On The Complexity of Virtual Topology Design for Multicasting in WDM Trees with Tap-and-Continue and Multicast-Capable Switches*

E. Miller R. Libeskind-Hadas† D. Barnard‡ W. Chang
K. Dresner§ W. M. Turner¶ J. R. Hartline‖

Department of Computer Science
Harvey Mudd College
Claremont, California 91711

Abstract

This paper investigates the problem of finding optimal multicast virtual topologies, with respect to minimizing the maximum hop distance, in WDM multicast trees. Although the problem of finding optimal multicast trees is itself known to be NP-complete under many optimization metrics, high-quality approximation algorithms are known for this problem. We investigate the case that a multicast tree has been selected and seek to embed an optimal virtual topology in this multicast tree. We show that the problem can be solved in polynomial time when tap-and-continue switches are employed which allow a light-path to be tapped by some number of intermediate nodes. However, the problem becomes NP-complete when fully multicast-capable switches are employed. Our results suggest that tap-and-continue switches can be used to obtain high quality multicast virtual topologies while heuristics will be required to find good solutions in fully multicast-capable networks.

1 Introduction

Wavelength-division multiplexing (WDM) is a key optical networking technology for realizing low cost, high bandwidth, and scalable data services. Each fiber optical physical link in a WDM network is partitioned into multiple data channels each of which operates on a separate wavelength. Thus, WDM permits use of enormous fiber bandwidth by providing

*This work was supported by the National Science Foundation under grants CCR-9900491 and ANI-0207754.

†Communicating author.

‡D. Barnard is now with the Department of Physics, University of California Los Angeles.

§K. Dresner is now with the Department of Computer Science, University of Texas at Austin.

¶W. M. Turner is now with the Department of Computer Science, University of Pennsylvania.

‖J. R. Hartline is now with the Department of Computer Science, Cornell University.

data channels whose individual bandwidths more closely match those of the electronic devices at their endpoints [20]. As WDM technology matures, it is likely to be widely used in systems ranging from local and metropolitan area networks to the backbone of the Next Generation Internet.

Many applications require efficient support for multicast communication in which a single source node in a network must send the same data to a number of destination nodes. Typically, the message is transmitted along a set of physical channels which constitute a *multicast tree*. The root of the tree is the source node and the remaining nodes in the tree are either destination nodes or intermediate nodes used to reach the destination nodes. Multicast in WDM networks has been studied actively and has been proposed for applications such as streaming audio and video [12], HDTV programming [14, 15, 21], and optical storage area networks [14].

Ideally, each message is transmitted from the source to a destination without any optical-to-electronic conversion within the network. Such *all-optical* communication can be realized by using a single wavelength in the multicast tree to establish a connection to each destination, but in general it may not be possible to find a single wavelength which is available on every physical link in the tree. Alternatively, all-optical wavelength converters may be used to convert from one wavelength to another within the network but such converters are likely to be prohibitively expensive for most applications in the foreseeable future [20].

A second approach is to use *multi-hop* routing in which a path from the source to a destination may comprise multiple subpaths which may use different wavelengths. In this approach a set of *light-paths* or *light-trees* are embedded in the multicast tree. A light-path is a path comprising channels on a single wavelength. The message is transmitted at the origin of the light-path on a particular wavelength. Within a light-path, transmission is entirely optical. At the terminus of a light-path the data is converted into electronic form and is delivered to the local node if it is a destination node. In addition, the data may be retransmitted at this node on one or more light-paths to reach other destination nodes. Intermediate nodes on a light-path allow the data to pass through optically, but do not necessarily access the data themselves. Thus, a single light-path from node x to z passing through intermediate node y is different from two light-paths, one from x to y and one from y to z, even if these two light-paths use the same wavelength. Similarly, light-trees [16] are a generalization of light-paths in which the source of the light-tree transmits the data on a particular wavelength and this may be split in a tree-like fashion to reach multiple nodes. The collection of light-paths or light-trees is called the *virtual topology*.

While light-trees appear to be more desirable than light-paths, light-trees require *fully multicast-capable* switches which can "split" an incoming wavelength to multiple output ports. Such switches are inherently complex and also incur a reduction in signal strength known as *splitting loss*. Although splitting loss can be partially mitigated by the introduction of amplifiers, this further increases the complexity and cost of the switches. Ali and Deogun [1] therefore proposed an alternative to fully multicast-capable switches, called *tap-and-continue* switches, in which data proceeds strictly along a path but intermediate nodes on the path may access the data themselves by tapping a small fraction of the signal. Ali and Deogun show that such switches can be implemented with only a very modest addition in hardware complexity. This mechanism substantially reduces the splitting loss in comparison to a fully multicast-capable switch but still has an inherent limit on the number of times that a signal can be tapped before it loses integrity. Tap-and-continue mechanisms have also been discussed in [21].

Multicasting in WDM networks has received considerable attention [1, 3, 4, 7, 9, 10, 11, 13, 16, 17, 19]. Many of the existing results employ heuristics or integer linear programming (which is itself NP-complete). Since the problem of constructing a multicast tree is known to be computationally intractable under many objective functions, tree construction and virtual topology construction are often decoupled; first a multicast tree is found and then a virtual topology is constructed in this tree [4, 10, 18, 21]. For example, in much of the literature a cost is associated with each physical link and the primary consideration in tree construction is to minimize the cost of the selected physical links for the multicast tree. This is the NP-complete Steiner tree problem. Other multicast tree construction problems consider factors such as splitting loss and attenuation and these problems are known to be NP-complete as well [19]. Nonetheless, heuristics, approximation algorithms, and other approaches have been proposed for these and related tree construction problems [4, 8, 16, 19, 21].

In this paper we assume that an existing technique has been employed to construct a multicast tree. We show that for a particular optimization objective, optimal multicast virtual topologies can be found in a tree in polynomial time when tap-and-continue switches are employed but that the analogous problem is NP-complete when fully multicast-capable switches are used. This result demonstrates that there are algorithmic advantages to using the simpler tap-and-continue switches over the more complex multicast-capable switches. It should be noted that our results do not suggest that fully multicast-capable switches are not useful, but rather that we should not expect to make optimal use of such resources.

In particular, in this paper we examine the objective of minimizing the maximum hop distance in a multicast virtual topology. In a virtual topology, the *hop distance* from node s to node d is defined to be the minimum number of light-paths over all paths from s to d. Each hop in a path incurs latency due to conversion of the data from the optical to the electronic domain, application of the routing function, queueing, and retransmission on the next light-path. Thus, minimizing the maximum hop distance is a performance metric of interest in geographically small networks in which hop latency can dominate channel propagation latency [2]. We note that there are many other possible objective functions that one might wish to optimize in a virtual topology. The techniques proposed here are likely to be of use for a variety of optimization criteria. A network designer contemplating the use of fully multicast-capable switches and some heuristic to use these switches could use our algorithmic approach to determine how much benefit this would provide over using the simpler tap-and-continue switches with an optimal virtual topology algorithm.

The remainder of this paper is organized as follows: In Section 2 we formally define the problems studied in this paper. In Section 3 we describe several fundamental results leading to a polynomial-time algorithm for constructing optimal virtual topologies under the tap-and-continue model. We also give experimental results based on an implementation of this algorithm. In Section 4 we show that the corresponding problem in networks employing fully multicast-capable switches is NP-complete. We conclude in Section 5.

2 Problem Statement

Let a multicast tree $T = (V, E)$ be a directed tree with root $r \in V$ such that every edge is directed from a parent vertex to a child vertex. For simplicity we assume that every vertex in V other than r is a destination vertex.[1] Let $\mathcal{W}$ denote the number of wavelengths available on each edge.

Definition 1 *A* basic virtual topology *for a tree $T = (V, E)$ is a pair $\Gamma = (P, f)$ where P is a set of directed paths in T called* light-paths *and $f : P \to \{1, \ldots, \mathcal{W}\}$ is an assignment of light-paths to wavelengths such that:*

- *Any two light-paths in P which share a directed edge are assigned distinct wavelengths. That is, if $p, p' \in P$ and $\exists e \in E$ such that $e \in p \cap p'$ then $f(p) \neq f(p')$.*
- *For each $v \in V - r$ there exists a set of light-paths $\{p_1, \ldots, p_k\} \subseteq P$ such that p_1 originates at r, p_k terminates at v, and for $1 \leq i \leq k-1$, the end vertex of p_i is the starting vertex of p_{i+1}.*

For example, consider the multicast tree T in Figure 1(a). A basic virtual topology for this tree using two wavelengths is shown in Figure 1(b) where light-paths using the first wavelength are indicated by solid edges and light-paths on the other wavelength are indicated by dashed edges.

Let $\mathcal{P}$ be a positive integer which models the amount of power used to transmit a message. A *tap-and-continue virtual topology* is a generalization of a basic virtual topology in which each light-path may deliver the message to up to $\mathcal{P}$ vertices on the path. In other words, up to $\mathcal{P}$ vertices may tap the message as it passes. This is formalized in the following definition where the notation 2^X denotes the power set of X.

