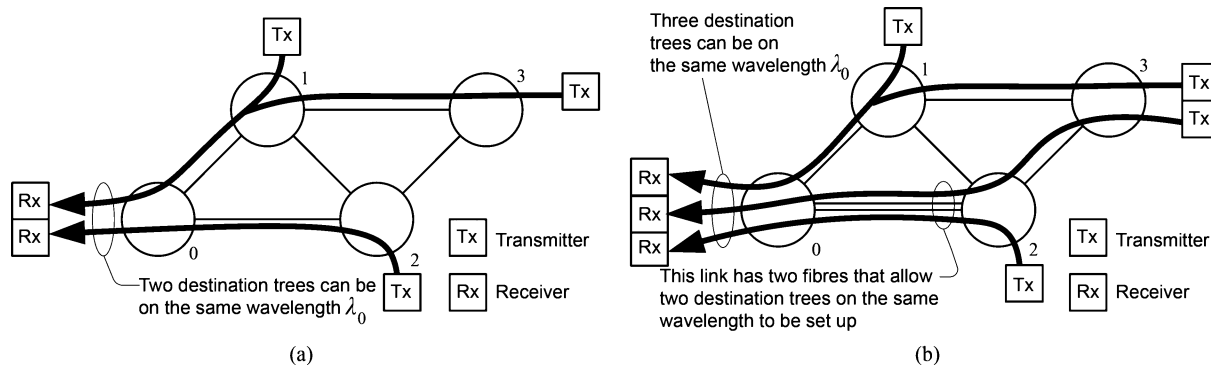


Fig. 2. Illustration of spatial reuse in HC-TWIN. (a) Two destination trees without common edges can use the same wavelength. (b) When there is one additional fiber between node 0 and node 2, three destination trees can use the same wavelength.

accommodate a certain degree of changing traffic patterns via bandwidth sharing. In both TWIN and HC-TWIN, certain source nodes share the data rate C provided by a destination tree, so their data rates can vary as long as their total data rate does not exceed C . Fig. 3(a) and (b) shows two examples to illustrate this bandwidth sharing feature of TWIN and HC-TWIN.

1.2) HC-TWIN provides large network capacity, which can be increased by using a larger degree of spatial reuse without increasing the number of wavelengths per fiber W or the channel rate C .

1.3) HC-TWIN can support more than W nodes while TWIN can support at most W nodes.¹ [1]. It is because HC-TWIN can provide more than W destination trees via spatial reuse for more than W nodes.

2) *Cost and Overhead:* HC-TWIN involves the following cost and overhead:

2.1) HC-TWIN needs additional transmitters and receivers and this involves additional cost. When a node has a large traffic demand, it needs multiple channels (destination trees) for concurrent transmission/reception and hence it needs multiple transmitters/receivers. However, it is unavoidable to use multiple transmitters/receivers for any network design whenever a node needs multiple channels for concurrent transmission/reception of large traffic volume. Nevertheless, HC-TWIN may reduce the cost of these transmitters/receivers by adopting lower channel rates and more destination trees via a larger degree of spatial reuse.

2.2) When HC-TWIN is set up or reconfigured, it is necessary to measure the traffic demand for each node pair and determine the destination trees (see Sections III–IV). This involves overhead. We believe that this overhead is mild for the following reasons.

- Traffic measurement is usually done in daily network management. In this case, we can make use of the

¹TWIN can support at most W nodes because the network can provide W destination trees at distinct wavelengths for W respective nodes. A generalized TWIN [6] adds a simple gateway function to each node such that the network uses multihop transmission and wavelength reuse to support more than W nodes. In the generalized TWIN, the per-node capacity is upper bounded by C because each node handles its own traffic as well as transit traffic, so the network capacity is upper bounded by NC . For the special case where $N \approx W^2$ and the traffic is uniform, a tighter upper bound on the network capacity is given by $0.5NC \approx 0.5W^2C$.

existing measurement data without extra measurement overhead.

- The destination trees can be determined quickly by the proposed three-stage algorithm (see Section IV).
- Network reconfiguration is needed only when the traffic patterns change significantly (say, once per few weeks or months, depending on the rate of change of traffic patterns). It is because HC-TWIN can already accommodate a certain degree of changing traffic patterns via bandwidth sharing [see item 1.1 in the above discussion and the example shown in Fig. 3(b)]. In reconfiguration, it is only necessary to control the wavelength selective switches in every node to change their input-to-output connections. The overhead is not large.

3) *Tradeoff Between Cost and Flexibility of Bandwidth Sharing:* If a source node uses more transmitters for concurrent transmission through destination trees, it is more costly but it can share the bandwidth of more destination trees at any time (i.e., larger flexibility of bandwidth sharing). Therefore, there is a tradeoff between cost and flexibility of bandwidth sharing. Fig. 3(b) and (c) shows two examples to illustrate this tradeoff. In HC-TWIN, each source node uses the fewest transmitters while it can still realize a certain degree of bandwidth sharing [see item 1.1 in the above discussion and the example shown in Fig. 3(b)], so it achieves a favorable tradeoff between cost and flexibility of bandwidth sharing.

4) *Applicability of HC-TWIN:* HC-TWIN includes the original TWIN design at one extreme (when each node uses one destination tree for reception from all other nodes) and the lightpath-based design at the other extreme (when each node uses one or more dedicated lightpaths for reception from every other node). We remind that TWIN is cost-effective for relatively light traffic load (the total traffic load from all nodes to any one node is smaller than or equal to the channel rate C) while the lightpath-based design is suitable for relatively heavy traffic load (the traffic load from one node to another node is a multiple of the channel rate C). HC-TWIN, being an interim between TWIN and the lightpath-based design, is cost-effective and suitable when the traffic load is also an interim (i.e., larger than that supported by TWIN but smaller than that supported by the lightpath-based design).

