MICRON – A Framework for Connection Establishment in Optical Networks

Srinivasan Ramasubramanian *Member, IEEE,* and and Arun K. Somani *Fellow, IEEE*

***Abstract*—Traffic grooming in optical networks has gained significance due to the prevailing sub-wavelength requirement of end users. Optical networks get upgraded to the latest technology slowly with time with only a subset of nodes being upgraded to the latest technology. The networks are thus comprised of nodes employing heterogeneous switching architectures. In this paper, we develop a framework called Methodology for Information Collection and Routing in Optical Networks (MICRON) for connection establishment in optical grooming networks with heterogeneous switching architectures. We illustrate with examples the information that may be collected from the links, and operators that may be used to obtain information along a path. The information can be used to select a path dynamically depending on the network status. We complete the MICRON framework by providing a generic channel assignment procedure that could be employed to implement different channel assignment schemes. Various routing and channel assignment algorithms can be developed from the proposed framework. The framework may be easily implemented with simple traffic engineering extensions to the already existing routing protocols in the wide-area networks.**

I. INTRODUCTION

Optical communication employing wavelength division multiplexing (WDM) has emerged as a viable solution for satisfying the ever-increasing quest for bandwidth due to emerging Internet applications. WDM divides the available fiber bandwidth into multiple wavelengths each of which operates at peak electronic speed. Present day networks have a transmission capacity of 40 Gbps (OC-768) on a wavelength. However, the user requirements are of sub-wavelength capacity, typically ranging from 155 Mbps (OC-3) to 622 Mbps (OC-12), and rarely in the range of a few gigabits per second. Hence, alternatives for provisioning of sub-wavelength traffic in WDM networks has received significant attention in the recent past. One approach to provisioning sub-wavelength traffic is to divide a wavelength into time slots that would allow multiple traffic to be time multiplexed on the wavelength. However, employing TDM at high transmission speeds in long-haul networks requires stringent synchronization across the network. Hence, Code Division Multiple Access (CDMA) approaches may be employed to share a wavelength bandwidth across multiple users. The resulting multi-wavelength multi-time slot or multi-code network is referred to as a WDM grooming network.

Wavelength converter (WC) is a device that converts an optical signal from one wavelength to another. Similarly, a time slot interchanger (TSI) is a device that converts an optical signal from one time slot to another. The optical version of TSI is realized using fiber delay lines [1], [2]. All-optical implementations of WC and TSI are expensive, hence cannot be implemented at all nodes in the network. CDMA code converters can be implemented using passive components such as fiber-bragg gratings. In the rest of the paper, we use the terminology of time slots, however the research developed in this paper is applicable to CDMA-based systems as well.

WDM grooming networks can be classified into two categories [3]: dedicated-wavelength grooming (DWG) networks and shared-wavelength grooming (SWG) networks. In DWG networks, a lightpath between two nodes is shared only by the traffic between them. In SWG networks, the lightpath can be shared by traffic from other nodes as well. The performance of SWG networks depends on efficient merging of fractional wavelength requirements into full or almost-full wavelength requirements. Such merging of smaller capacity requirements into higher capacity lightpaths is called *traffic grooming*.

Traffic grooming in SWG networks can be performed in a static or dynamic manner. In static grooming, the source-destination pairs whose traffic requirements will be combined are pre-determined. In dynamic grooming, connections of different source-destination pairs are combined based on the existing lightpaths at the time of a request arrival.

In this paper, we consider dynamic grooming in SWG networks. The performance of SWG networks depends on efficient grooming of sub-wavelength traffic and therefore require sophisticated routing algorithms based on up-to-date network information. Nodes in a SWG network can have different levels of grooming capability. For example, a node might employ TSI but not WC. Such nodes are referred to as wavelength-level grooming nodes as channels can be switched only within a wavelength. Similarly some other nodes might implement WC but not TSI, referred to as time slot level grooming nodes. A node employing both TSI and WC are referred to as full-grooming node.

As the present day optical technology is not mature enough to make routing decisions at run time, wide-area optical networks are expected to remain circuit-switched in nature for the near future. Operating a circuit-switched wide-area network involves establishing connections of a certain bandwidth between two nodes in the network. It is well understood that networks in future would comprise of nodes with heterogeneous switching architectures. One of the reasons is that upgrading of existing switch architectures to the latest optical technology takes both time and money. Financial constraints limit the

The research developed in this paper is funded in part by the National Science Foundation under grant ANI-9973102 and Defense Advanced Research Projects Agency and National Security Agency under grant N66001-00-18949.

Srinivasan Ramasubramanian is with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721. E-mail: srini@ece.arizona.edu. Arun K. Somani is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011. E-mail: arun@iastate.edu.

network upgrade procedure to enhancing the capability of only a few nodes or increasing the capacity of only a few links depending on the traffic flowing through them. The networks are thus heterogeneous in nature as they constantly evolve with time. Hence, it is important to address the issue of connection establishment in such heterogeneous networks.

Connection establishment in a connection-oriented network consists of two steps: *path selection* and *channel assignment*. Path selection refers to selecting a path from source to destination based on certain criteria. Channel assignment refers to assigning one or more channels, depending on the requirement of the request, on every link of the chosen path. Path selection may be carried out in several ways. If a source-destination pair has one pre-selected path, then it is referred to as *fixed-path* approach. If a path is selected depending on the network status from a pre-selected set of candidate paths, then it is referred to as *dynamic path selection*. The set of candidate paths remain the same at all times and do not change with the network status. If the candidate paths are chosen based on the network status, the path selection process is referred to as *exhaustive routing*. Channel assignment refers to allocation of specific resources on every link of a chosen path, for example: (a) fiber, wavelength, and time slot assignment on the links in a WDM grooming network; and (b) fiber and wavelength assignment in a multi-fiber wavelength-routed network. Irrespective of the path selection or channel assignment strategy employed in the network, obtaining information along a path to assess the availability of resources to establish the connection becomes the fundamental requirement. Information collection in optical grooming networks involve identifying availability of resources on the links along with the grooming capability of intermediate nodes on a specific path to identify capacity availability on the path.

The connection establishment problem is typically modeled as an optimization (mixed-integer linear programming) problem when the set of connections to be established are known a priori. Typically heuristic approaches are employed as the solution times for the optimization problem may be prohibitively high for large-scale networks. The objective in such formulations is to reduce the network cost while being able to accommodate all the connections. Some earlier work concentrate on specific network topologies, such as rings [4], [5], [6], [7], while others are developed for arbitrary (or mesh) topologies [8], [9], [10], [11]. In [12], the authors consider the problem of rerouting the existing sub-wavelength traffic in the network to optimize the wavelength utilization.