Definition 2 *A* tap-and-continue virtual topology *for a tree $T = (V, E)$ is a triplet $\Gamma = (P, f, t)$ where P is a set of light-paths in T, $f : P \to \{1, \ldots, \mathcal{W}\}$ is an assignment of light-paths to wavelengths, and $t : P \to 2^{(V-r)}$ specifies which vertices tap each path such that:*

- *Any two light-paths in P which share a directed edge are assigned distinct wavelengths under function f.*
- *For each $p \in P$, the function t assigns at most $\mathcal{P}$ vertices on that light-path to tap the message. That is, for each path $p \in P$, $|t(p)| \leq \mathcal{P}$ and if $v \in t(p)$ then v is on path p.*
- *For each $v \in V - r$ there exists a set of light-paths $\{p_1, \ldots, p_k\} \subseteq P$ such that p_1 originates at r, $v \in t(p_k)$, and for $1 \leq i \leq k-1$, $\exists v \in t(p_i)$ such that v is the starting vertex of p_{i+1}.*

[1] Our results are easily extendible to the general case that only a subset of nodes in the multicast tree are destination nodes.

Note that when $\mathcal{P} = 1$, a tap-and-continue virtual topology degenerates into a basic virtual topology. At the other extreme, unlimited power can be modeled by letting $\mathcal{P}$ be equal to the height of the tree. For simplicity, we henceforth say that a vertex v *terminates* a light-path p if v taps p but no descendant of v taps p.

For example, consider again the multicast tree in Figure 1(a). A tap-and-continue virtual topology for this tree using two wavelengths and $\mathcal{P} = 4$ is shown if Figure 1(c). The solid and dashed lines indicate the two different wavelengths and the rectangles indicate intermediate vertices at which a light-path is tapped.

A *splitting virtual topology* is a further generalization of a basic virtual topology employing *light-trees* instead of light-paths where each light-tree may deliver the message to up to $\mathcal{P}$ of its vertices. This is formalized in the following definition.

Definition 3 A splitting virtual topology *for a tree* $T = (V, E)$ *is a triplet* $\Gamma = (P, f, t)$ *where* P *is a set of directed trees in* T *called light-trees,* $f : P \rightarrow \{1, \ldots, \mathcal{W}\}$ *is an assignment of light-trees to wavelengths, and* $t : P \rightarrow 2^{(V-r)}$ *specifies which vertices tap each light-tree such that:*

- *Any two light-trees in* P *which share a directed edge are assigned distinct wavelengths under function* f*.*
- *For each tree* $p \in P$*, the function* t *assigns at most* $\mathcal{P}$ *vertices on that light tree to tap the message. That is, for each* $p \in P$*,* $|t(p)| \leq \mathcal{P}$ *and if* $v \in t(p)$ *then* v *is in the tree* p*.*
- *For each* $v \in V - r$ *there exists a set of light-trees* $\{p_1, \ldots, p_k\} \subseteq P$ *such that* p_1 *originates at* r*,* $v \in t(p_k)$*, and for* $1 \leq i \leq k-1$*,* $\exists v \in t(p_i)$ *such that* v *is the starting vertex of* p_{i+1}*.*

For example, Figure 1(d) shows a splitting virtual topology using two wavelengths and $\mathcal{P} = 4$ for the multicast tree shown in Figure 1(a).

We use the term *virtual path* to indicate a path from the source to a destination vertex comprising one or more light-paths or light-trees. For simplicity and without loss of generality, we assume that in any virtual topology there exists exactly one virtual path to each destination. In other words, while there may be multiple light-paths or light-trees passing through a destination vertex, a destination vertex taps exactly one of these light-paths or light-trees. The *hop distance* of a destination vertex is the number of distinct light-paths or light-trees on the virtual path to that vertex. Thus, if the hop distance to a destination vertex is one then the connection from the source to that vertex is entirely optical. If the hop distance is larger than one then the message incurs retransmissions en route to the destination. The *maximum hop distance* for a virtual topology is the maximum hop distance to any destination vertex.

For example, in the basic virtual topology shown in Figure 1(b), the maximum hop distance is 3. In particular the hop distance to vertex 9 is 3: The virtual path consists of a light-path from vertex 1 to vertex 4, a light-path from vertex 4 to vertex 6, and a light-path from vertex 6 to vertex 9. It is easily verified that for this multicast tree and 2 wavelengths, any basic virtual topology must have maximum hop distance at least 3. In contrast, the tap-and-continue virtual topology shown in Figure 1(c) has maximum hop distance 2. In particular, vertices 6, 7, 8, and 9 all have hop distance 2. It is easily verified that for this multicast tree with 2 wavelengths and $\mathcal{P} = 4$, any tap-and-continue virtual topology must have maximum hop distance at least 2. Finally, the splitting virtual topology shown in Figure 1(d) has maximum hop distance 1; all vertices receive the message on a light-tree directly from the source vertex.

In general, we consider the problem of finding virtual topologies which minimize the maximum hop distance. We define the following three optimization problems.

Problem 1 (Basic Virtual Topology Optimization Problem) *Given a directed tree $T = (V, E)$ with root $r \in V$, a positive integer number of wavelengths $\mathcal{W}$ and a positive integer $\mathcal{H}$, construct a basic virtual topology with maximum hop distance at most $\mathcal{H}$ or determine that no such virtual topology exists.*

Problem 2 (Tap-and-Continue Virtual Topology Optimization Problem) *Given a directed tree $T = (V, E)$ with root $r \in V$, a positive integer number of wavelengths $\mathcal{W}$, a positive integer transmission power $\mathcal{P}$, and a positive integer $\mathcal{H}$, construct a tap-and-continue virtual topology with maximum hop distance at most $\mathcal{H}$ or determine that no such virtual topology exists.*

Problem 3 (Splitting Virtual Topology Optimization Problem) *Given a directed tree $T = (V, E)$ with root $r \in V$, a positive integer number of wavelengths $\mathcal{W}$, a positive integer transmission power $\mathcal{P}$, and a positive integer $\mathcal{H}$, construct a splitting virtual topology with maximum hop distance at most $\mathcal{H}$ or determine that no such virtual topology exists.*

Problem 1 was studied by Gerstel *et al.* [6] in the context of ATM networks and a linear time algorithm was found for this problem. In Section 3 we show that Problem 2 can also be solved in polynomial time, albeit with a substantially more complicated algorithm. As noted earlier, Problem 1 is a special case of Problem 2 in which the parameter $\mathcal{P}$ is set to 1. In Section 4 we show that Problem 3 is NP-complete.

3 The Tap-and-Continue Virtual Topology Optimization Problem

In this section we solve the splitting virtual topology optimization problem. For simplicity, throughout this section we use the term "virtual topology" to mean a "tap-and-continue virtual topology" using at most $\mathcal{W}$ wavelengths per edge, $\mathcal{P}$ power per light-path, and with maximum hop distance at most $\mathcal{H}$. In subsection 3.1 we state and prove a necessary and sufficient condition for the existence of a virtual topology in a given multicast tree. We use these results in subsection 3.2 to give a polynomial time algorithm for constructing a virtual topology or determining that none exists. In subsection 3.3 we describe experimental results obtained using this algorithm.

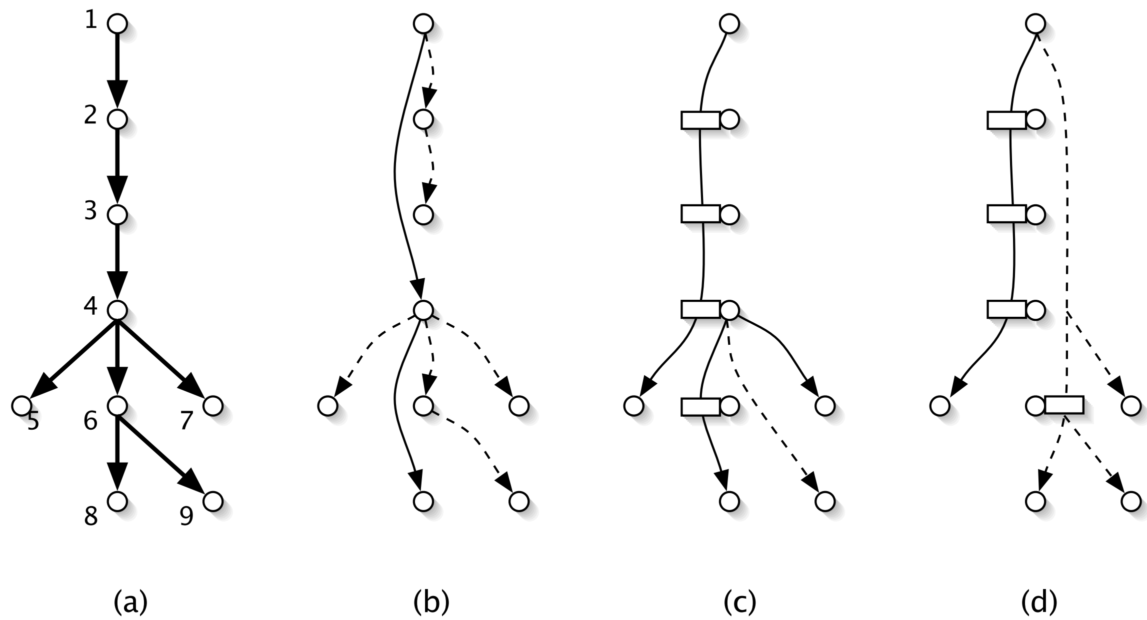

Figure 1: (a) A multicast tree. The source is vertex 1 and all other vertices are destinations. (b) A basic virtual topology with maximum hop distance 3. (c) A tap-and-continue virtual topology with $\mathcal{P} = 4$ and maximum hop distance 2. (d) A splitting virtual topology with $\mathcal{P} = 4$ and maximum hop distance 1. Rectangles indicate intermediate tap points on light-paths and light-trees.