5) *Cost-Effectiveness Under High Traffic Demand:* HC-TWIN can cost-effectively handle high pairwise demand. Specifically, when a node pair has large traffic demand,

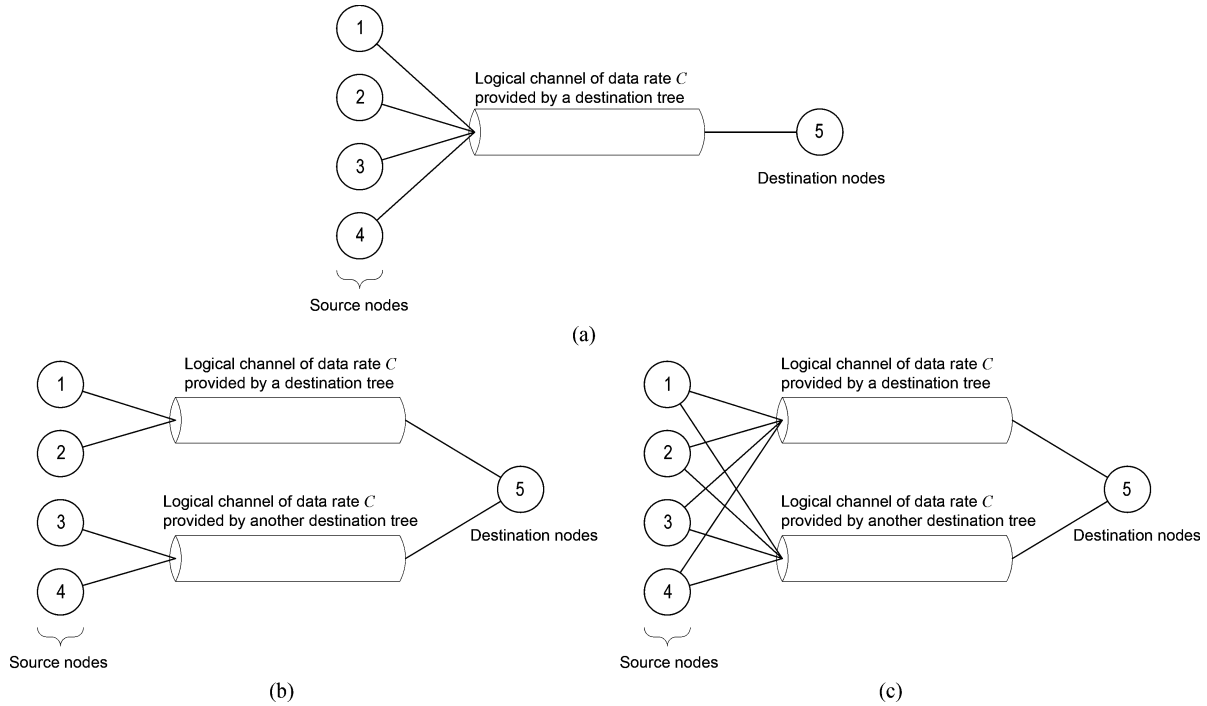


Fig. 3. Examples to illustrate the bandwidth sharing feature of TWIN and HC-TWIN. In these examples, the network has five nodes and only the destination trees to node 5 are shown for clarity. (a) In this example of TWIN, all the four source nodes share the data rate C provided by a destination tree to node 5. (b) In this example of HC-TWIN, two source nodes (nodes 1 and 2) share the data rate C provided by a destination tree to node 5 and the other two source nodes (nodes 3 and 4) share the data rate C provided by another destination tree to node 5. (c) This example shows the tradeoff between cost and flexibility of bandwidth sharing in HC-TWIN. If every source node uses two transmitters for simultaneous transmission through two destination trees, then all the source nodes can completely share the total available data rate $2C$ (i.e., largest flexibility of bandwidth sharing) but they require more transmitters (i.e., more costly).

HC-TWIN would assign some dedicated source-to-destination channels and one or a few destination trees to this node pair (see the three-stage algorithm in Section IV). For example, if a node pair needs a capacity of $3.5C$, this node pair would be assigned three dedicated channels (providing a capacity of $3C$) and one destination tree [providing a capacity of $0.5C$ to this node pair while the remaining capacity $0.5C$ is shared by other node(s)]. The dedicated channels are used via plain circuit switching, while the destination tree is shared via fast tunable lasers at the network edge without using fast switching within the network core (this nice property is inherited from the original TWIN).

III. DESTINATION TREE CONSTRUCTION: PROBLEM DEFINITION

In this section, we formulate the problem of determining the destination trees on a given physical network to fulfill the given traffic demands. Our objective is to minimize the total number of destination trees required (see the motivation discussed in Section II-A). This problem is called *destination tree construction* (DTC). We prove that the DTC problem is NP-hard.

A. Network Model and Problem Formulation

We model the network as a directed graph $G(\mathbf{V}, \mathbf{E})$ where \mathbf{V} is a set of nodes and \mathbf{E} is a set of edges (links). Each link has one or more fibers, each fiber provides W wavelength channels, and each wavelength channel is operated at a data rate of C .

We denote the number of nodes of the network by N (i.e., $N = |\mathbf{V}|$). Let $\mathbf{D} = [d_{i,j}]_{N \times N}$ be a given *traffic demand matrix*, where $d_{i,j}$ is the data rate required from node i to node

j . $d_{i,j}$ can be obtained by traffic measurement which is usually done in daily network management. We define the following notations.

$\mathbf{\Gamma}$	Set of all the destination trees to be determined.
$\mathbf{\Gamma}_j$	Subset of $\mathbf{\Gamma}$ in which the destination trees have destination node j .
$\mathbf{\Gamma}_{\lambda_k}$	Subset of $\mathbf{\Gamma}$ in which the destination trees are assigned the wavelength λ_k .
\mathbf{V}_t	Set of nodes in destination tree t .
F_e	Number of fibers in edge e .
$B(t, i)$	Allocated data rate for node i that transmits through destination tree t .
$O(t, e)$	If destination tree t contains edge e , $O(t, e) = 1$; otherwise, $O(t, e) = 0$.

The DTC problem is to minimize the total number of destination trees required (i.e., minimize $|\mathbf{\Gamma}|$; see the motivation discussed in Section II-A) while fulfilling the following constraints.

- 1) The given traffic demands are accommodated (i.e., $\sum_{t \in \mathbf{\Gamma}_j} B(t, i) = d_{i,j}$ for all $1 \leq i, j \leq N$).
- 2) The aggregated data rate of all source nodes in each destination tree cannot exceed the channel rate (i.e., $\sum_{i=1}^N B(t, i) \leq C$ for all $t \in \mathbf{\Gamma}$).
- 3) For each edge e , at most F_e channels with the same wavelength can be used to construct the destination trees (i.e., $\sum_{t \in \mathbf{\Gamma}_{\lambda_k}} O(t, e) = F_e$ for all $e \in \mathbf{E}$ and $1 \leq k \leq W$).

- 4) Only the nodes in destination tree t can transmit through this destination tree (i.e., $B(t, i) = 0$ for all $i \notin \mathbf{V}_t$ and $t \in \Gamma$).