Connection establishment has been extensively studied in the context of wavelength-routed WDM networks under dynamic traffic scenarios [13], [14], [15]. Under dynamic traffic scenarios, the objective is to compute a path and a channel assignment for a given request based on the current network status. Connection establishment in WDM grooming networks has been addressed in [16], [17], [18]. In [16], the authors develop a family of routing, wavelength and time-slot assignment algorithms for a TDM-based wavelength routing network. The nodes are assumed to have switching capability at the granularity of a time slot. The nodes, however, do not have time slot interchangers or wavelength converters. If W denotes the number of wavelengths per link and T denotes the number of time slots per wavelength, then the routing, wavelength, and time slot assignment problem is similar to routing and wavelength assignment problem in a wavelength routed network with $W \times T$ wavelengths per link.

In [17], [18], the authors develop a graph-based model in which every node is replaced by W wavelength-nodes, where W denotes the number of wavelengths per fiber. A fiber link between nodes i and j is replaced by W auxiliary links, each connecting a specific wavelength-node of i to a specific wavelength-node of j. If a node i has full-wavelength conversion, then links between all wavelength-node pairs at node i are added. Such a graph model translates a WDM network with N nodes, L links, and W wavelengths per link to a graph with $N \times W$ nodes and $W \times L$ links. If the nodes employ wavelength conversion, then additional links are required. The maximum number of additional links required is $\frac{NW(W-1)}{2}$, when all the nodes have wavelength conversion. It is worth noting that when all the nodes have wavelength conversion, the network may be simply viewed as N nodes connected by L links with W indistinguishable channels in each link. However, the graph model does not automatically allow such a reduction in network representation. In addition, such a graph-based model may be employed only as a centralized mechanism to identify paths, hence suited for networks that employ link-state protocols. If the networks employ alternate path routing [19] or least-loaded path routing [20], [21], the availability of capacity on a path may be computed by probing (sending messages along) the desired paths.

Earlier works on routing in traffic grooming networks simply consider a wavelength routed network where some nodes may employ wavelength conversion. However, given that multiplexing of low-rate traffic to a high-capacity lightpath may be accomplished in several dimensions, e.g. WDM, TDM, CDMA, it is conceivable that nodes might employ switching capabilities only in some dimensions. The research in this paper addresses the problem of efficient representation of link and node information in a traffic grooming network with heterogeneous switching architectures.

In this paper, we develop a framework for connection establishment called MICRON (Methodology for Information Collection and Routing in Optical Networks) in WDM grooming networks, specifically emphasizing on what and how information is collected on the links, the aggregation strategy to obtain information on a path considering different grooming capabilities of the intermediate nodes, and mechanisms for path selection and channel assignment. Several routing and channel assignment strategies may be developed from the framework developed in this paper.

The remainder of the paper is organized as follows: Section II explains the assumptions on the network model, node architecture, and notations employed. Section III describes an example network. Section IV introduces the framework and illustrates different path selection and channel assignment schemes on the example network. Section V concludes the paper.

II. NETWORK MODEL

Consider an optical grooming network with nodes employing heterogeneous switching architecture. The nodes in the network are connected using links. Let N denote the number of nodes and L denote the number of links in the network. Each link is assumed to carry F fibers, each fiber carrying W wavelengths. Each wavelength is divided into frames which are further sub-divided into T time slots. Every slot within a frame is denoted by a 4-tuple, (l, f, w, t), where $1 \leq l \leq L$, $1 \leq f \leq F$, $1 \leq w \leq W$, and $1 \leq t \leq T$. For example, the tuple $(1, 1, 2, 1)$ (read from right to left) denotes first time slot in a frame on the second wavelength of the first fiber on the first link. A *channel* on a link is defined as a collection of a particular occurrence of time slot across successive frames. Hence, the number of channels in a link is the same as the number of slots in a frame, $F \times W \times T$. Each channel is also represented by a 4-tuple, (l, f, w, t).

A. Node architecture

A WDM grooming network with heterogeneous network architecture is modeled as a Trunk Switched Network (TSN). The TSN model was introduced in [22] to enable modeling of multi-wavelength optical networks and evaluate them for blocking probability. The MICRON framework is developed for the TSN model, therefore we describe the TSN model in detail.

A TSN is a two-level network model in which every link in the network is viewed as multiple channels. A node in a TSN groups the channels with similar characteristics in a link into groups called *trunks*. The definition of a trunk at a node depends on the switching resources available at a node. We illustrate the notion of trunks with an example. Consider a WDM grooming network where every link has four fibers, three wavelengths per fiber and two time slots per frame ($F = 4$, $W = 3$, $T = 2$). Fig. 1 shows the channels on a link. The shapes represent the time slots and the shades represent wavelengths.

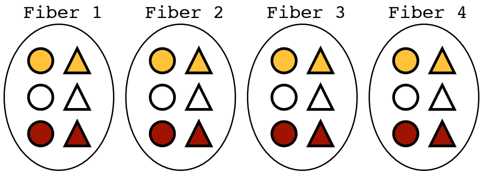

Fig. 1. Representation of twenty four channels in a link having four fibers, three wavelengths per fiber, and two time slots per wavelength. Shapes represent time slots, shades represent wavelengths, number of shapes of a certain shade represents the number fibers.

If time slot interchange and wavelength conversion are not permitted, a node i views link ℓ as $W \times T$ trunks where each wavelength and time slot combination (w, t) forms a trunk. Every trunk has F channels as shown in Figure. 2(a). If time slot interchange is permitted, but not wavelength conversion, a node i views link ℓ as W trunks, where each wavelength forms a trunk. Every trunk has $F \times T$ channels as shown in Fig. 2(b). A node with such a capability is referred to as a *wavelength-level grooming* node. If full-wavelength conversion is permitted, but not time slot interchange, then each time slot t on the link l forms a trunk. Every trunk has $F \times W$ channels as shown in Fig. 2(c). A node with such a capability is referred to as a *time slot-level grooming* node. If both full-wavelength conversion and time slot interchange are permitted, then the entire link is treated as one trunk with $F \times W \times T$ channels, as shown in Fig. 2(d). A node with such a capability is referred to as a *full grooming* node.