3.1 A Necessary and Sufficient Condition

Consider a virtual topology, Γ, for tree T. When the root r transmits a message on a light-path, that light-path is said to have $\mathcal{H}$ hops remaining. If a destination vertex taps that light-path and retransmits a new light-path, the new light-path is said to have $\mathcal{H}-1$ hops remaining. In general, when a vertex v receives a message on a light-path with h hops remaining, any light-path transmitted by v will have $h-1$ hops remaining. Each light-path must have at least one hop remaining. Similarly, a light-path has some amount of power, k, remaining at each vertex v on the path. The light-path has power $\mathcal{P}$ remaining at the vertex at which it is transmitted. Let v be a vertex on the light-path and let $p(v)$ denote the parent of v. If the light-path has power k remaining at $p(v)$ then the light-path has power k remaining at v if $p(v)$ does not tap the light-path and has power $k-1$ remaining at v if $p(v)$ does tap the light-path. Each light-path must have at least one unit of power remaining at every vertex on the path.

Our definitions will be simplified by allowing a vertex in a virtual topology to "artificially reduce" the number of hops and amount of power remaining when transmitting a light-path. Thus, when a vertex v receives a message with $h > 1$ hops remaining, it may transmit a new light-path with any number of hops remaining between 1 and $h-1$ rather than strictly $h-1$. Similarly, v may transmit a message with any amount of power remaining between 1 and $\mathcal{P}$ rather than strictly $\mathcal{P}$. Clearly, if there exists a virtual topology which artificially reduces the number of hops and power remaining at some transmitting vertices, then there exists a virtual topology which transmits at the standard number of hops remaining and full power.

Throughout this section, $\mathcal{H} \times \mathcal{P}$ matrices will be used in our algorithm and proofs. Henceforth, we use the notation X_i to denote row i of matrix X and we use the notation $X_{(i,j)}$ to denote the entry at row i and column j.

Definition 4 *A matrix is* valid *if its dimensions are $\mathcal{H} \times \mathcal{P}$, all of its entries are non-negative, and the sum of its entries is at most $\mathcal{W}$.*

Definition 5 *Let T be a multicast tree and let v be a destination vertex in T. Let T_v be the subtree rooted at v with the addition of a source vertex r and an edge from r to v. Let X be a valid matrix. Vertex v is said to be* routable *by X if there exists a virtual topology Γ_v for T_v such that v receives exactly $X_{(i,j)}$ light-paths with i hops and j power remaining, $1 \leq i \leq \mathcal{H}$, $1 \leq j \leq \mathcal{P}$.*

Definition 6 *Let T be a multicast tree, let Γ be a virtual topology with respect to T, and let v be a destination vertex in T. Let the* supply matrix *for v with respect to Γ be the $\mathcal{H} \times \mathcal{P}$ matrix $s_\Gamma(v)$ such that $s_\Gamma(v)_{(i,j)}$ is the total number of light-paths entering v with i hops and j power remaining.*

Note that since Γ is a virtual topology then $s_\Gamma(v)$ is necessarily valid for each destination vertex v in the multicast tree.

We begin with a tree T and no virtual topology. Our objective is to determine whether or not a virtual topology exists for T. (Recall that our use of the term "virtual topology" means a tap-and-continue virtual topology using at most $\mathcal{W}$ wavelengths per edge, $\mathcal{P}$ power per light-path, and with maximum hop distance at most $\mathcal{H}$.) To this end, our algorithm associates a $\mathcal{H} \times \mathcal{P}$ matrix, called a *constraint matrix*, with each destination vertex in the tree. The constraint matrices for the destination vertices are computed according to a set of rules in a bottom-up fashion beginning at the leaves. We show that there exists a virtual topology if and only if all constraint matrices are valid. Moreover, if this condition is met, we show how to construct the actual virtual topology.

The following definitions provide the components used to compute the constraint matrices.

Definition 7 *Let $U_{(i,j)}$ denote the $\mathcal{H} \times \mathcal{P}$ matrix with all entries 0 except for the entry in row i and column j, which is 1.*

Definition 8 *An $\mathcal{H} \times \mathcal{P}$ constraint matrix is said to be* reducible *at row i if the first entry in row i is at least 1 and the sum of the first $\mathcal{P} - 1$ entries in row i is at least 2.*

Definition 9 *The* reduction function, *$R_i(X)$, is a function from $\mathcal{H} \times \mathcal{P}$ matrices to $\mathcal{H} \times \mathcal{P}$ matrices defined as follows. If X is not reducible at row i then $R_i(X) = X$. Otherwise, let $X' = X - U_{(i,1)}$. Let j be the smallest index such that $X'_{(i,j)} > 0$. (Note that j may be 1.) Then $R_i(X) = X' - U_{(i,j)} + U_{(i,j+1)}$.*

The reduction function is now used to define the following function which will be used to compute the constraint matrices.

Definition 10 *$M(X)$ is a function from $\mathcal{H} \times \mathcal{P}$ matrices to $\mathcal{H} \times \mathcal{P}$ matrices defined as follows. Let i be the smallest index for which X_i contains a non-zero entry. Let $X' = R_i(X)$. Let s be the sum of all entries in X'. If $i = \mathcal{H}$ or $s \leq \mathcal{W}$ then $M(X) = X'$. Otherwise let X'' be identical to X' except that row i of X'' is all 0s. Then $M(X) = M(X'' + U_{(i+1,1)})$.*

We now define the constraint matrices constructively.

Definition 11 *For a vertex v, let $C(v)$, the* constraint matrix *of v, be defined as follows. If v is a leaf, then $C(v) = U_{(1,1)}$. Otherwise, let the children of v be denoted by $v_1, v_2, \ldots v_k$. Then $C(v) = M(\sum_{i=1}^{k} C(v_i) + U_{(1,1)})$.*

For example, consider the virtual topology in Figure 1(a) with 2 wavelengths ($\mathcal{W} = 2$), 2 hops permitted ($\mathcal{H} = 2$), and power 4 per light-path ($\mathcal{P} = 4$). Then 2×4 constraint matrices will be associated with each vertex. The constraint matrices for the leaves, vertices 5, 7, 8, and 9, are

$$U_{(1,1)} = \begin{pmatrix} 1000 \\ 0000 \end{pmatrix}$$

These constraint matrices indicate that each of the leaves requires, at a minimum, a single entering light-path with 1 power and 1 hop remaining. By Definition 11, the constraint matrix for vertex 6 is

$$M\left(\begin{pmatrix} 2000 \\ 0000 \end{pmatrix} + \begin{pmatrix} 1000 \\ 0000 \end{pmatrix}\right) = M\left(\begin{pmatrix} 3000 \\ 0000 \end{pmatrix}\right) = \begin{pmatrix} 1100 \\ 0000 \end{pmatrix} \tag{1}$$

Intuitively, the two leftmost expressions in Equation 1 indicates that vertex 6 and its descendants, vertices 8 and 9, can be reached if vertex 6 receives 3 light-paths with 1 hop and 1 power remaining on each light-path. However, since there are only 2 wavelengths available, vertex 6 cannot receive 3 light-paths. Thus, the reduction function applies here and gives the constraint matrix on the right-hand side. This constraint matrix indicates that vertices 6, 8, and 9 can be reached if vertex 6 receives 1 light-path with 1 hop and 1 power remaining and 1 light-path with 1 hop and 2 power remaining.

Similarly, the constraint matrix for vertex 4 is, by Definition 11,

$$M\left(\left(\begin{array}{c}1100\\0000\end{array}\right)+\left(\begin{array}{c}1000\\0000\end{array}\right)+\left(\begin{array}{c}1000\\0000\end{array}\right)+\left(\begin{array}{c}1000\\0000\end{array}\right)\right)=M\left(\left(\begin{array}{c}4100\\0000\end{array}\right)\right)=\left(\begin{array}{c}0000\\1000\end{array}\right) \quad (2)$$

Intuitively, the two leftmost expressions in Equation 2 indicates that vertex 4 and all of its descendants can be reached if vertex 4 receives 4 light-paths with 1 hop and 1 power remaining and 1 light-path with 1 hop and 2 power remaining. However, since there are only 2 wavelengths available, vertex 4 cannot receive 5 light-paths. The reduction function is applied to yield the matrix on the right-hand side, indicating that vertex 4 and its descendants can be reached if it receives 1 light-path with 2 hops and 1 power remaining.

Continuing up the tree in Figure 1(a) we see that the constraint matrices for vertices 3, 2, and 1, as defined by Definition 11, are

$$\left(\begin{array}{c}1000\\1000\end{array}\right), \left(\begin{array}{c}0100\\1000\end{array}\right), \text{ and } \left(\begin{array}{c}0010\\1000\end{array}\right)$$

respectively. Notice that in this example, the constraint matrix computed by Definition 11 is valid for every vertex in the multicast tree (since every constraint matrix has the property that the sum of its entries does not exceed 2, the given number of wavelengths). The following lemma and theorem state that if every vertex has a valid constraint matrix then there exists a tap-and-continue multicast topology for the given wavelength, maximum hop distance and power constraints.