Mathematically, the DTC problem is formulated as follows:

$$\begin{aligned} & \text{Minimize} && |\Gamma| \\ & \text{Subject to} && 1. \sum_{t \in \Gamma_j} B(t, i) = d_{i,j} \text{ for all } 1 \leq i, j \leq N \\ & && 2. \sum_{i=1}^N B(t, i) \leq C \text{ for all } t \in \Gamma \\ & && 3. \sum_{t \in \Gamma_{\lambda_k}} O(t, e) \leq F_e \text{ for all } e \in \mathbf{E} \text{ and } 1 \leq k \leq W \\ & && 4. B(t, i) = 0 \text{ for all } i \notin \mathbf{V}_t \text{ and } t \in \Gamma. \end{aligned}$$

B. Complexity

We prove that the DTC problem is NP-hard as follows. We consider the special case in which all elements in the traffic demand matrix are multiples of the channel rate C . Then each source node requires multiple and dedicated destination trees to each destination node, so each of these destination trees contains only one source node and hence it is a lightpath. The total number of destination trees (lightpaths) required becomes a constant, and the resulting problem reduces to the problem of determining the necessary lightpaths. The latter problem is the traditional *routing and wavelength assignment* (RWA) problem [14], which is NP-complete [15]. Therefore, the DTC problem is NP-hard.

IV. THREE-STAGE ALGORITHM FOR DESTINATION TREE CONSTRUCTION

In this section, we propose an efficient three-stage heuristic algorithm to solve the DTC problem. For convenience of description, we represent each destination tree t by $(j, \mathbf{S}_t, \mathbf{E}_t, \omega_t)$, where j represents destination node j , \mathbf{S}_t is a set of source nodes in t for destination node j , \mathbf{E}_t is a set of edges in t , and ω_t is the wavelength assigned to t .

A. Main Ideas

The proposed algorithm has three stages. The goal and the main ideas of each stage are as follows.

1) *Tree Construction*: In this stage, we determine as few destination trees as possible to fulfill the given traffic demands. This is done in three main steps. First, for each destination node, we divide the source nodes into as few groups as possible such that the source nodes in each group can share a destination tree (i.e., their total data rate is at most C). This is done by solving a generalized version of the *bin packing problem* [16]. Second, for each destination node and each group of source nodes, we determine a tree to connect these nodes such that this tree has as few edges as possible. This is done by solving a *Steiner tree problem* [17]. Third, we assign a wavelength to each of these destination trees. If we cannot assign wavelengths to all the trees, we proceed to the second stage.

2) *Tree Reconstruction*: In this stage, we repeatedly rearrange the existing destination trees to reduce the total number of edges of these trees, so that we may be able to assign wavelengths to all the trees. If we cannot still assign wavelengths to all the trees, we proceed to the third stage.

3) *Tree Addition and Reconstruction*: In this stage, we incrementally add a few destination trees in order to increase the flexibility of wavelength assignment and repeatedly rearrange the trees in order to assign wavelengths to all the trees.

Using the above framework, if the first stage cannot assign wavelengths to all the destination trees, the algorithm proceeds to the second stage to better optimize the destination trees. If the second stage cannot assign wavelengths to all the destination trees, the algorithm proceeds to the third stage to further optimize the destination trees.

B. First Stage: Tree Construction

In this stage, we determine as few destination trees as possible based on the main ideas described in Section IV-A.

For each $d_{i,j} \geq C$, we first construct $\lfloor d_{i,j}/C \rfloor$ destination trees that have one source node i and one destination node j using any shortest path routing algorithm. Each of these destination trees is used to handle a traffic demand of C from node i to node j . Let \mathbf{P} be a set of these destination trees. After excluding the traffic demands handled by these destination trees, the resulting demand matrix is $\mathbf{D}' = [d'_{i,j}]_{N \times N}$, where $d'_{i,j} = d_{i,j} - C \lfloor d_{i,j}/C \rfloor$. Then we determine the other necessary destination trees based on \mathbf{D}' for the rest of the algorithm.

We first divide the source nodes into as few groups as possible such that the source nodes in each group share a destination tree (i.e., their total data rate is at most C). For this purpose, we first solve a bin packing problem for each destination node j as follows. The source nodes, the traffic demands to node j (i.e., $d'_{i,j}$) and the channel rate C are treated as items, item size and bin capacity of the bin-packing problem, respectively. We apply a bin packing algorithm (e.g., the first-fit descending algorithm [16]) to solve this bin packing problem so that we divide the source nodes for destination node j into as few groups as possible. After executing the bin packing algorithm, each source node is assigned to one group. Since a source node can transmit data through more than one destination tree via its tunable laser, it can be assigned to more than one group. To exploit this flexibility, we iteratively reassign source nodes to more than one group in order to further reduce the total number of groups (i.e., total number of destination trees).

For each destination node j and each group of source nodes, we determine a destination tree to connect these source nodes to destination node j such that this tree has as few edges as possible. This is done by solving a Steiner tree problem [17] for these nodes (e.g., using the minimal spanning tree (MST) Steiner heuristic [17]).

We then assign wavelengths to the destination trees one after the other in the following order: the tree with more edges is assigned a wavelength first.

The procedure of the first stage is as follows.

PROCEDURE TREE_CONSTRUCTION

- 1 **Construct some destination trees.**

- 1.1 For each $d_{i,j} \geq C$, construct $\lfloor d_{i,j}/C \rfloor$ destination trees with source node i and destination node j using any shortest path routing algorithm.
- 1.2 Set $\mathbf{D}' = [d'_{i,j}]_{N \times N}$, where $d'_{i,j} = d_{i,j} - C \lfloor d_{i,j}/C \rfloor$.
- 2 **Construct other necessary destination trees to accommodate all traffic demands.**
 For EACH destination node j DO
 BEGIN
 - 2.1 Execute a bin packing algorithm (e.g., the first-fit descending algorithm [16]) to divide the source nodes into as few groups as possible such that the source nodes in each group need a total data rate of at most C .
 - 2.2 Reduce the number of groups iteratively as follows.
 - 2.2.1 Select group g such that it has the smallest total traffic demand.
 - 2.2.2 Select source node s from group g such that this node has the largest traffic demand.
 - 2.2.3 Excluding group g , select another group g' such that it has the smallest total traffic demand.
 - 2.2.4 Reallocate as much traffic demand of source node s as possible from group g to group g' .
 - 2.2.5 Repeat steps 2.2.2–2.2.4 until no further reallocation in step 2.2.4 is possible. If the resulting group g does not contain any source node, delete it.
 - 2.2.6 Repeat steps 2.2.1–2.2.5 until no further reallocation is possible.
 - 2.3 Determine a destination tree to connect the source nodes in each group and destination node j by executing a Steiner tree algorithm (e.g., MST Steiner heuristic [17]).
 END
- 3 **Assign wavelengths to all destination trees.**
 - 3.1 Sort all the destination trees in descending order of the number of edges required.
 - 3.2 Assign a wavelength of the lowest possible index to each of the destination trees in the above order.

C. Second Stage: Tree Reconstruction

In this stage, we repeatedly rearrange the existing destination trees to reduce the total number of edges of these trees so that we may be able to assign wavelengths to all the trees.