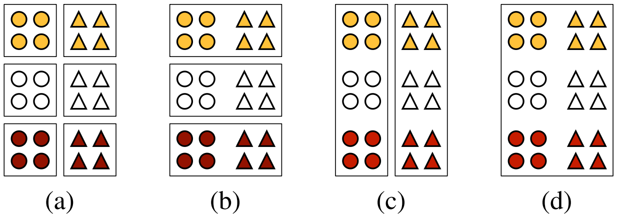
(a) (b) (c) (d)

Fig. 2. Possible grouping of channels in a link as trunks. (a) Each wavelength and time slot combination is a trunk; (b) Each wavelength is a trunk; (c) Each time slot is a trunk; and (d) The link is a trunk.

A single-fiber wavelength-routed WDM network employing W wavelengths can be modeled as W trunks with one channel per trunk. A multi-fiber multi-wavelength wavelength-routed network with F fibers and W wavelengths with no wavelength conversion can be viewed as W trunks with F channels per trunk. If full-wavelength conversion is available, then a link can be viewed as a single trunk with FW channels.

Fig. 3 shows the node architecture in a TSN. The node in the figure is assumed to have three links attached to it and views each link as a set of K trunks. The trunks are first de-multiplexed from the link. The trunks from different links are then sent to their respective trunk switches where the channels are switched. We impose trunk-continuity constraint at a node – i.e. a channel in a trunk on a link may be switched only to a channel that falls within the same trunk on another link. Such a restriction stems from an architectural point of view. The complexity of a full permutation switching over all channels across all links is prohibitively large. Therefore, switch designs for the near future are likely to be based on simple architectures that would switch channels within restricted groups.

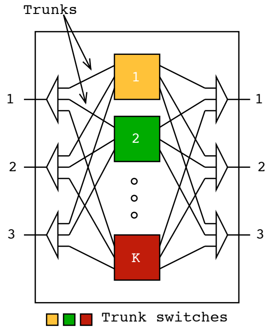

Fig. 3. Node architecture in a Trunk Switched Network.

A TSN is said to be *homogeneous* if the collection of channels that constitute a trunk at a node is the same for all the nodes in the network. Otherwise, it is said to be *heterogeneous*. In this paper, it is assumed that a full-permutation switch is employed for every trunk in a node, i.e., a free channel of a trunk at the input of the switch can be switched to any free channel of the same trunk at its output. We assume that switching channels across fibers is available at all nodes as it requires only space switching. We, therefore consider only wavelength and time-slots as trunks in this paper.

A node i in the network views a link ℓ connected to it as a set of K_i trunks each containing S_i channels. Let $\chi^i_{\ell,x}$ denote the channels on a link that fall within trunk x at node i. Let Θ^{ij}_{xy} denote the channels on link (i,j) that fall within trunk x at node i and trunk y at node j, i.e., $\Theta^{ij}_{xy} = \chi^i_{\ell,x} \cap \chi^j_{\ell,y}$. The group of channels that fall within a set Θ^{ij}_{xy} is referred to as a *sub-trunk*.

Calls arriving in the network request for a connection to be established from a source to destination. The connection establishment involves selection of a path and assignment of channels on the path such that the channel on one link can be switched to the successive link on the path by the node connecting the links. In a TSN, connection establishment consists of three steps: (1) selecting a path; (2) assigning a sub-trunk on every link, or equivalently assigning a trunk at every node; and (3) assigning one or more channels depending on the call requirement on every sub-trunk on every link. Hence, a connection in a network is represented by a sequence of link and sub-trunk pairs, or equivalently as a sequence of node and trunk pairs. If every node in the network employs full permutation switching for every trunk, then any channel that falls within the selected sub-trunk on a link can be chosen for establishing a connection.

III. EXAMPLE NETWORK

Consider the example network shown in Fig. 4 illustrating two paths from node 1 to 5. Let the nodes be connected using three fibers each carrying three wavelengths and two time slots per wavelength. Also assume that nodes 1, 3, 6, and 7 are wavelength-level grooming nodes; nodes 2 and 5 are time-slot-level grooming nodes; and node 4 is a full-grooming node. Wavelength-level grooming nodes view the link as 3 wavelength trunks (denoted by W_1, W_2, and W_3) with 6 channels in each, time slot-level grooming nodes view a link as two time slot trunks (denoted by T_1 and T_2) with 9 channels in each, and a full-grooming node views a link as one trunk (denoted by F_1) with 18 channels.

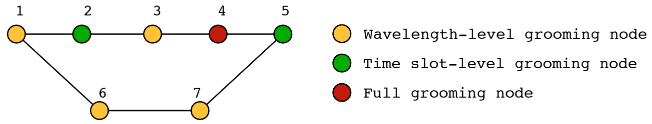

Fig. 4. An example network showing two paths from node 1 to node 5.

Fig. 5 shows the expanded view of the network indicating the different trunks at the nodes. For example, consider trunk W1 of node 1 and trunk T1 of node 2. The number of channels in the link 1–2 (denoted by ℓ_{12}) that belongs to both the trunks is 3. The channels are $(\ell_{12},1,1,1)$, $(\ell_{12},2,1,1)$ and $(\ell_{12},3,1,1)$, each channel belonging to a distinct fiber. The arrow connecting trunk W1 of node 1 to trunk T1 of node 2 indicates the number of free channels that belong to both the trunk definitions. A value of 3 indicates that all the channels belonging to both the trunk definitions are free.

Assume that the network is observed at some instant of time during its operation and the channel occupancy in the links are known. Let $(l,f,w,t).Availability$ denote the availability of the channel: denoted by 0 if occupied by a connection, 1 if the channel is free.

IV. MICRON FRAMEWORK

The MICRON framework developed in this paper addresses in detail the information representation on a link, aggregation of link information to obtain the path information, selection of a path, and assignment of sub-trunks in WDM grooming networks with heterogeneous grooming capabilities. Each of these steps are described in detail in the following subsections.

A. Link information

A link connecting nodes i and j is represented by a matrix L_{ij}.

$$L_{ij} = \begin{bmatrix} l_{11} & l_{12} & \ldots & l_{1K_j} \\ l_{21} & l_{22} & \ldots & l_{2K_j} \\ . & & & \\ . & & & \\ l_{K_i1} & l_{K_i2} & \ldots & l_{K_iK_j} \end{bmatrix} \tag{1}$$

Each element l_{xy} denotes a certain property about the channels in the link that belong to Θ^{ij}_{xy}. Consider the link 1–2 in the example network shown in Fig. 4. Node 1 views each wavelength as a trunk, hence has 3 trunks. Node 2 views each time slot as a trunk, hence has 2 trunks. Therefore, L_{12} is a 3×2 matrix.