Lemma 1 *If v is a destination vertex and every vertex in the subtree rooted at v has a valid constraint matrix, then v is routable by $C(v)$.*

Proof: The proof is by strong induction on the height of the subtree rooted at v. For the basis, assume that the height of the multicast subtree rooted at v is 0. Then v is a leaf and its constraint matrix is therefore $C(v) = U_{(1,1)}$. Moreover, any single vertex is routable by $U_{(1,1)}$ since it requires only a single light-path with one hop and one power remaining.

For the induction step, assume that the claim is true if v is the root of a subtree of height less than n. Now, consider a vertex v whose subtree has height n and let $v_1, \ldots, v_k$ denote the children of v. For convenience, let $S(v) = \sum_{i=1}^{k} C(v_i)$. By Definition 11, $C(v) = M(S(v) + U_{(1,1)})$.

Let r denote the smallest index such that row r of $C(v)$ contains a non-zero entry. By Definition 10, $C(v)$ and $S(v)$ do not differ in rows $r+1$ through $\mathcal{H}$. We now consider two possible cases.

Case 1: $r = 1$. If $S(v) + U_{(1,1)}$ is not reducible at row 1 then by Definition 10, $C(v) = S(v) + U_{(1,1)}$. In this case, v taps and terminates a light-path with 1 hop and 1 power remaining. The remaining light-paths represented by $S(v)$ are distributed to the children of v such that child v_i receives supply vector $C(v_i)$. By the induction hypothesis, each v_i is routable by $C(v_i)$ and thus v is routable by $C(v)$. If $S(v)+U_{(1,1)}$ is reducible at row 1 then $C(v) = R_1(S(v) + U_{(1,1)})$. Therefore, by Definition 9, $C(v) = S(v) - U_{(1,j)} + U_{(1,j+1)}$ for some $j < \mathcal{P}$. In this case, vertex v taps a light-path with 1 hop and $j+1$ power remaining. This decreases the number of light-paths with 1 hop and $j+1$ power remaining by 1 and thus increases the number of light-paths with 1 hop and j power remaining by 1. Consequently, the remaining light-paths are represented by $S(v)$. Thus, the light-paths represented in $S(v)$ are distributed to the children of v as above and v is again routable by $C(v)$.

Case 2: $r > 1$. Let $\hat{S}(v)$ be identical to $S(v)$ except that all entries in rows 1 through $r-1$ are 0. By Definition 10, $C(v) = R_r(\hat{S}(v) + U_{(r,1)})$. If $\hat{S}(v) + U_{(r,1)}$ is not reducible at row r then $C(v) = \hat{S}(v)+U_{(r,1)}$. In this case, v taps and terminates a light-path with r hops and 1 power remaining. Vertex v forwards $C(v_i)_{(j,k)}$ light-paths with j hops and k power remaining to each child v_i, $1 \leq i \leq k$, $r \leq j \leq \mathcal{H}$, $1 \leq k \leq \mathcal{P}$. Thus, the total number of forwarded light-paths with j hops and k power remaining is exactly $\hat{S}(v)_{(j,k)}$, $r \leq j \leq \mathcal{H}$, $1 \leq k \leq \mathcal{P}$. In addition, v transmits $C(v_i)_{(j,k)}$ light-paths with j hops and k power remaining to each child v_i, $1 \leq i \leq k$, $1 \leq j < r$, $1 \leq k \leq \mathcal{P}$. Vertex v can transmit such light-paths since it taps a light-path with r hops remaining. Moreover, the total number of light-paths transmitted by v to any child v_i does not exceed $\mathcal{W}$ since each $C(v_i)$ is valid. The supply matrix for each v_i is now exactly $C(v_i)$ and thus, by the induction hypothesis, v is routable by $C(v)$. If $\hat{S}(v) + U_{(r,1)}$ is reducible at row r then $C(v) = \hat{S}(v) - U_{(r,j)} + U_{(r,j+1)}$ for some $1 \leq j < \mathcal{P}$. In this case, v taps a light-path with r hops and $j+1$ power remaining and the forwarding and transmitting of light-paths is otherwise exactly identical to the above. □

Theorem 1 *If every destination vertex v in the multicast tree T has a valid constraint matrix then there exists a virtual topology for T.*

Proof: Let r denote the root of the multicast tree and let $v_1, \ldots, v_k$ denote the children of r. For each i, $1 \leq i \leq k$, all vertices in the subtree rooted at v_i have valid constraint matrices by assumption. Therefore, by Lemma 1, v_i is routable by $C(v_i)$. Thus, vertex r simply transmits to each v_i the $\mathcal{W}$ or fewer light-paths specified in $C(v_i)$. □

Next, we prove that if there exists a virtual topology then every destination vertex has a valid constraint matrix.

Definition 12 *A virtual topology is said to be* frugal *if each light-path is transmitted at some power between* 1 *and* $\mathcal{P}$ *and each transmission incurs between* 1 *and* $\mathcal{H}-1$ *hops such that*

- *If vertex v is a leaf of the multicast tree then the light-path p entering v has exactly 1 hop remaining.*

- *If vertex v terminates a light-path p, then p has exactly 1 power remaining at v.*

Note that if there exists a virtual topology Γ for multicast tree T then Γ can be trivially modified to become a frugal virtual topology.

Lemma 2 *Let Γ be a frugal virtual topology and let vertex v terminate a light-path with i hops remaining. If $s_\Gamma(v)$ is reducible at row i then there exists a frugal virtual topology Γ' such that:*

- $s_{\Gamma'}(v) = R_i(s_\Gamma(v))$ *and*
- $s_{\Gamma'}(u) = s_\Gamma(u)$ *for all vertices u not on the path from the root to v.*

Proof: Let p_1 denote the light-path with i hops remaining which terminates at v. Since $s_\Gamma(v)$ is reducible at row i, by Definition 9 there exists at least one other light-path with i hops and strictly less than $\mathcal{P}$ power remaining which passes through vertex v without being tapped. Among all such light-paths, let p_2 denote one that enters v with the least power, $j < \mathcal{P}$, remaining. If light-path p_2 is not tapped by any ancestor of v then construct virtual topology Γ' to be identical to Γ except that p_2 is transmitted with one more unit of power, v taps p_2, and p_1 is terminated prior to entering v. All vertices that are not on the path from the root to v are unaffected by this transformation and $s_{\Gamma'}(v) = s_\Gamma(v) - U_{(i,1)} - U_{(i,j)} + U_{(i,j+1)}$ and therefore $s_{\Gamma'}(v) = R_i s_\Gamma(v)$.

The remaining case is that p_2 is tapped at an ancestor of v. Let vertex u denote the least ancestor of v which taps p_2. Note that p_1 may be transmitted by an ancestor of u or by some descendent of u above vertex v. However, p_1 cannot be transmitted by u since any light-path transmitted by u has at most $i-1$ hops remaining.

Case 1: p_1 is transmitted by an ancestor of u. Construct virtual topology Γ' to be identical to Γ except that p_1 is tapped at u and p_2 is tapped at v. All vertices that are not on the path from the root to v are unaffected by this transformation. Because light-path p_2 is not tapped at u in Γ', p_2 has one more unit of power remaining at v than it does in Γ. Light-path p_1 is terminated before arriving at v. Thus $s_{\Gamma'}(v) = s_\Gamma(v) - U_{(i,1)} - U_{(i,j)} + U_{(i,j+1)}$ and therefore $s_{\Gamma'}(v) = R_i(s_\Gamma(v))$.

Case 2: p_1 is transmitted at some vertex x below u but above v. Let p denote the amount of power remaining on light-path p_2 at x. Note that $p < \mathcal{P}$ since u tapped p_2. Construct virtual topology Γ' from Γ by partitioning p_2 into two light-paths: p_2' and p_2''. Light-path p_2' terminates at the last vertex above x at which p_2 was tapped and is transmitted with reduced power in order to ensure that the new virtual topology is frugal. Light-path p_2'' is transmitted by x with power $p+1$. Vertex v taps p_2'' rather than p_1 which is terminated prior to entering to v. All vertices that are not on the path from the root to v are unaffected by this transformation. Moreover, $s_{\Gamma'}(v) = s_\Gamma(v) - U_{(i,1)} - U_{(i,j)} + U_{(i,j+1)}$ and therefore $s_{\Gamma'}(v) = R_i(s_\Gamma(v))$. □

Lemma 3 *Let Γ be a frugal virtual topology and let vertex v tap but not terminate a light-path with i hops and ℓ power remaining at v. If there exists another light-path with i hops and $k < \ell$ power remaining at v then there exists a frugal virtual topology Γ' such that:*

- $s_{\Gamma'}(v) = R_i(s_\Gamma(v) + U_{(i,1)} + U_{(i,\ell-1)} - U_{(i,\ell)})$ *and*

- *$s_{\Gamma'}(u) = s_\Gamma(u)$ for all vertices u not on the path from the root to v.*

Proof: Let p_1 denote the light-path with i hops and ℓ power remaining tapped by v and let p_2 denote the light-path passing through v with i hops and the least power $k < \ell$ remaining. If light-path p_2 is not tapped by any ancestor of v then construct virtual topology Γ' to be identical to Γ except that p_2 is transmitted with one more unit of power, v taps p_2, and p_1 is transmitted with one less unit of power. All vertices that are not on the path from the root to v are unaffected by this transformation and $s_{\Gamma'}(v) = R_i(s_\Gamma(v) + U_{(i,1)} + U_{(i,\ell-1)} - U_{(i,\ell)})$ since path p_1 with i hops and ℓ power remaining is replaced by a path with i hops and $\ell - 1$ power remaining and tapping p_2 with one more unit of power corresponds to performing the reduction R_i. The remaining case is that p_2 is tapped at an ancestor of v. This case is handled identically to that in the proof of Lemma 2. □

We now define a recursive transformation τ on frugal virtual topologies.