To decrease the total number of edges required, we use a sub-procedure that operates on two destination trees having the same destination node and decreases the number of edges of these trees by repeatedly 1) relocating a source node from one destination tree to another and 2) swapping a pair of source nodes between two destination trees. For each destination node j , we repeatedly perform this sub-procedure for each pair of destination trees in $\Gamma_j \setminus \mathbf{P}$ until we cannot further decrease the total number of edges of these trees. Note that a *possible* relocation/swapping means that each resulting destination tree t after relocation/swapping can still accommodate the traffic demands of its source nodes (i.e., $\sum_{i \in S_t} d'_{i,j} \leq C$ for all $t \in \Gamma_j \setminus \mathbf{P}$).

The details of the sub-procedure are as follows.

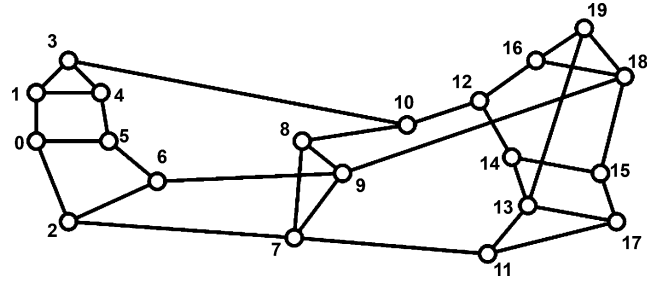


Fig. 4. Network adopted in computer simulation; it is the original Arpanet with 20 nodes and 32 bidirectional links.

SUB-PROCEDURE DECREASE_EDGES

1 **Iterative relocation of source nodes.**

1.1 For each source node in the two given destination trees, if it is relocated from its original destination tree to the other destination tree, determine the number of edges in the resulting two trees. Perform a relocation such that the number of edges in the resulting two trees is the smallest.

1.2 Repeat step 1.1 until no further relocation is possible.

2 **Iterative swapping of source nodes.**

2.1 Search for all possible swapping of source node pairs. Perform a swapping that can decrease the total number of edges by the largest amount.

2.2 Repeat step 2.1 until no further swapping is possible.

3 **Delete the obsolete destination trees.**

Delete the destination trees that have no source node.

The procedure of the second stage is as follows.

PROCEDURE TREE_RECONSTRUCTION

1 **Rearrange the destination trees.**

FOR EACH destination node j DO

BEGIN

1.1 For each pair of destination trees in $\Gamma_j \setminus \mathbf{P}$, execute **SUB-PROCEDURE DECREASE_EDGES**.

1.2 Repeat 1.1 until the total number of edges of the destination trees cannot be further decreased.

END

2 **Reassign wavelengths to all destination trees.**

2.1 Sort all the destination trees in Γ in descending order of the number of edges.

2.2 Assign a wavelength of the lowest possible index to each of the destination trees in the above order.

D. Third Stage: Tree Addition and Reconstruction

In this stage, we incrementally add destination trees in order to increase the flexibility of wavelength assignment and iteratively rearrange the trees in order to assign wavelengths to all the trees. This is done repeatedly until we find a feasible solution or no further change of the destination trees is possible.

For each destination node j , we split the destination tree with the largest number of edges in order to add a destination tree. Then we rearrange the destination trees in $\Gamma_j \setminus \mathbf{P}$ to decrease the total number of edges required (the sub-procedure for this purpose is described in the next paragraph). After rearrangement, if the total number of edges of the destination trees cannot be

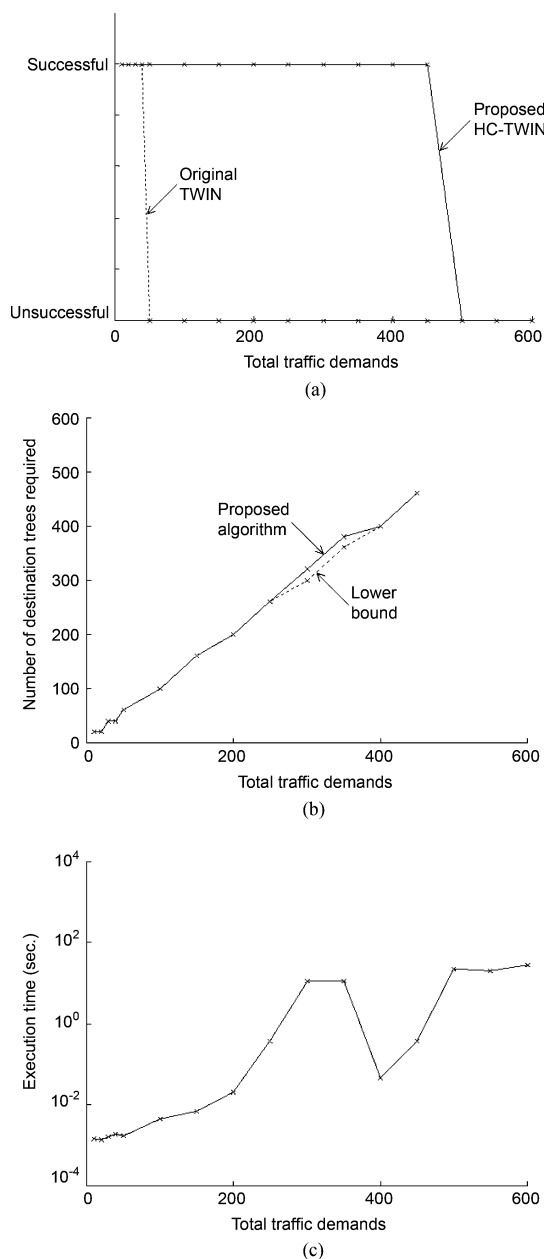


Fig. 5. Effectiveness of HC-TWIN and the three-stage algorithm for single-fiber links under uniform traffic. “Successful” means that the network can accommodate the given traffic demand; and “unsuccessful” otherwise. (a) Effectiveness of HC-TWIN. HC-TWIN can accommodate significantly larger traffic demand than TWIN. (b) Effectiveness of the three-stage algorithm. This algorithm can effectively reduce the number of destination trees required (i.e., reduce the numbers of transmitters and receivers required). (c) Execution time required on a computer using an Intel Pentium 2.80-GHz processor.

decreased, we restore the destination trees to the original ones (i.e., the ones before adding destination trees).

To rearrange the destination trees in $\Gamma_j \setminus \mathbf{P}$, we require a sub-procedure that is identical to DECREASE_EDGES (used in the second stage) except that it keeps a constant number of destination trees. The details of this sub-procedure are as follows.