The matrix can denote different properties of the channels. We discuss two specific examples in this paper.

Case 1: Connectivity
In this case, every element l_{xy} of the matrix L_{ij} is denoted by 1 if the total number of free channels that belong to Θ^{ij}_{xy} has a capacity of at least B. The matrix L_{ij} is defined as:

$$l_{xy} = \begin{cases} 1 & \text{if } \left(\sum_{(l,f,w,t)\in\Theta^{ij}_{xy}} (l,f,w,t).Availability\right) \geq B \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $1 \leq x \leq K_i$ and $1 \leq y \leq K_j$. This matrix provides the connectivity information to route a call that requires a capacity of B without splitting the connection. For a call with one channel capacity requirement ($B = 1$), the matrices for different links are shown in Fig. 6.

Case 2: Available capacity
In this case, every element l_{xy} of the matrix L_{ij} is defined as the number of free channels that belong to Θ^{ij}_{xy} as:

$$l_{xy} = \sum_{(l,f,w,t)\in\Theta^{ij}_{xy}} (l,f,w,t).Availability \tag{3}$$

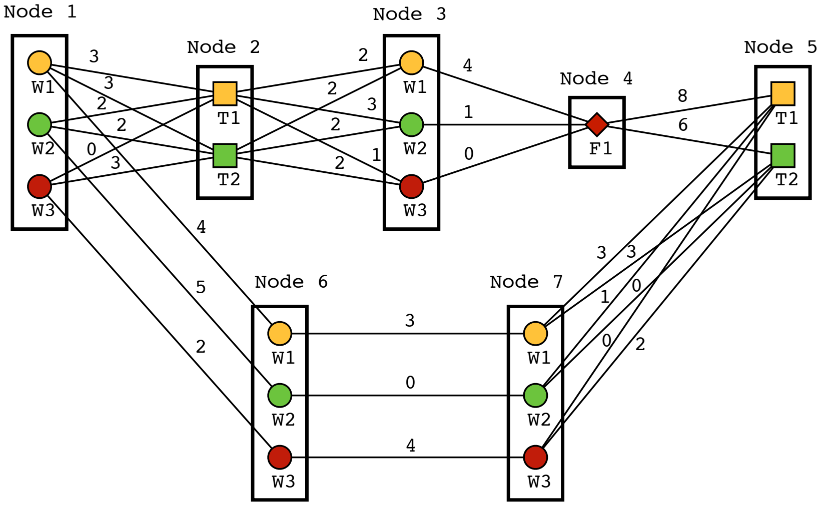

Fig. 5. Expanded view of the network with channel occupancy information.

$$L_{12} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \quad L_{23} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad L_{34} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \quad L_{45} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

$$L_{16} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad L_{67} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad L_{75} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Fig. 6. Link information matrices indicating if there is at least one free channel in a sub-trunk.

$$L_{12} = \begin{bmatrix} 3 & 3 \\ 2 & 2 \\ 0 & 3 \end{bmatrix} \quad L_{23} = \begin{bmatrix} 2 & 3 & 1 \\ 2 & 2 & 2 \end{bmatrix} \quad L_{34} = \begin{bmatrix} 4 \\ 1 \\ 0 \end{bmatrix} \quad L_{45} = \begin{bmatrix} 8 & 6 \end{bmatrix}$$

$$L_{16} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad L_{67} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 4 \end{bmatrix} \quad L_{75} = \begin{bmatrix} 3 & 3 \\ 1 & 0 \\ 0 & 2 \end{bmatrix}$$

Fig. 7. Link information matrices indicating the number of free channels in a sub-trunk.

where $1 \leq x \leq K_i$ and $1 \leq y \leq K_j$. The matrix representation for different links are shown in Fig. 7 that correspond to the network state shown in Fig. 5.

Note that the matrices obtained using the available sub-trunk capacity also contains the information of the matrices representing connectivity information. Depending on the level of information required, different matrix representations may be employed.

B. Path information

The information about a certain path from a node i to k that are not physically connected by a fiber is obtained by combining the link information along the path. The matrix representation for a path is defined in a manner similar to that of a link. A path matrix from node i to k through j is obtained as a matrix multiplication of individual path segments P_{ij} and P_{jk} as:

$$P_{ik} = P_{ij} P_{jk} \tag{4}$$

We employ a generalized version of matrix multiplication to compute the path matrix. An element p^{ik}_{xy} (the superscript ik denotes the matrix to which the element belongs to) is obtained as:

$$p^{ik}_{xy} = (p^{ij}_{x1} \otimes p^{jk}_{1y}) \oplus (p^{ij}_{x2} \otimes p^{jk}_{2y}) \oplus ... \oplus (p^{ij}_{xK_j} \otimes p^{jk}_{K_j y}) \tag{5}$$

The operators $\otimes$ and $\oplus$, denoted as a tuple $(\otimes, \oplus)$, may be defined in different combinations so that several meaningful results are obtained. It may be observed that when $\otimes$ is integer multiplication and $\oplus$ is integer addition, the above equation denotes the traditional matrix multiplication.

To illustrate the significance of different operators, we take the two example matrix representations of links and apply two different set of operators to obtain different information from the network.

Case 1: Arithmetic operators

In this case, we consider integer multiplication ($\times$) and integer addition ($+$) as the operators for $\otimes$ and $\oplus$, respectively.

Consider the matrix representation shown in Fig. 6. Applying the operator on the path 1–2–3–4–5, we obtain the path information matrix as:

$$P_{1-2-3-4-5} = \begin{bmatrix} 4 & 4 \\ 4 & 4 \\ 2 & 2 \end{bmatrix} \tag{6}$$

An element p_{xy} of the above matrix denotes the number of distinct sub-trunk selections available from trunk x of node 1 to trunk y of node 5. For example, there are four paths that can start at trunk W1 of node 1 and end at trunk T1 of node 5. These four trunk assignments on the path are represented as a set of tuples containing node numbers and trunk identifier on that node. The four possible trunk assignments on the path, denoted by P_1 through P_4 are:

$$\begin{aligned} P_1 :&\quad \{(1, W_1), (2, T_1), (3, W_1), (4, F_1), (5, T_1)\} \\ P_2 :&\quad \{(1, W_1), (2, T_1), (3, W_2), (4, F_1), (5, T_1)\} \\ P_3 :&\quad \{(1, W_1), (2, T_2), (3, W_1), (4, F_1), (5, T_1)\} \\ P_4 :&\quad \{(1, W_1), (2, T_2), (3, W_2), (4, F_1), (5, T_1)\} \end{aligned}$$

The existence of trunk assignment for other trunk pairs can be easily verified from Fig. 4.