Definition 13 *Let Γ be a frugal virtual topology and let v be a destination vertex. Transformation τ is performed with respect to Γ and v as follows:*

1. *If v is a leaf then let $\Gamma' = \Gamma$. Return Γ'.*

2. *If v is not a leaf then let $v_1, \ldots, v_k$ denote the children of v. Transformation τ is first applied recursively to each of the children of v in arbitrary order. Let Γ_1 denote the resulting virtual topology. Let p denote the light-path tapped or terminated by v and let h denote the number of hops remaining on light-path p. Let r denote the smallest index such that row r of $C(v)$ contains a non-zero entry. Let u denote the parent of v.*

3. *If $h = r$ then*

   (a) *Let Γ_2 be identical to Γ_1 except that all light-paths with fewer than h hops remaining at v are terminated prior to entering v. For each such light-path that previously passed through v, vertex v now transmits a new corresponding light-path with the same number of hops and power remaining. (Note that vertex v can transmit light-paths with $h-1$ hops and $\mathcal{P}$ power remaining, but hops and power are reduced as necessary to ensure that the supply matrices at each child of v remain unchanged from Γ_1 to Γ_2.)*

   (b) *If v terminates light-path p then*

      i. *If $s_{\Gamma_2}(v)$ is not reducible at row h then Γ' is identical to Γ_2.*

      ii. *If $s_{\Gamma_2}(v)$ is reducible at row h then Lemma 2 is used to construct a virtual topology Γ' such that $s_{\Gamma'}(v) = R_i(s_{\Gamma_2}(v))$ and $s_{\Gamma'}(u) = s_{\Gamma_2}(u)$ for all vertices u not on the path from u to v.*

   Else (v taps but does not terminate light-path p)

      i. *Let j denote the amount of power remaining on p at vertex v. If there exists a light-path with h hops and less than j power remaining which passes through v then Lemma 3 is used to construct a virtual topology Γ' such that $s_{\Gamma'}(v) = R_i(s_{\Gamma_2}(v) + U_{(i,1)} + U_{(i,\ell-1)} - U_{(i,\ell)})$ and $s_{\Gamma'}(u) = s_{\Gamma_2}(u)$ for all vertices u not on the path from the root to v.*

ii. Else Γ' is identical to Γ_2.

4. *Else if $h > r$*

   (a) *Vertex v no longer taps p.*

   (b) *If u is the root or u taps a light-path with at least h hops remaining then p is transmitted with one less power. Else (u is not the root and u taps a light-path with fewer than h hops remaining) let p' denote the light-path tapped by u. light-path p' is transmitted with one less power and u taps path p.*

   (c) *All light-paths with fewer than h hops remaining at v are terminated prior to entering v. If such a light-path previously passed through v with between r and $h-1$ hops remaining then u now transmits the corresponding light-path with the same number of hops and power remaining. If such a light-path previously passed through v with fewer than r hops remaining then denote the light path a* pending light-path. *The pending light-paths are considered in step 4(e) below.*

   (d) *If row r of $s_{\Gamma_1}(v)$ contains a non-zero entry in some position less than $\mathcal{P}$ then let j be the least such index. From step 4(c), this corresponds to a light-path transmitted by u with r hops and $j < \mathcal{P}$ power remaining. Vertex u now transmits this light-path with power $j+1$ and v taps this light-path. Else (row r of $s_{\Gamma_1}(v)$ contains 0s in all entries with position less than $\mathcal{P}$) vertex u transmits an additional light-path with r hops and 1 power remaining and v taps this light-path.*

   (e) *From step 4(d), vertex v taps a light-path with r hops remaining. Vertex v now transmits a new light-path in lieu of each pending light-path in step 4(c), with the same number of hops and power remaining as the pending light-path. The resulting virtual topology is Γ'.*

5. *Return Γ'.*

Note that the transformation τ described above has no provision for the case that $h < r$. It will be shown in the proof of the following lemma that this case cannot arise.

Lemma 4 *Let Γ be a frugal virtual topology for tree T and let v be a destination vertex. Transformation τ constructs a frugal virtual topology Γ' such that*

1. *$s_{\Gamma'}(u) = C(u)$ for each vertex u in the subtree rooted at v.*

2. *If w is on the path from the root to v then $s_{\Gamma'}(v)$ may differ from $s_{\Gamma}(v)$. For each vertex w not on the path from the root to v and not in the subtree rooted at v, $s_{\Gamma'}(w) = s_{\Gamma}(w)$.*

Proof: The proof is by strong induction on the height of the subtree rooted at v. If v is a leaf then Γ' is identical to Γ and thus the lemma is true. Assume that the claim is true for any vertex whose subtree height is less than n and let v be a vertex whose subtree height is $n > 0$. Let $v_1, \ldots, v_k$ denote the children of v. By the induction hypothesis, after applying transformation τ to each of the k children of v in step 2, the resulting virtual topology Γ_1

is a frugal virtual topology such that $s_{\Gamma_1}(u) = C(u)$ for each vertex u in the subtree rooted at v except possibly at vertex v itself. The second part of the lemma is also true by the induction hypothesis. We now consider separately the case that $h = r$ and the case that $h > r$ and show that in both cases the requirements of the lemma are satisfied. Afterwards, we show that the case $h < r$ cannot occur.

Henceforth, let $S(v) = \sum_{i=1}^{k} s_{\Gamma_1}(v_i)$. By the induction hypothesis $C(v) = M(S(v) + U_{(1,1)})$.

Case 1: $h = r$. Let $\hat{S}(v)$ denote the matrix identical to $S(v)$ except all entries in rows 1 through $h-1$ are 0. Since Γ_2 terminates all light-paths with fewer than h hops prior to entering v, $s_{\Gamma_2}(v)$ contains only 0s in rows 1 through $h-1$. Moreover, since v taps a message with h hops remaining in Γ_2, any light-path transmitted by v has at most $h-1$ hops remaining. Thus, $s_{\Gamma_2}(v)$ does not differ from $\hat{S}(v)$ in rows with indices higher than h.

Since h is the least index such that $C(v)$ contains a non-zero row, $\hat{S}(v)$ and $C(v)$ are identical in rows 1 through $h-1$. By Definition 10 and the fact that row h of $C(v)$ is not entirely 0s, $C(v)$ does not differ from $S(v)$ in rows $h+1$ through $\mathcal{H}$. Thus $C(v)$ can only differ from $\hat{S}(v)$, and thus $s_{\Gamma_2}(v)$, in row h.

Since Γ' is either identical to Γ_2 or results from using Lemma 2 or Lemma 3, $s_{\Gamma'}(v)$ and $s_{\Gamma_2}(v)$ can only differ in row h. Thus, $s_{\Gamma'}(v)$ and $C(v)$ can only differ at row h.

Consider the case that v terminates a light-path with h hops remaining in Γ_2. Then $s_{\Gamma_2}(v) = \hat{S}(v) + U_{(h,1)}$ where $U_{(h,1)}$ accounts for the light-path which v taps with h hops and 1 power remaining. If $s_{\Gamma_2}(v)$ is not reducible at row h then $s_{\Gamma'}(v) = s_{\Gamma_2}(v)$ and thus row h of $s_{\Gamma'}(v)$ is identical to row h of $\hat{S}(v) + U_{(h,1)}$. In this case, Definition 10 implies that row h of $C(v)$ is also identical to row h of $\hat{S}(v) + U_{(h,1)}$. If $s_{\Gamma_2}(v)$ is reducible at row h then Lemma 2 is used to construct Γ' where $s_{\Gamma'}(v) = R_h(\hat{S}(v) + U_{(h,1)})$. In this case, Definition 10 implies that $C(v)$ is also $R_h(\hat{S}(v) + U_{(h,1)})$.

Next, consider the case that v taps but does not terminate a light-path with h hops and power $j > 1$. Then $s_{\Gamma_2}(v) = \hat{S}(v) - U_{(h,j-1)} + U_{(h,j)}$ since some path with h hops and $j-1$ power remaining at v is now replaced with a path with h hops and j power remaining at v. If there exists a light-path with h hops and less than j power remaining which passes through v then Lemma 3 is used and $s_{\Gamma'}(v) = R_h(s_{\Gamma_2}(v) + U_{(h,1)} + U_{(h,j-1)} - U_{(h,j)}) = R_h(\hat{S}(v) + U_{(h,1)}) = C(v)$. If there does not exist a light-path with h hops and less than j power remaining which passes through v then $\Gamma' = \Gamma_2$. Thus, $s_{\Gamma'}(v) = \hat{S}(v) - U_{(h,j-1)} + U_{(h,j)}$. Therefore, $\hat{S}(v)_{(h,j-1)} \geq 1$ and $\hat{S}(v)_{(h,\ell)} = 0$ for $1 \leq \ell < j-1$. Consequently, $\hat{S}(v) + U_{(1,1)}$ is reducible at row h and by Definition 10, $C(v) = \hat{S}(v) - U_{(h,j-1)} + U_{(h,j)}$.

Thus, $s_{\Gamma'}(u) = C(u)$ for each vertex u in the subtree rooted at v. Moreover, each vertex w not on the path from the root to v, $s_{\Gamma'}(w) = s_{\Gamma}(w)$.