SUB-PROCEDURE DECREASE_EDGES_ONLY

1 Iterative relocation of source nodes.

1.1 Search for all possible relocations of a source node from one destination tree t to another tree t' . Perform a relocation that can decrease the total number of edges

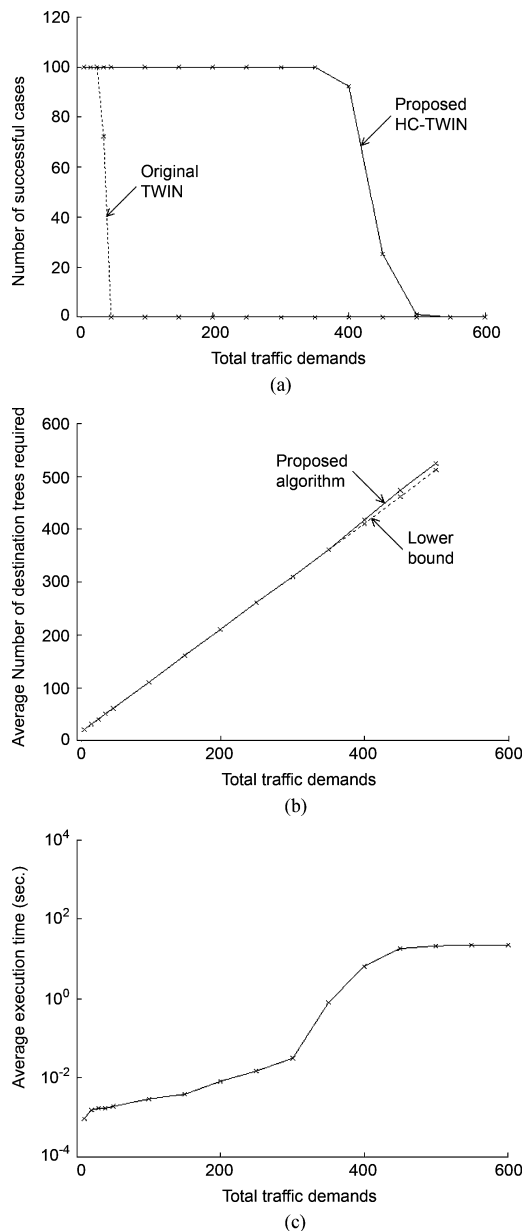


Fig. 6. Effectiveness of HC-TWIN and the three-stage algorithm for single-fiber links under nonuniform traffic. Out of 100 nonuniform traffic patterns for each value of the total traffic demand, we record the number of successful cases in which feasible destination trees are found to accommodate the given traffic patterns. (a) Effectiveness of HC-TWIN. (b) Effectiveness of the three-stage algorithm. (c) Execution time required on a computer using an Intel Pentium 2.80-GHz processor.

by the largest amount and the relocated node is not the only source node in t .

1.2 Repeat step 1.1 until no further relocation is possible.

2 Iterative swapping of source nodes.

2.1 Search for all possible swapping of source node pairs. Perform a swapping that can decrease the total number of edges by the largest amount.

2.2 Repeat step 2.1 until no further swapping is possible.

The procedure of the third stage is as follows.

PROCEDURE TREE_ADDITION_RECONSTRUCTION

1 Add and reconstruct destination trees.

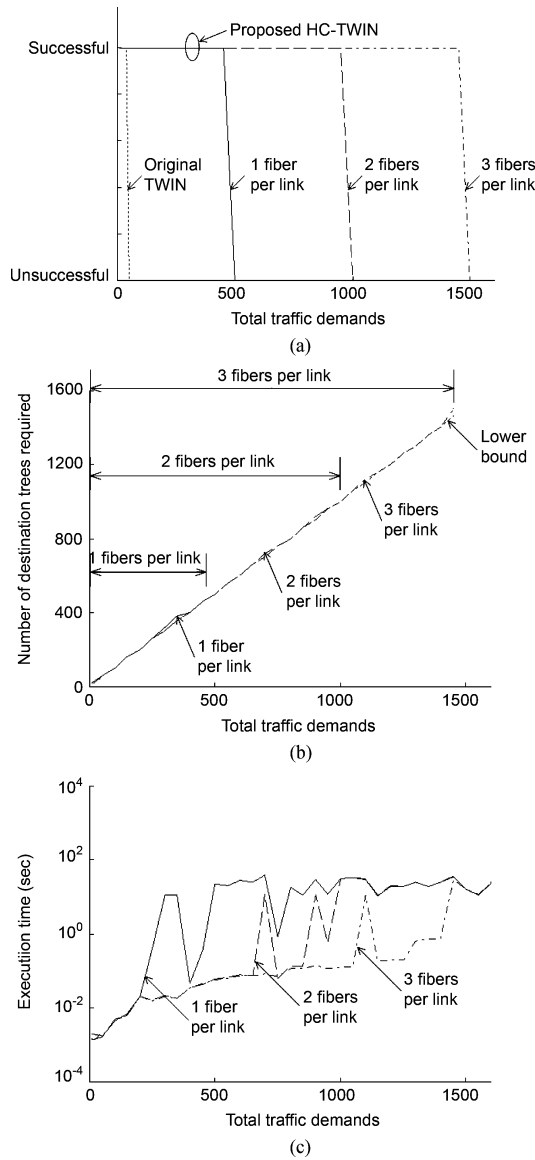


Fig. 7. Effectiveness of HC-TWIN and the three-stage algorithm for multi-fiber links under uniform traffic. “Successful” means that the network can accommodate the given traffic demand; and “unsuccessful” otherwise. (a) Effectiveness of HC-TWIN. (b) Effectiveness of the three-stage algorithm. (c) Execution time required on a computer using an Intel Pentium 2.80-GHz processor.

FOR EACH destination node j DO
BEGIN

- 1.1 Add a destination tree as follows. Among the destination trees with multiple source nodes, select the destination tree t that contains the largest number of edges. Add a new destination tree t' that has destination node j and a source node that is relocated from t .
- 1.2 Reconstruct the destination trees as follows.
 - 1.2.1 For each pair of destination trees in $\Gamma_j \setminus \mathbf{P}$, execute **SUB-PROCEDURE DECREASE_EDGES_ONLY**.
 - 1.2.2 Repeat step 1.2.1 until the total number of edges of all the destination trees cannot be further decreased.