Consider the link information as shown in Fig. 7. Applying the operator $(\times, +)$ on these matrices results in the path information matrix for path 1–2–3–4–5 as:

$$P_{1-2-3-4-5} = \begin{bmatrix} 504 & 378 \\ 336 & 252 \\ 240 & 180 \end{bmatrix} \tag{7}$$

An element p_{xy} of the above matrix denotes the number of possible channel assignment combinations on the path that start at a certain trunk x at node 1 and end at a trunk y at node 5. On every sub-trunk on the path the number of ways of assigning a channel is the same as the number of channels in the sub-trunk. Hence, the number of possible channel assignments on a specific trunk assignment on the path is the product of the number of free channels on the assigned sub-trunk on every link. The number of possible channel assignments on the four possible trunk assignments P_1 through P_4 that start the connection at trunk W_1 at node 1 and end at trunk T_1 at node 5 are 192, 72, 192, and 48, respectively, adding up to 504 possible ways of channel assignment.

Case 2: Selection operators

In this case, we assume that the operator $\otimes$ indicates the minimum of the two operands while the operator $\oplus$ indicates the maximum of the two operands. Applying this set of operation to the matrix representation in Fig. 6, we obtain the matrix representation for the path 1–2–3–4–5 as:

$$P_{1-2-3-4-5} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{8}$$

The elements of this matrix indicate the existence of a channel allocation scheme for call requiring one channel capacity that would start at trunk x at node 1 and end at trunk y at node 5. The matrix in Equation (8) may be obtained from the matrix in Equations (6) or (7) by replacing every non-zero element in the matrix by 1.

Applying this set of operation to the matrix representation in Fig. 7, we obtain the maximum capacity that can be routed from node 1 to node 5 without splitting the connection. The matrix representation for the path is obtained as:

$$P_{1-2-3-4-5} = \begin{bmatrix} 2 & 2 \\ 2 & 2 \\ 2 & 2 \end{bmatrix} \tag{9}$$

Consider the possible trunk assignments that start the connection at trunk W_1 at node 1 and end at trunk T_1 at node 5. Consider the four possible trunk assignments P_1 through P_4. It is observed from Fig. 4 that the trunk assignments P_1 and P_3 have the link connecting node 2 to 3 as bottleneck with two channel capacity. The trunk assignments P_2 and P_4 have the link connecting node 3 to 4 as bottleneck with one channel capacity. Hence, a maximum of two-channel capacity connection can be routed from node 1 to 5 starting at trunk W_1 at node 1 and ending at trunk T_1 at node 5.

C. Two-pass approach to connection establishment

When a call arrives at a node, a request for connection establishment is sent along a set of candidate paths. The connection establishment is carried out in two passes: *Forward pass and Reverse pass* [23]. During the forward pass, the connection request is forwarded to the nodes along the path with a vector, called Path Information Vector (PIV). The path information vector at a node k for a path with source i and destination k, denoted by V_{ik} is of dimension $1 \times K_k$. V_{ik} is obtained as a product of the path information vector at the source node and the information matrix of the path connecting nodes i and k:

$$V_{ik} = U_i P_{ik} \tag{10}$$

where U_i denotes the path information matrix at the source node which is always set as a unit row vector.

Assume that the path from node i to k passes through node j. Rewriting the above equation gives the relationship between the PIV vectors at node j and node k.

$$\begin{aligned} V_{ik} &= U_i P_{ik} && (11) \\ &= U_i P_{ij} P_{jk} && (12) \\ &= V_{ij} P_{jk} && (13) \end{aligned}$$

The matrix-vector multiplication employed above is similar to the generalized matrix multiplication proposed earlier in the paper with the operator tuple $(\otimes, \oplus)$. The elements of PIV at a node indicates specific properties about paths that end at a certain trunk. For example, if the link information matrix represented in Fig. 6 and operator $(\times, +)$ are employed, then the resulting PIV at each node indicates the number of possible trunk assignments on the path that would terminate the connection on a certain trunk at that node.

During the forward pass of the connection establishment, an intermediate node j on the path from source i may forward either the path information matrix P_{ij} to its neighboring node or the path information vector V_{ij}. Forwarding the latter has the advantage of minimizing the amount of information forwarded. Note that the reduction in information exchange will be significant when the number of trunks at a node is large. Hence, we assume that only the path information vector is forwarded to the successive node along the path and will be employed to assign a sub-trunk on the reverse pass.

D. Path selection

The path information vector may be used to select a suitable path for a given source-destination pair.

For example, consider the two paths from node 1 to 5: 1–2–3–4–5 and 1–6–7–5. Employing the matrix information represented in Fig. 6 and operator $(\times, +)$, we obtain the path information vector for the two paths as:

$$V_{1-2-3-4-5} = \left[\begin{array}{cc} 10 & 10 \end{array}\right] \qquad V_{1-6-7-5} = \left[\begin{array}{cc} 1 & 2 \end{array}\right]$$

With these matrices known at the destination, one could employ different comparison algorithms to select a path. For example, the total number of trunk assignments possible on a path is obtained by summing all the elements of the matrix. A path that has the maximum value for this metric can be chosen for establishing the connection.

If the matrix representation in Fig. 7 and operator (max, min) are employed, the path information vector for the two paths are obtained as:

$$V_{1-2-3-4-5} = \left[\begin{array}{cc} 2 & 2 \end{array}\right] \qquad V_{1-6-7-5} = \left[\begin{array}{cc} 3 & 3 \end{array}\right]$$

It can be observed that the path 1–6–7–5 can route a call for three channel capacity request without splitting, while the other path cannot. Hence, if traffic requirements in the network are diverse and destination-based path-selection is employed, then the path 1–6–7–5 would be chosen so as to minimize the blocking at that instant of time.

Different metrics such as hop-length, delay, cost, etc. that could be included for link state vector and possible path selection schemes for WDM grooming networks are discussed in [24]. The path information vector can also be extended to include multiple metrics in order to select a path. The above mechanism to select a path may be employed in networks where alternate path [19] or least loaded path [20], [21] routing algorithms are employed, where every source-destination pair has a set of alternate paths. In alternate path routing, the paths are probed one by one sequentially for available capacity and the first available path. In the latter, all the paths are probed for available capacity and the least loaded path is chosen. In both these approaches, a node in the network need not be aware of the grooming capabilities at other nodes.