Case 2: $h > r$. Let $\overline{S}(v)$ denote the matrix identical to $S(v)$ except that all entries in rows 1 through $r-1$ are 0. From steps 4(c) and 4(e), rows 1 through $r-1$ of $s_{\Gamma'}(v)$ consist entirely of 0s and thus these rows are identical to the corresponding rows in $\overline{S}(v)$. From step 4(c), rows r through $h-1$ of $s_{\Gamma'}(v)$ do not differ from the corresponding rows of $\overline{S}(v)$. Moreover $s_{\Gamma'}(v)$ and $\overline{S}(v)$ cannot differ in rows with indices higher than h since v taps a light-path with h hops remaining and thus any light-path transmitted by v has at most $h-1$ hops remaining. Thus, $s_{\Gamma'}(v)$ can only differ from $\overline{S}(v)$ in row h due to the fact that v taps a light-path at row h in Γ_1. However, in step 4(a) and 4(e) the virtual topology Γ_1 is modified so that vertex v no longer taps a light-path with h hops remaining. Thus, $\overline{S}(v)$ and $s_{\Gamma'}(v)$ do not differ in row h and are thus identical in all rows.

By assumption, rows 1 through $r-1$ of $C(v)$ consist entirely of 0s. Moreover, by Definition 10 and 11, $C(v)$ does not differ from $S(v)$ in rows with indices higher than r. Thus, $C(v)$ can only differ from $\overline{S}(v)$, and thus $s_{\Gamma'}(v)$, in row r. If row r of $s_{\Gamma_1}(v)$ contains a non-zero entry in some position less than $\mathcal{P}$ and j is the least such index then $S(v)+U_{(r,1)}$ is reducible at row r and by Definition 10 row r of $C(v)$ is identical to row r of $s_{\Gamma'}(v)$ as constructed in step 4(d). Otherwise, $S(v)+U_{(r,1)}$ is not reducible at row r and again $C(v) = s_{\Gamma'}(v)$ by the construction of Γ' in step 4(d).

Finally, we must show that the construction in step 4 does not use more than $\mathcal{W}$ wavelengths on the edge from u to v. Assume by way of contradiction that the edge from u to v uses more than $\mathcal{W}$ wavelengths. Then $s_{\Gamma'}(v)$ is not valid and thus $C(v)$ is also not valid. However, by Definition 10, if $C(v)$ is not valid then $r = \mathcal{H}$. However, $h > r$ by assumption, which is a contradiction. Therefore, Γ' is a virtual topology and $s_{\Gamma'}(u) = C(u)$ for each vertex u in the subtree rooted at v. Moreover, each vertex w not on the path from the root to v, $s_{\Gamma'}(w) = s_{\Gamma}(w)$.

Case 3: $h < r$. We show that this case cannot occur. Assume by way of contradiction that $h < r$. Recall that $S(v) = \sum_{i=1}^{k} s_{\Gamma_1}(v_i)$ and $C(v) = M(S(v)+U_{(1,1)})$. Let σ denote the sum of the entries in rows $r-1$ through $\mathcal{H}$ of $S(v)$. Since row $r-1$ of $C(v)$ is entirely 0s, by Definition 10 $\sigma + 1 > \mathcal{W}$. Thus, in Γ_1 the total number of light-paths entering the children of v with $r-1$ or more hops remaining is at least $\sigma \geq \mathcal{W}$. Since $h < r$, any light-path transmitted by v may have at most $h-1 < r-1$ hops remaining. Therefore, at least σ light-paths with $r-1$ or more hops remaining must pass through v to its children. Clearly, $\sigma \leq \mathcal{W}$ since at most $\mathcal{W}$ light-paths may enter v and thus $\sigma = \mathcal{W}$. Therefore, v must tap, but not terminate, one of these $\mathcal{W}$ light-paths with at least $r-1$ hops remaining. By assumption, v taps a light-path with $h < r$ hops remaining which implies that v taps, but does not terminate, a light-path with exactly $r-1$ hops remaining. Now consider the computation of $C(v)$ according to Definition 10. Either the matrix was reducible at row $r-1$ or not. If it was not reducible, then row $r-1$ of $S(v)$ contains 0s at positions 1 through $\mathcal{P}-1$. Thus, v cannot tap a light-path with $r-1$ hops remaining since to do so would require a non-zero entry at some position between 1 and $\mathcal{P}-1$ in row $r-1$ of $S(v)$, a contradiction. If the matrix was reducible at row $r-1$ then performing the reduction leaves the first entry in row $r-1$ of $S(v)$ unchanged and then increments one entry and decrements another. Thus, the total sum of the entries in row $r-1$ of $S(v)$ is unchanged by the reduction. However, since this row contains only 0s in $C(v)$, by Definition 10 it must be that the sum of the entries in rows $r-1$ through $\mathcal{H}$ of $S(v)$ exceeds $\mathcal{W}$. Thus, more than $\mathcal{W}$ wavelengths pass through v, which is again a contradiction. □

Theorem 2 *If there exists a virtual topology* Γ *for tree* T *then every destination vertex has a valid constraint matrix.*

Proof: Without loss of generality, Γ is a frugal virtual topology. Let $v_1, \ldots, v_k$ denote the children of the root. Using Lemma 4 repeatedly at each v_i results in a virtual topology Γ' such that $s_{\Gamma'}(u) = C(u)$ for every destination vertex u. Since $s_{\Gamma'}(u)$ is valid, $C(u)$ is also valid, for each destination vertex u. □

Theorems 1 and 2 state that a tree T has a virtual topology if and only if the constraint matrices are valid at all destination vertices.

3.2 A Polynomial Time Algorithm

The results of the previous subsection immediately imply a polynomial time algorithm for determining whether a virtual topology exists in a given tree and, if so, constructing the actual virtual topology.

First, Definition 11 is applied to compute the constraint matrices for all destination vertices in a bottom-up manner. If every constraint matrix is valid, the virtual topology is constructed as suggested by Lemma 1. The algorithm is summarized below.

1. Compute the constraint matrix for each destination vertex v in the tree as specified in Definition 11. Also store $S(v)$, the sum of the constraint matrices of the children of v.

2. If any vertex has an invalid constraint matrix then no virtual topology exists and exit.

3. Else (all vertices have valid constraint matrices)

   (a) The root vertex transmits to each child v_i the set of light-paths specified in $C(v_i)$.

   (b) For each vertex v which receives a set of light-paths from its parent, let r denote the smallest index such that row r of $C(v)$ contains a non-zero entry.

      i. If $r = 1$ and $S(v) + U_{(1,1)}$ is not reducible at row 1 then v taps and terminates a light-path with 1 hop and 1 power remaining and distributes the rest of the incoming light-paths to its children so that each child receives the set of light-paths specified in its constraint matrix.

      ii. Else if $r = 1$ and $S(v) + U_{(1,1)}$ is reducible at row 1 then find j such that $C(v) = S(v) - U_{(1,j)} + U_{(1,j+1)}$. Vertex v taps a light path with 1 hop and $j + 1$ power remaining and distributes the rest of the incoming light-paths to its children so that each child receives the set of light-paths specified in its constraint matrix.

      iii. Else let $\hat{S}(v)$ be identical to $S(v)$ except that all entries in rows 1 through $r - 1$ are 0.

iv. If $r > 1$ and $\hat{S}(v) + U_{(r,1)}$ is not reducible at row r then v taps and terminates a light-path with r hops and 1 power remaining, forwards all light-paths with r or more hops remaining to its children as specified in their constraint matrices, and transmits light-paths with $r - 1$ or fewer hops remaining to its children as specified by their constraint matrices.

v. Else ($r > 1$ and $\hat{S}(v) + U_{(r,1)}$ is reducible at row r) find j such that $C(v) = S(v) - U_{(r,j)} + U_{(r,j+1)}$. Vertex v taps a light path with r hops and $j + 1$ power remaining and forwards and transmits light-paths as in the previous case.

The running time of this algorithm is easily verified to be $O(\mathcal{H} \times \mathcal{P} \times N)$. Since $\mathcal{H}$ and $\mathcal{P}$ need not exceed the height of the tree, in the worst case the running time of the algorithm is $O(N^3)$. However, if the tree is asymptotically height balanced, then the worst case running time is $O(N \log^2 N)$.

3.3 Experimental Results

The algorithm described above can be used to experimentally investigate a variety of attributes of virtual topologies for multicast. For example, Figure 2 summarizes experimental results in which 100 multicast trees were generated at random where each tree had height exactly 10, each vertex had between 1 and 3 children, and each edge had 5 wavelengths available. The multicast requests in these experiments were, in fact, broadcasts; all nodes in the tree were destination nodes. For maximum hop distances fixed at 2, 3, and 4, we examined the percentage of the randomly generated trees which have virtual topologies. On the horizontal axis, transmission power ranges from 1 unit (basic virtual topology) to 10 units (effectively "unlimited power", since the tree height is 10). The vertical axis indicates the percentage of cases in which a virtual topology could be found. These results suggest that increasing the maximum number of hops permitted can have a significant impact on the probability of the existence of a virtual topology. Moreover, while 2 units of initial power provide an advantage over 1 unit of power, increasing the power above 2 units has a very modest impact.