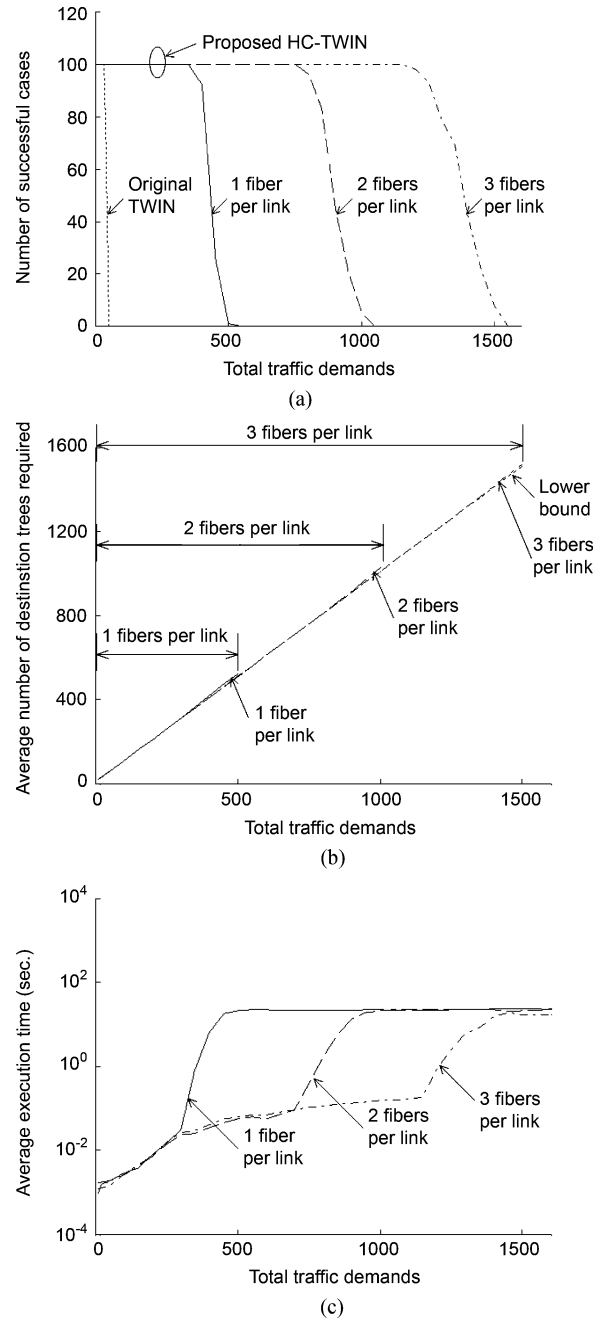


Fig. 8. Effectiveness of HC-TWIN and the three-stage algorithm for multi-fiber links under nonuniform traffic. Out of 100 nonuniform traffic patterns for each value of the total traffic demand, we record the number of successful cases in which feasible destination trees are found to accommodate the given traffic patterns. (a) Effectiveness of HC-TWIN. (b) Effectiveness of the three-stage algorithm. (c) Execution time required on a computer using an Intel Pentium 2.80-GHz processor.

- 1.3 If the total number of edges of all the destination trees has not been decreased, restore $\Gamma_j \setminus \mathbf{P}$ to the original one (i.e., the set before adding the destination tree t' in step 1.1).
- END

2 Reassign wavelengths to all destination trees.

- 2.1 Sort all the destination trees in Γ in descending order of the number of edges.

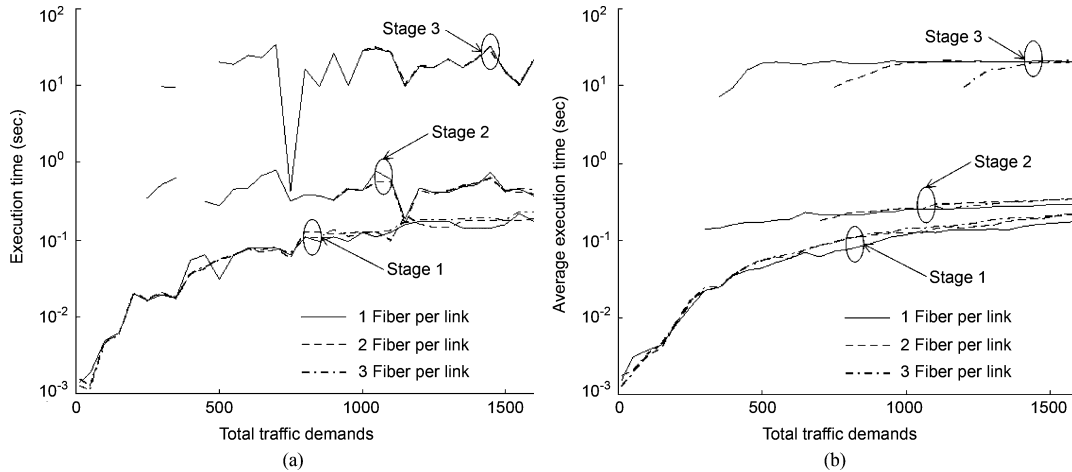


Fig. 9. Average execution time required by each stage of the proposed algorithm on a computer using an Intel Pentium 2.80-GHz processor. (a) Uniform traffic. (b) Nonuniform traffic.

2.2 Assign a wavelength of the lowest possible index to each of the destination trees in the above order.

3 Repeat

Repeat steps 1 and 2 until wavelengths have been assigned to all the destination trees or step 1 cannot further change the destination trees.

E. Three-Stage Algorithm

Overall, the three-stage algorithm is as follows.

ALGORITHM THREE-STAGE

1 First stage.

Determine as few destination trees as possible to accommodate the given traffic demands as follows.

1.1 1 Execute **PROCEDURE TREE_CONSTRUCTION**.

1.2 If wavelengths have been assigned to all the destination trees, stop.

2 Second stage.

Reconstruct the destination trees iteratively as follows.

2.1 Execute **PROCEDURE TREE_RECONSTRUCTION**.

2.2 If wavelengths have been assigned to all the destination trees, stop.

3 Third stage.

Incrementally add some destination trees and iteratively rearrange all the destination trees in $\Gamma_j \setminus \mathbf{P}$ by executing **PROCEDURE TREE_ADDITION_RECONSTRUCTION**.

V. SIMULATION AND RESULTS

We perform computer simulation to evaluate the effectiveness of HC-TWIN and the three-stage heuristic algorithm.

A. Simulation Model and Performance Measures

We adopt the Arpanet's topology shown in Fig. 4. There are 20 nodes and 32 bi-directional links. We consider single-fiber links as well as multi-fiber links. Each fiber provides 50 wavelengths and each wavelength channel is operated at a rate of one capacity unit (e.g., one capacity unit is 10 Gbps). We consider

both uniform and nonuniform traffic. For uniform traffic, $d_{i,j}$ is constant for all $i \neq j$ and $d_{i,j} = 0$ for all $i = j$. For nonuniform traffic, we randomly generate 100 traffic demand matrices for each value of the total traffic demand. Specifically, we generate each traffic demand matrix as follows. Let Z be a given total traffic demand. We randomly generate $a_{i,j}$ for all $i \neq j$ based on a uniform distribution over the range $[0, 1]$, then compute $A \equiv \sum_{i \neq j} a_{i,j}$, and get $d_{i,j} = a_{i,j} \times Z/A$ for all $i \neq j$ and $d_{i,j} = 0$ for all $i = j$. The proposed algorithm is executed for each traffic demand pattern generated.