E. Path computation

If paths have to be computed based on the up-to-date network status, then the Dijkstra's shortest path algorithm may be extended to operate on the PIV and some additional metrics such as hop length, cost, etc. These metrics may be collected at the nodes through a link-state protocol [23]. The centralized version of the working of Dijkstra's algorithm using the path information vector (employing available capacity) and hop length is shown in Fig. 8. Steps 1 through 4 are initialization steps. In step 5, a selected node is marked. The neighbors of the nodes are updated with the new available capacity vector and path length values in steps 9 through 14. The vectors are updated if the new path has (1) a shorter path length (step 10); or (2) if the path lengths are same, then the path is replaced with a probability $\frac{1}{c_j}$, where c_j denotes the number of paths encountered so far with the same hop length.

A function that maps a vector to a scalar is employed to compare the PIVs. With such an approach, the routing algorithm makes a collective decision regarding all the trunks at a node. One such approach is to compare the maximum value of an element in the path information vector. If the maximum available capacity of a trunk exceeds the capacity of the request, then the connection can be established on that path. Steps 9 through 14 select the shortest path among the set of available paths. Step 15 identifies if there are any unmarked nodes left. If no such nodes are present, then the algorithm terminates. Otherwise, steps 16 through 23 select a node for marking. A node is selected must have an available path from the source and its path length must be the smallest among all such nodes. Step 24 initializes the selected node as i and proceeds with updating its neighbors.

The algorithm in Fig. 8 is similar to the Dijkstra's algorithm for a scalar metric, except that at every step of the algorithm a matrix-vector multiplication is carried out. The complexity of the Dijkstra's shortest path algorithm for a scalar metric is $O(N \log N + L)$. The complexity of matrix-vector multiplication of a link connecting two nodes with K and K' trunks is $O(KK')$. Let $\overline{KK'}$ indicate the average over all links the product of the number of trunks at the nodes connected by a link. The complexity of the algorithm in Fig. 8 is $O(\overline{KK'}(N \log N + L))$.

F. Sub-trunk assignment

At the end of the forward pass, the destination node has the path information vector for the different probed paths and selects a path based on a certain path selection algorithm. Once a path is chosen, a sub-trunk has to be selected on every link of the path in order to complete the channel establishment. The sub-trunk assignment is carried out as follows. (1) the destination node first selects the trunk at its node where the connection would terminate; and (2) every node along the path selects the output trunk at its previous node. If a link connects node i and j, then the node j selects the output trunk at node i, hence the sub-trunk assignment on the link i–j.

Consider the information matrix represented in Fig. 6 and operator $(\times, +)$. The path information vectors obtained at different nodes are shown in Fig. 9.

The trunk assignment to end the connection at the destination node may be made using the path information vector. Several trunk selection schemes such as first-fit, best-fit, random, etc. may be employed. In this paper, we illustrate the random sub-trunk assignment. Let x_k denote the trunk that is chosen to accommodate the connection at node k.

In order to select a sub-trunk on link $j - k$, a *ratio vector* is computed at node k. The vector, denoted by R_{jk}, is obtained as the product of the vector at the path information

Notations

V_{ij}	Available capacity vector from node i to node j along a certain path.
A_{ij}	Available capacity matrix of link $i-j$.
h_{ij}	Hop length of an available path from node i to j.
t_v	Temporary vector to store the available capacity.
t_h	Temporary variable to store hop length.
$cap(V,x)$	Every element of vector V can have a maximum value of x.
nbr_{ij}	Neighbor of j used to reach node j from i.
RandomInteger(x,y)	Generates an integer random number from x to y (both inclusive).
$count_j$	Denotes the number of paths seen so far with the same least hop length.
$count$	Denotes the number of unmarked nodes that have an available path from the source with the same minimum hop count.

ComputePath(s,d,c)

```
1.  Unmark all nodes.
2.  Initialize $V_{ss}$ as a unit row vector and $h_{ss} = 0$.
3.  Initialize $V_{sj}$ as a 0-row vector and $h_{sj} = \infty$, $\forall j \neq s$.
4.  Set $i = s$.
5.  Mark node $i$.
6.  For every neighbor $j$ of of $i$:
7.      $t_v \leftarrow cap(V_{si}.A_{ij},\ \ c)$   //Employs (max, min) operator, if an element is greater than $c$, it is replaced with $c$
8.      $t_h \leftarrow h_{si} + 1$
9.      if $(t_v \geq c)$   //At least one element is greater than $c$, therefore the path can accommodate the request for capacity $c$
10.         if $(t_h < h_{sj})$ $\{V_{sj} \leftarrow t_v;\ h_{sj} \leftarrow t_h;\ nbr_{sj} \leftarrow i;\ count_j \leftarrow 1.\ \}$
11.         else if $(t_h = h_{sj})$
12.             $count_j \leftarrow count_j + 1;$
13.             if (RandomInteger(1, $count_j$) $= 1$) //Modify the entry with a probability of $1/count_j$.
14.                 $\{V_{sj} \leftarrow t_v;\ h_{sj} \leftarrow t_h;\ nbr_{sj} \leftarrow i.\ \}$
15. If no unmarked nodes left, go to Step 25.
16. Initialize $h \leftarrow \infty$.
17. For every unmarked node $j$ do:
18.     if $(h_{sj} < h)$ $\{\ k \leftarrow j;\ h \leftarrow h_{sj};\ t_v \leftarrow V_{sj};\ count \leftarrow 1.\ \}$
19.     else if $(h_{sj} = h)$ and $(V_{sj} > V)$ $\{\ k \leftarrow j;\ h \leftarrow h_{sj};\ t_v \leftarrow V_{sj};\ count = 1.\ \}$
20.     else if $(h_{sj} = h)$ and $(\max(V_{sj}) = \max(t_v))$
21.         $count \leftarrow count + 1$.
22.         if (RandomInteger$(1, count) = 1$) // Modify with a probability of $1/count$.
23.             $\{\ k \leftarrow j;\ h \leftarrow h_{sj};\ t_v \leftarrow V_{sj}.\ \}$
24. Set $i \leftarrow k$. Go to step 5.
25. Stop.
```

Fig. 8. Steps involved in path computation using Dijkstra's algorithm operating on the path information vector and hop length. The algorithm attempts to select the shortest path among those that are available, referred to as *Available Shortest Path* (ASP) algorithm.