Figure 3 shows data from a second set of experiments using the same 100 multicast tree demonstrated in the first experiment. In these experiments, the initial power was set at 1unit (basic virtual topology) and 10 units (unlimited power). The maximum number of permitted hops is indicated on the horizontal axis and the percentage of instances with virtual topologies is again shown on the vertical axis. Note that as suggested by the previous data, initial power of 10 units is only slightly better than initial power of 2 units in the trees used in these experiments. The data for 2 units is not perceptibly different from the data for 10 units and is therefore not shown here. Collectively, these results suggest that with tap-and-continue switches a small amount of power suffices (2 units of power for the trees used in these experiments) and that this additional power has an impact for some maximum hop distance values. However, when the maximum hop distance constraint is sufficiently relaxed, tap-and-continue switches are no more effective than switches with no multicast functionality.

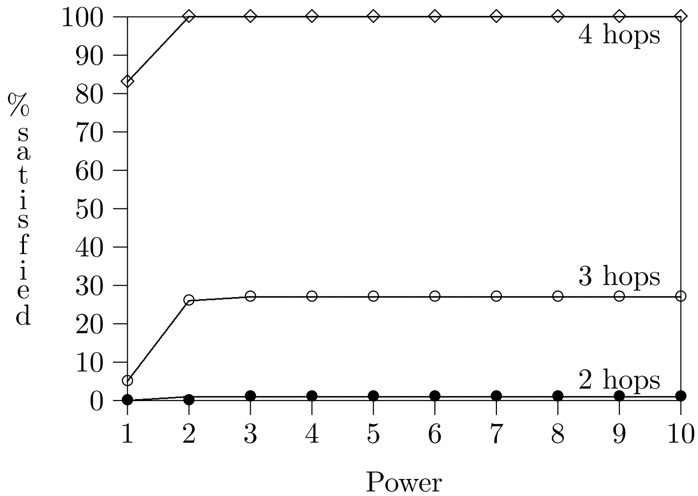

Figure 2: Percentage of randomly generated trees with virtual topologies as a function of transmission power when maximum hop distance is fixed to be 2, 3, or 4. All trees have height 10 and each vertex has between 1 and 3 children selected at random. Each edge has 5 wavelengths available.

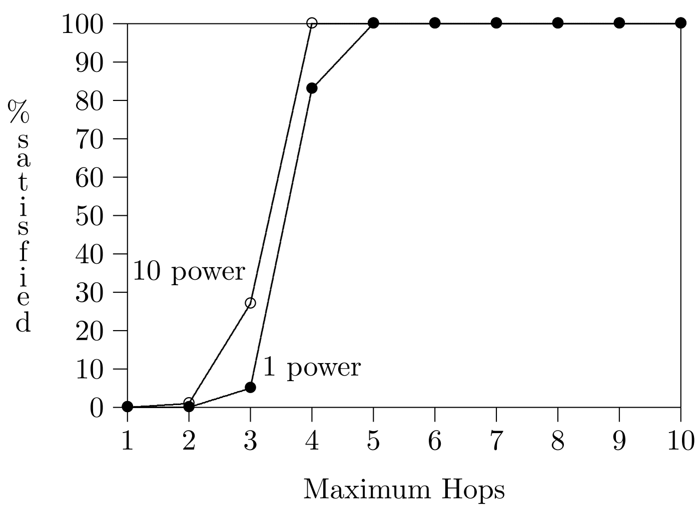

Figure 3: Percentage of randomly generated trees with virtual topologies as a function of maximum permitted hop distance when transmission power is fixed to be 1 or 10. All trees have height 10 and each vertex has between 1 and 3 children selected at random. Each edge has 5 wavelengths available.

4 The Splitting Virtual Topology Problem

In the previous section we have shown that the tap-and-continue virtual topology problem is solvable in polynomial time. In this section we show that the related splitting virtual topology problem is NP-complete. The splitting virtual topology decision problem is the decision version of the optimization problem and is stated as follows:

Splitting Virtual Topology Decision Problem (SVTDP): *Given a directed tree $T = (V, E)$ with root $r \in V$, a positive integer number of wavelengths $\mathcal{W}$, a positive integer transmission power $\mathcal{P}$, and a positive integer $\mathcal{H}$, does there exist a splitting virtual topology with maximum hop distance at most $\mathcal{H}$?*

The NP-completeness proof uses a reduction from the Bin Packing decision problem which is stated as follows:

Bin Packing: *Given a set $U = \{u_1, \ldots, u_n\}$ of items, a positive integer size $s(u_i)$ associated with each item, bin capacity B, and a positive integer number of bins K, can U be partitioned into K disjoint sets $U_1, \ldots, U_K$ such that the sum of the sizes of the items in each set U_i is B or less?*

The Bin Packing problem is known to be NP-complete in the strong sense[2] [5].

Theorem 3 SVTDP *is NP-complete.*

Proof: SVTDP is clearly in the class NP since a certificate can be easily verified in polynomial time. We show that SVTDP is NP-hard by a reduction from Bin Packing. Consider an instance of Bin Packing. Let $S = \sum_{i=1}^{n} s(u_i)$. Without loss of generality, we may assume that $S < BK$: If $S > BK$ no solution to the instance exists. If $S = BK$ we can modify the given instance by adding one more bin and adding an additional element to set U with size $B - 1$. It is easily verified that this new instance has answer "yes" if and only if the original instance has answer "yes".

Since $S < BK$, we define $d = BK - S - 1$ and note that d is non-negative. We construct an instance of SVTDP with $\mathcal{W} = K$, $\mathcal{P} = B$, and $\mathcal{H} = 2$. In order to construct the multicast tree, we first define some gadgets. A *gray gadget* is a tree with a root and $(\mathcal{W} - 1)\mathcal{P} - 1$ children. A *white gadget* is a tree with a root and $\mathcal{W}\mathcal{P} + 1$ children. A *k-white gadget* is a path of k vertices, called *path vertices*, each of which has an additional $\mathcal{W}\mathcal{P} + 1$ children. (Thus, a 1-white gadget is the same as a white gadget.) These gadgets, with their iconic representations, are shown in Figure 4. The multicast tree T is now constructed as follows: The root of the tree is a single vertex r with one child s, which is the root of a white gadget. In addition to the $\mathcal{W}\mathcal{P} + 1$ children in the white gadget, s also has $n + d$ children, $v_1, \ldots, v_n$, $x_1, \ldots, x_d$, each of which is the root of a gray gadget. Each vertex v_i has one additional child, w_i, which is the root of a $s(u_i)$-white gadget and each vertex x_i has one additional child, y_i, which is the root of a white gadget. This construction is shown in Figure 5.

Since Bin Packing is NP-complete in the strong sense, we may assume that all parameters in the given instance are bounded by a polynomial in the size of the instance.

[2] Recall that NP-completeness in the strong sense means that the problem is NP-complete even when all of the integer parameters of the problem are bounded by a polynomial in the size of the problem instance.

Consequently, the number of vertices in the constructed multicast tree is bounded by a polynomial in the size of the Bin Packing instance and thus the reduction can be performed in polynomial time.

We now show that the answer to the given instance of the Bin Packing decision problem is "yes" if and only if the answer to the constructed instance of SVTDP is "yes". Assume that the answer to the given instance of Bin Packing is "yes". Then we construct a virtual topology in the corresponding multicast tree as follows. Let $U_1, \ldots, U_K$ be a partition of U such that the sum of the sizes of the elements in each U_i is at most B. Let $u_{i_1}, \ldots, u_{i_m}$ denote the items in U_i and let $S(U_i) = \sum_{j=1}^{m} s(u_{i_j})$. Then, a single light-tree p_i is transmitted on wavelength λ_i from r to the $S(U_i)$ path vertices in the gadgets with roots $w_{i_1}, \ldots, w_{i_m}$. A total of $K = \mathcal{W}$ light-trees are thus constructed. Since $S(U_i) \leq B = \mathcal{P}$, each constructed light-tree does not exceed the available power. Moreover, the total amount of unused power among these K light-trees is $BK - S = d+1$. Therefore, the K light-trees can be collectively augmented to reach the $d+1$ vertices s and $y_1, \ldots, y_d$. The remaining vertices in the white and k-white gadgets can now be reached from these vertices in a second hop. Since only one wavelength has been used so far between s and each of its children $v_1, \ldots, v_n$ and $x_1, \ldots, x_d$, s can now transmit $\mathcal{W} - 1$ light-trees on each such edge with full power $\mathcal{P}$. Each such light-tree can thus reach all $(\mathcal{W} - 1)\mathcal{P}$ vertices in each gray gadget. Thus, all vertices are reached in at most 2 hops and the answer to the constructed instance of SVTDP is also "yes".