We evaluate the effectiveness of HC-TWIN and the three-stage heuristic algorithm through the following measures.

1) *Lower Bound On the Minimal Number of Destination Trees Required:* We remind that the problem of minimizing the number of destination trees is NP-hard. To evaluate the solution quality of the three-stage heuristic algorithm, we compare the number of destination trees determined by this algorithm with a lower bound on the minimal number of destination trees required. This lower bound is derived as follows. For destination node j , it receives a total traffic load of $\sum_{1 \leq i \leq N, i \neq j} d_{i,j}$ from all other nodes, so it needs at least $\lceil \sum_{1 \leq i \leq N, i \neq j} d_{i,j} / C \rceil$ destination trees. Overall, the network requires at least $\sum_{1 \leq j \leq N} \lceil \sum_{1 \leq i \leq N, i \neq j} d_{i,j} / C \rceil$ destination trees and this quantity is the lower bound.

2) *Successful/Unsuccessful:* To evaluate the capacity of a given network, we progressively increase the total traffic demand until the network cannot accommodate it. If the network can accommodate a given total traffic demand, this case is called "successful"; otherwise, it is called "unsuccessful."

3) *Execution Time:* We measure the execution time required on a computer with an Intel Pentium 2.80-GHz processor.

B. Results

We first consider HC-TWIN with single-fiber links under uniform traffic. Fig. 5 shows the effectiveness of HC-TWIN and the three-stage algorithm. Fig. 5(a) shows that HC-TWIN can accommodate significantly larger traffic demands than TWIN on the same physical network. Specifically, HC-TWIN can accommodate up to 450 capacity units (remind that one capacity unit

TABLE I
 PERCENTAGE OF CASES IN WHICH: I) ONLY STAGE 1 IS EXECUTED, II) ONLY STAGES 1 AND STAGES 2 ARE EXECUTED, AND III) ALL THE THREE STAGES ARE EXECUTED. SIMULATION SETTING IS THE SAME AS THAT FOR FIG. 8

Total traffic demands	Percentage of cases in which (%)								
	only stage 1 is executed			only stages 1 and 2 are executed			stages 1, 2 and 3 are executed		
	1 fiber per link	2 fibers per link	3 fibers per link	1 fiber per link	2 fibers per link	3 fibers per link	1 fiber per link	2 fibers per link	3 fibers per link
100	100	100	100	0	0	0	0	0	0
200	100	100	100	0	0	0	0	0	0
300	96	100	100	4	0	0	0	0	0
400	8	100	100	40	0	0	52	0	0
500	0	100	100	0	0	0	100	0	0
600	0	100	100	0	0	0	100	0	0
700	0	98	100	0	2	0	100	0	0
800	0	76	100	0	16	0	100	8	0
900	0	11	100	0	20	0	100	69	0
1000	0	0	100	0	0	0	100	100	0
1100	0	0	98	0	0	2	100	100	0
1200	0	0	87	0	0	10	100	100	3
1300	0	0	59	0	0	13	100	100	28
1400	0	0	17	0	0	11	100	100	72
1500	0	0	0	0	0	2	100	100	98

corresponds to the data rate of a wavelength channel such as 10 Gbps). In other words, HC-TWIN can provide the necessary destination trees even when the total traffic demand is as high as 450 capacity units. In contrast, TWIN can only accommodate up to 40 capacity units.² While HC-TWIN can provide larger capacity, it needs more transmitters and receivers because more destination trees are used. Fig. 5(b) shows that the three-stage algorithm can effectively minimize the number of destination trees required. The number of destination trees determined by this algorithm is either equal to the lower bound (i.e., it is optimal) or close to the lower bound (i.e., it is close-to-optimal). Fig. 5(c) shows that the three-stage algorithm is fast. The execution time required ranges from several mini-seconds to less than about half a minute. Since destination trees are determined in network design or reconfiguration phases, the three-stage algorithm is fast enough for practical applications.

We consider HC-TWIN with single-fiber links under nonuniform traffic. Fig. 6 shows the effectiveness of HC-TWIN and the three-stage algorithm. Since we generate 100 nonuniform traffic patterns (i.e., 100 randomly generated traffic patterns) for each value of the total traffic demand, we record the number of successful cases in which feasible destination trees are found to accommodate the given traffic patterns. Fig. 6(a)–(c) shows that HC-TWIN and the three-stage algorithm are still very effective under nonuniform traffic. In particular, Fig. 6(a) shows that HC-TWIN can accommodate significantly larger traffic demands than TWIN on the same physical network, Fig. 6(b) shows that the three-stage algorithm can find optimal

²This phenomenon (i.e., TWIN can accommodate up to 40 capacity units while the fiber provides 50 wavelengths) can be explained as follows. We remind that the network has 20 nodes (see Fig. 4). Under uniform traffic, every node in TWIN requires the same number of destination trees. When the total traffic demand is small, every node requires one destination tree so that the network provides 20 destination trees to accommodate 20 capacity units. When the total traffic demand becomes larger, every node requires two destination trees so that the network provides 40 destination trees to accommodate 40 capacity units. When the total traffic demand is further increased, every node requires three destination trees but the network cannot provide 60 destination trees because there are only 50 wavelengths. Therefore, TWIN can only accommodate up to 40 capacity units.

or close-to-optimal solutions, and Fig. 6(c) shows that the three-stage algorithm is fast enough for practical applications.

We now consider HC-TWIN with multi-fiber links under both uniform and nonuniform traffic. Figs. 7 and 8 show the effectiveness of HC-TWIN and the three-stage algorithm. We see that HC-TWIN can accommodate even larger traffic demands by using more fibers per link. It is because HC-TWIN can realize a larger degree of spatial reuse on multi-fiber links to provide more destination trees to accommodate larger traffic demands. With this flexibility, HC-TWIN can provide large enough capacity to fulfill the ever-increasing traffic demands by using more fibers per link.

We now study the characteristics of the three-stage algorithm. Table I shows the percentage of cases in which: 1) only stage 1 is executed; 2) only stages 1 and 2 are executed; and 3) all the three stages are executed. We observe that, as the traffic demand increases, the three-stage algorithm would likely proceed to stage 2 or even stage 3. It is because more destination trees are needed under larger traffic demand, so it is more difficult to optimize the destination trees and hence the proposed algorithm would likely proceed to stages 2 and 3 in order to better optimize these trees. Fig. 9 shows the average execution time required by each stage. We see that stage 1 is the fastest, stage 3 is the slowest, and the total execution time required is fast enough for practical deployment. When the network is large with 100 nodes (its topology is randomly constructed by the method in [18] and [19], and each link has one fiber) and the total traffic demand is up to 20 000 capacity units such that all the three stages are executed, the execution time required is up to 4000 s. Therefore, even for these large networks, the execution time is still manageable for practical deployment (as destination trees are determined in network design or reconfiguration phases and reconfiguration is typically done once per few weeks or months).