$$\begin{aligned}
V_{11} &= \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \\
V_{12} &= \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 \end{bmatrix} \\
V_{13} &= \begin{bmatrix} 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 5 & 5 \end{bmatrix} \\
V_{14} &= \begin{bmatrix} 5 & 5 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 10 \end{bmatrix} \\
V_{15} &= \begin{bmatrix} 10 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 10 & 10 \end{bmatrix}
\end{aligned}$$

Fig. 9. Path information vector computed at the nodes along the path 1–2–3–4–5.

vector previous node, V_{ij} and the column vector of the link information matrix L_{jk} corresponding x_k:

$$\begin{aligned}
R_{jk} &= V_{ij} \times L_{jk}^T(x_k) && (14) \\
&= \begin{bmatrix} v_1 & \ldots & v_{K_j} \end{bmatrix} \circ \begin{bmatrix} l_{1x_k} & \ldots & l_{K_j x_k} \end{bmatrix} && (15) \\
&= \begin{bmatrix} v_1 \circ l_{1x_k} & \ldots & v_{K_j} \circ l_{K_j x_k} \end{bmatrix} && (16)
\end{aligned}$$

where $L_{jk}^T(x_k)$ denotes the transpose of the column vector corresponding to the column x_k of the matrix L_{jk} and the operator $\circ$ denotes the element-wise operation on the row vectors. Again, one could define different operators depending on the construction of the information matrix. Since the input trunk at node k is decided, the choices of output trunk at node j is also dictated by the channel occupancy of the channels that fall within $\Theta_{yx_k}^{jk}$. The output channel at node j can be selected in various ways using the ratio vector.

During the reverse pass, a sub-trunk is allocated on the path. As node 5 is the destination, it selects a trunk for the connection to terminate. We illustrate the random sub-trunk assignment here. We assume that the operator $\circ$ denotes integer multiplication[1].

The PIV at node 5 indicates that there are 20 possible sub-trunk assignments with each trunk being able to terminate 10 each. Hence, one of the two is chosen with equal probability. In general, if p_x sub-trunk assignments are possible on the path that would terminate the connection at the destination node at trunk x, then the trunk x is chosen with a probability $\frac{p_x}{\sum_{y=1}^{K_d} p_y}$, where K_d denotes the number of trunks at the destination node d. In the example considered here, one of the two trunks is selected with equal probability. Assume that the trunk chosen is $T2$. The node also selects the output trunk at its previous node. In order to select this, the ratio vector is computed as:

$$R_{45} = \left[\begin{array}{c} 10 \end{array}\right] \circ \left[\begin{array}{c} 1 \end{array}\right] = \left[\begin{array}{c} 10 \end{array}\right] \qquad (17)$$

In this case, as only one output trunk is available, it is selected. Hence, on link 4–5, a channel that belongs to trunk $F1$ of node 4 and trunk $T2$ of Node 5 is selected. Node 5 confirms the selection of output trunk $F1$ to Node 4 during the reverse pass in the network.

As the trunk assignment for the connection at node 4 is decided by node 5, node 4 chooses the output trunk at node 3 by computing a similar ratio vector as:

$$R_{34} = \left[\begin{array}{ccc} 5 & 5 & 5 \end{array}\right] \circ \left[\begin{array}{ccc} 1 & 1 & 0 \end{array}\right] = \left[\begin{array}{ccc} 5 & 5 & 0 \end{array}\right] \qquad (18)$$

R_{34} vector denotes the selection ratio for the three output trunks at Node 3. Note that although there are 5 possible paths that could end at trunk $W3$ at Node 3, there are no free channels on link 3–4 that fall within trunk W_3 of Node 3 and trunk F_1 of Node 4. This information is reflected in the selection ratio vector R_{34} as a zero entry corresponding to the ratio for trunk W3. Hence, trunk W1 or W2 is selected with equal probability. Assume that trunk W2 is selected in this case.

At Node 3, the vector R_{23} is computed as:

$$R_{23} = \left[\begin{array}{cc} 2 & 3 \end{array}\right] \circ \left[\begin{array}{cc} 1 & 1 \end{array}\right] = \left[\begin{array}{cc} 2 & 3 \end{array}\right] \qquad (19)$$

Node 3 selects the output trunk at Node 2 in the ratio of 2:3, i.e, trunk $T1$ is selected with a probability of 0.4 while trunk $T2$ is selected with a probability of 0.6. Assume that trunk $T1$ is chosen.

At Node 2, the vector R_{12} is computed as:

$$R_{12} = \left[\begin{array}{ccc} 1 & 1 & 1 \end{array}\right] \circ \left[\begin{array}{ccc} 1 & 1 & 0 \end{array}\right] = \left[\begin{array}{ccc} 1 & 1 & 0 \end{array}\right] \qquad (20)$$

One of the trunks W_1 or W_2 is chosen with equal probability. Assume that W_1 is chosen. This selection is sent to node 1 completing the sub-trunk assignment. Now, the path established for the connection can be written as a set of node-trunk pair assigned at each node on the path $\{(1, W_1), (2, T_1), (3, W_2), (4, F_1), (5, T_2)\}$ or equivalently as a set of link and sub-trunks pair on the path $\{(\ell_{12}, \Theta_{W_1 T_1}), (\ell_{23}, \Theta_{T_1 W_2}), (\ell_{34}, \Theta_{W_2 F_1}), (\ell_{45}, \Theta_{F_1 T_2})\}$. Any channel belonging to the sub-trunk assigned at a link can be chosen for establishing the channel as every node has full-permutation switching capability within a trunk.

It can be observed that such a trunk selection strategy selects with uniform probability a possible sub-trunk assignment on the path. For example, if we employ information matrix shown in 7, operator $(\times, +)$, operator $\circ$ set to multiplication, and random channel assignment within a chosen trunk on a link, then the resulting channel assignment algorithm selects a channel uniformly from the set of all possible channel assignments possible on the path.

Several other trunk selection approaches can be employed, such as selecting the trunk that has the minimum or maximum value in the selection ratio vector, which would have effect similar to packing or spreading the connections across the available sub-trunk assignments in the path. In order to implement a first-fit sub-trunk assignment, the first available trunk is chosen from the ratio vector. The framework may then be used to implement and evaluate the performance of the different path selection and sub-trunk (hence, channel) assignment schemes on different network architectures.

G. *Modeling blocking trunk switches*

The framework assumes that the trunk switch employed at every node has full-permutation switching capability. This implies that any channel on a trunk on a link can be switched to any other channel on any output link but within the same trunk. In such a scenario, a connection cannot be accommodated on a trunk only due to lack of capacity on the trunk and not due to switching. All-optical implementation of full-permutation switches would require a large number of stages of switching, hence may not be practical due to power and synchronization issues. Hence, simpler but blocking switching architectures that involves fewer stages of switching may be considered for implementation.