Conversely, assume that the answer to the given SVTDP instance is "yes". Let u denote the root of a white gadget or a path vertex in a k-white gadget. Since u has more than $\mathcal{WP}$ leaf children, at least one of these children must receive the message on a second hop on a light-tree transmitted by u. Thus, each such vertex u must receive the message in one hop. The total number of such vertices is $1+S+d = BK = \mathcal{WP}$. Thus, the single edge from r to s must transmit $\mathcal{W}$ distinct light-trees, each with power $\mathcal{P}$, in order to reach these vertices. Let these $\mathcal{W}$ light-trees be called *white gadget light-trees*. Moreover, each edge from s to a child $v_1, \ldots, v_n$ or $x_1, \ldots, x_d$ has at least one wavelength used by a white gadget light-tree. Since all wavelengths on the edge from r to s have been used, each vertex in a gray gadget must receive the message on a second hop via a light-tree transmitted by s. Since each gray gadget has a total of $(\mathcal{W}-1)\mathcal{P}$ vertices and at most $\mathcal{W}-1$ wavelengths remain on each edge from s to $v_1, \ldots, v_n$ and $x_1, \ldots, x_d$, s must use all $\mathcal{W}-1$ wavelengths at power $\mathcal{P}$ on each such edge. Therefore, exactly one wavelength may be used on each such edge by the white gadget light-trees. Thus, all $s(u_i)$ path vertices in a gadget labeled w_i must receive the message on the same white gadget light-tree. Consider the white gadget light-tree transmitted by r using wavelength λ_j, $1 \leq j \leq \mathcal{W}$. For each gadget labeled w_k which it reaches, include item u_k with size $s(u_k)$ in bin U_j. In this way, the items $u_1, \ldots, u_n$ are partitioned into $K = \mathcal{W}$ bins. Moreover, since $B = \mathcal{P}$, the total size of the items placed in each bin does not exceed $\mathcal{B}$. Therefore, the answer to the Bin Packing instance is "yes". □

Finally, we note that if the switches have limited splitting capabilities, permitting each wavelength to be split to just two outgoing ports in addition to the local node, the optimal virtual topology problem remains NP-complete. The gadgets used in the proof of Theorem 3 can be trivially modified so that each vertex v with out-degree larger than two is simply replaced by a binary tree in which the leaves represent the children of v.

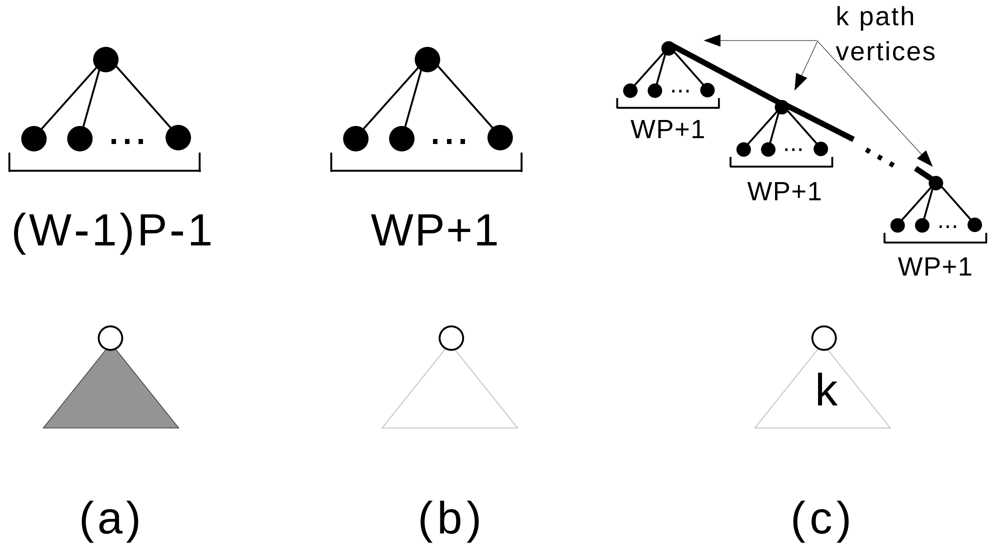

Figure 4: The three gadgets used in the reduction. The iconic representation is shown below each gadget. (a) A gray gadget. (b) A white gadget. (c) A k-white gadget.

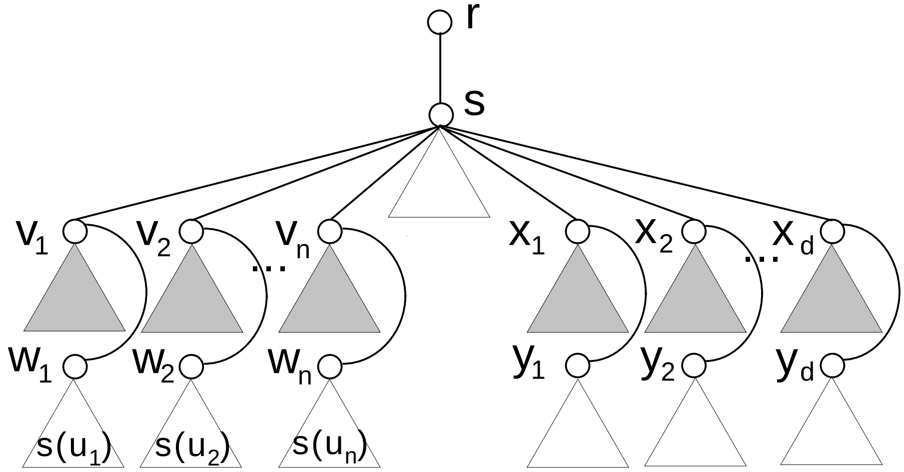

Figure 5: The multicast tree constructed in the reduction.

5 Conclusion

We have investigated the problem of finding an optimal multicast topology, with respect to minimizing the maximum hop distance, for a given multicast tree. We have shown that the problem can be solved by a polynomial-time algorithm for the case of tap-and-continue virtual topologies but is NP-complete for splitting virtual topologies. This fundamental difference in the computational complexity of the two problems is both theoretically interesting and also suggests that heuristics will be required for splitting virtual topologies.

The techniques proposed in this paper may be applicable to other optimization objectives and this is an interesting direction for future research. In addition, our results apply for a single multicast (or for multiple multicasts using edge-disjoint multicast trees). An interesting and important extension is to the case of multiple multicasts using shared or overlapping multicast trees. A related problem arises when multicast sessions requires only a fraction of the bandwidth provided by a WDM channel. In this case, a traffic grooming problem arises and such problems merit further research. Finally, our work considers splitting loss alone. Other factors such as attenuation and other impairments could be modeled in future work.

References

[1] M. Ali and J. Deogun. Power-efficient design of multicast wavelength-routed networks. *IEEE Journal on Selected Areas in Communications*, 18(10):1852–1862, October 2000.

[2] A. Arora, S. Subramaniam, and H-A Choi. Logical topology design for linear and ring optical networks. *IEEE Journal on Selected Areas in Communications*, 20(1):62–74, January 2002.

[3] J.-C. Bermond, L. Gargano, S. Perennes, A. Rescigno, and U. Vaccaro. Efficient collective communication in optical networks. In *ICALP: Annual International Colloquium on Automata, Languages and Programming*, 1996.

[4] B. Chen and J. Wang. Efficient routing and wavelength assignment for multicast in WDM networks. *IEEE Journal of Selected Areas in Communications*, 20(1), January 2002.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[6] O. Gerstel, A. Wool, and S. Zaks. Optimal layouts on a chain ATM network. *Discrete Applied Mathematics*, 83(1-3):157–178, 1998.

[7] W. Hu and Q. Zeng. Multicasting optical cross connects employing splitter-and-delivery switch. *IEEE Photonics Technology Letters*, 10(7):970–972, July 1998.

[8] D. Li, X. Du, X. Hu, L. Ruan, and X. Jia. Minimizing number of wavelengths in multicast routing trees in WDM networks. *Networks*, 35:260–265, 200.

[9] R. Libeskind-Hadas. Efficient collective communication in WDM networks with a power budget. In *Proceedings of the IEEE International Conference on Computer Communications and Networks*, pages 612–616, October 2000.

[10] R. Libeskind-Hadas and R. Melhem. Multicast routing and wavelength assignment in multi-hop optical networks. *IEEE/ACM Transactions on Networking*, 10(5):621–629, October 2002.

[11] R. Malli, X. Zhang, and C. Qiao. Benefit of multicasting in all-optical networks. In *SPIE Conference on All-Optical Networks*, volume 3531, pages 209–220, 1998.

[12] C. Noronha and F. Tobagi. Routing of multicast audio/video streams in reconfigurable WDM optical networks. *Journal of Network and Systems Management*, 4(2), 1996.

[13] R. K. Pankaj. Wavelength requirements for multicasting in all-optical networks. *IEEE/ACM Transactions on Networking*, 7(3):414–424, June 1999.

[14] G. Rouskas. Optical layer multicast: Rationale, building blocks, and challenges. *IEEE Network*, 17(1):60–65, January/February 2002.

[15] George Rouskas. Routing and wavelength assignment in optical WDM networks. In John Proakis, editor, *Wiley Encyclopedia of Telecommunications*. John Wiley and Sons, 2001.

[16] L. H. Sahasrabudde and B. Mukherjee. Light-trees: Optical multicasting for improved performance in wavelength-routed networks. *IEEE Communications Magazine*, 37(2):67–73, February 1999.

[17] D. Thaker and G. Rouskas. Multi-destination communication in broadcast WDM networks: A survey. *Optical Networks*, 3(1), January/February 2002 (to appear).

[18] J. Wang, B. Chen, and R. N. Uma. Dynamic wavelength assignment for multicast in all-optical WDM networks to maximize the network capacity. *IEEE Journal on Selected Areas in Communications*, 21(8):1274–1284, 2003.

[19] Y. Xin and G. Rouskas. Multicast routing under optical layer constraints. In *Proceedings of IEEE Infocom*, 2004.

[20] J. Yates, M. Rumsewicz, and J. Lacey. Wavelength converters in dynamically reconfigurable WDM networks. *IEEE Communications Surveys*, 1999.

[21] X. Zhang, J. Wei, and C. Qiao. Constrained multicast routing in WDM networks with sparse light splitting. In *Proceedings of INFOCOM*, pages 1781–1790, 2000.