VI. CONCLUSION

We investigated how to exploit and optimize wavelength reuse so that TWIN can be upgraded to provide larger capacity

in a resource-effective way. We formulated this optimization problem, proved its NP-hardness, and designed an efficient three-stage heuristic algorithm to solve it. We demonstrated that 1) the resulting network, called high-capacity TWIN (HC-TWIN), can better utilize the available network resources to provide larger network capacity while retaining the appealing advantages of TWIN, 2) the network capacity can be increased by realizing larger degree of wavelength reuse, and 3) the three-stage algorithm can find optimal or close-to-optimal solutions.

REFERENCES

- [1] I. Widjaja, I. Saniee, R. Giles, and D. Mitra, "Light core and intelligent edge for a flexible, thin-layered, and cost-effective optical transport network," *IEEE Commun. Mag.*, vol. 41, no. 5, pp. 2–9, May 2003.
- [2] K. Ross, N. Bambos, K. Kumaran, I. Saniee, and I. Widjaja, "Scheduling bursts in time-domain wavelength interleaved networks," *IEEE J. Select. Areas Commun.*, vol. 21, no. 11, pp. 1441–1451, Nov. 2003.
- [3] K. Ross, N. Bambos, K. Kumaran, I. Saniee, and I. Widjaja, "Dynamic scheduling of optical data bursts in time-domain wavelength interleaved networks," in *Proc. HOTI*, Aug. 2003.
- [4] A. K. Somani, M. Mina, and L. Li, "On trading wavelengths with fibers: A cost-performance based study," *IEEE/ACM Trans. Netw.*, vol. 12, no. 5, pp. 944–950, Oct. 2004.
- [5] E. B. Desurvire, "Capacity demand and technology challenges for lightwave systems in the next two decades (invited paper)," *IEEE J. Lightw. Technol.*, vol. 24, no. 12, pp. 4697–4710, Dec. 2006.
- [6] C. Nuzman and I. Widjaja, "Time-domain wavelength interleaved networking with wavelength reuse," in *Proc. IEEE INFOCOM 2006*, Apr. 23–29, 2006, pp. 1–12.
- [7] A. Gladisch, R. P. Braun, D. Breuer, A. Ehrhardt, H. M. Foisel, M. Jaeger, R. Leppla, M. Schneiders, S. Vorbeck, W. Weiershausen, and F. J. Westphal, "Evolution of terrestrial optical system and core network architecture," *Proc. IEEE*, vol. 94, no. 5, pp. 869–891, May 2006.
- [8] H. Suzuki, M. Fujiwara, and K. Iwatski, "Application of super-DWDM technologies to terrestrial terabit transmission systems," *IEEE J. Lightw. Technol.*, vol. 24, no. 5, pp. 1998–2005, May 2006.
- [9] S. Guizani, H. Hamam, Y. Bouslimani, and A. Cheriti, "High bit rate optical communications: Limitations and perspectives," *IEEE Canadian Rev.*, no. 50, pp. 11–15, 2005.
- [10] A. Scavennec and O. Leclerc, "Toward high-speed 40-Gbit/s transponders," *Proc. IEEE*, vol. 94, no. 5, pp. 986–996, May 2006.
- [11] A. K. Somani, M. Mina, and L. Li, "On trading wavelengths with fibers: A cost-performance based study," *IEEE/ACM Trans. Netw.*, vol. 12, no. 5, pp. 944–951, Oct. 2004.
- [12] A. Mokhtar and M. Azizoglu, "Adaptive wavelength routing in all-optical networks," *IEEE/ACM Trans. Netw.*, vol. 6, no. 2, pp. 197–206, Apr. 1998.
- [13] F. B. Shepherd and A. Vetta, "Lighting fibers in a dark network," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 11, pp. 1583–1588, Nov. 2004.
- [14] R. Ramaswami and K. N. Sivarajan, "Routing and wavelength assignment in all-optical networks," *IEEE/ACM Trans. Netw.*, vol. 3, no. 5, pp. 489–499, Oct. 1995.
- [15] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high-bandwidth optical WANs," *IEEE Trans. Commun.*, vol. 40, no. 7, pp. 1171–1182, Jul. 1992.
- [16] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman and Company, 1979.
- [17] F. K. Hwang, D. S. Richards, and P. Winter, *The Steiner Tree Problem*. New York: Elsevier, 1992.
- [18] B. M. Waxman, "Routing of multiple connections," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.
- [19] I. Kaj and R. Gaigalas, Waxman Random Network Topology Generator, Dec. 2005 [Online]. Available: <http://www.math.uu.se/research/telecom/software/stgraphs.html>



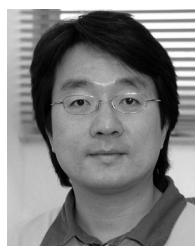
Tony K. C. Chan received the B.Sc. degree from the City University of Hong Kong in 1991, the M.Sc. degree from the Hong Kong Polytechnic University in 1997, and the Ph.D. degree from the Hong Kong Baptist University in 2005.

From 1993–2002, he worked in the IT industry in Hong Kong. From 2005–2007, he was a Research Fellow, conducting research on networking and multimedia systems. He is currently a System Analyst in a listed company in Hong Kong.



Eric W. M. Wong (S'87–M'90–SM'00) received the B.Sc. and M.Phil. degrees in electronic engineering from the Chinese University of Hong Kong, Hong Kong, in 1988 and 1990, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Massachusetts, Amherst, in 1994.

In 1994, he joined the City University of Hong Kong, where he is now an Associate Professor with the Department of Electronic Engineering. His current research interests are analysis and design of telecommunications networks, optical switching, and video-on-demand systems. His most notable research work involved the first accurate and workable model for state-dependent dynamic routing. Since 1991, the model has been used by AT&T to design and dimension its telephone network that uses real-time network routing.



Yiu-Wing Leung (M'92–SM'96) received the B.Sc. and Ph.D. degrees from the Chinese University of Hong Kong in 1989 and 1992, respectively. His Ph.D. advisor was Prof. P. T. S. Yum.

He is currently a Professor in the Department of Computer Science of the Hong Kong Baptist University, Hong Kong. He has been working on two main research areas: 1) networking and multimedia, and 2) cybernetics and systems engineering. He has published over 60 journal papers in these areas.