For example, consider the node architecture shown in Fig. 10. The node is assumed to have three incoming and outgoing links. Different trunks are de-multiplexed from the link. The trunks from different links are then switched using a channel-space-channel switch. The first and the third stages of switching have full-channel interchangers (FCI) that can convert any channel on the trunk on a link to any other channel on the same trunk on the same link. The FCI stages at the input and the output of the node allows the node to operate in an autonomous manner. Connections are assigned channels on the links. These channels are mapped by the FCI to channels within the node. The node could rearrange the connections for switching at the input and output of the switch as long as the channels allocated to the connections on the links (before the first FCI stage and after the last FCI stage) are retained.

For example, consider a wavelength-level grooming node in a WDM/TDM network. Assume that the link has three fibers, three wavelengths per fiber, and two time slots per wavelength. A wavelength-level grooming node would view

---

[1] The operator $\circ$ is the same as operator $\otimes$ since the element-wise operation that is evaluated here is similar to the matrix-vector multiplication employed for computing path information vector.

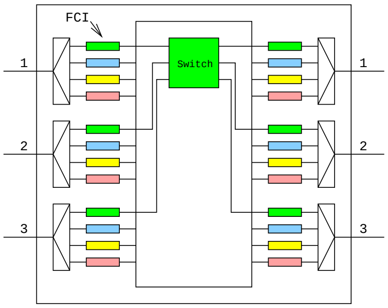

Fig. 10. Channel-Space-Channel switching architecture at a node in a TSN.

the link as three trunks with six channels in each. The channel-space-channel architecture would refer to a time-space-time switch employed for every wavelength for a wavelength-level grooming node. Consider a channel that belongs to an input link, say $(l, f, w, t)_{in}$. At a wavelength level grooming node, the channel is mapped in the first stage of switching to another channel, say $(l, f, w, t)_1$, where $l_{in} = l_1$ and $w_{in} = w_1$. The space switch maps the channel $(l, f, w, t)_1$ to another channel $(l, f, w, t)_2$, where $l_1 \neq l_2$, $f_1 = f_2$, $w_1 = w_2$, and $t_1 = t_2$. The last stage of FCI maps the channel $(l, f, w, t)_2$ to $(l, f, w, t)_{out}$ where $l_2 = l_{out}$ and $w_2 = w_{out}$. It is to be noted that the channel-space-channel switching employed here is a blocking switch. Therefore, the number of channels that could be switched from an input link to an output link could be less than the minimum number of free channels on the links.

Every node maintains a vector for every input and output link pair at the node, referred to as *switching constraint vector*. This vector at a node i has a dimension $1 \times K_i$, denoted by $X_{\ell_{in}\ell_{out}} = [x_1, x_2, ..., x_{K_i}]$. Every element in this vector denotes the number of channels that could be switched from input link ℓ_{in} to ℓ_{out}. Now, consider a path from node i to k through j. Assume that node j employs channel-space-channel switching architecture. The path information vector that is forwarded from node j to node k, denoted by V_{ij}:

$$V_{ij} = (V_{ii}.L_{ij}) \circ X_{i-j,j-k} \tag{21}$$

where $\circ$ refers to an element-wise operator on the two vectors. This operator could be chosen in many ways depending on the information collected on the path. For example, if the available capacity is stored in the matrices and the operator set (min, max) is employed to identify the maximum capacity that could be routed on the path without splitting the connection, then the operator $\circ$ can be made to be element-wise minimum. Thus, information on a path could be obtained where intermediate nodes need not have full-permutation switching capability.

It is worth noting at this point that the FCI at the input and output stages of switching eliminates channel-continuity constraint within a trunk. Therefore, it is sufficient to carry information about the number of channels that could be switched by the node. However, if the FCI stages are not present, then exact information on which channels can be switched must also be considered. In such a case, the elements in the switching constraint vector and the link information matrix will be matrix (or vectors) themselves.

H. Modeling trunk-routing networks

The framework for connection establishment assumes that switches employed for every trunk at a node has the capability of switching individual channels across different links but within the same trunk. In other words, different channels on a trunk at a link can be switched to channels on different links but in the same trunk. However, some switches may have further restricted switching capability in which an entire trunk is switched from one link to another. For example, consider a wavelength-routed WDM/TDM network. A wavelength-routing node that does not employ wavelength conversion would switch an entire wavelength from one link to another. Another example of such a switching node is a photonic slot routing node[25] that would switch all the wavelengths in a time slot from one link to another. Therefore, either all or none of the channels are switched from one link to another. Such a node in which either all or none of the channels are switched from one link to another are referred to as *trunk-routing* nodes.

In order to model a trunk-routing node, we employ switching constraint vector. The individual elements are computed depending on whether node i is a source, destination, or an intermediate node. If node i is a source, then the input link refers to the link through which the local traffic is inserted. In such a case, the information corresponding to a trunk x in the switching constraint vector is set to 1 if the node is allowed to transmit on the corresponding trunk, otherwise set to 0. If node i is the destination node, then the elements of the switching constraint vector are set to 1 if the node is allowed to receive on the specific trunk. If node i is an intermediate node, then the value corresponding to trunk x is set to 1 under any of the following conditions: (a) trunk x is routed from link ℓ_{in} to ℓ_{out}; and (b) node i can receive on trunk x from link ℓ_{in} and transmit on trunk x on link ℓ_{out}. Otherwise, it is set to 0. It is to be noted that in case that trunk x from link ℓ_{in} is routed to a link other than ℓ_{out}, the value is set to 0.

The switching constraint vector is maintained at a node and need not be exchanged with other nodes in the network. Upon receiving a connection establishment request, the node computes its path information vector as described in the earlier section. Before forwarding it to the neighboring node in the network, the path information vector and the switching constraint vector are combined using element-wise multiplication. The resultant vector denotes the updated path information vector that also takes into account the switching capability at a node.

Consider the example network shown in Fig. 5. Assume that node 5 is photonic slot routing node and that the trunk T1 on link 4–5 is routed to another link (not shown in figure). Therefore, node 5 cannot receive information on trunk T1. Assume that node 5 is receiving information on trunk T2. This information is available at node 5 as the switching constraint vector, denoted by $[0 \;\; 1]$. Consider the path information vector obtained on the path 1-2-3-4-5 at Node 5 using connectivity information and employing the operator set $(\times, +)$. The path information vector $[10 \;\; 10]$ indicates that there are 10 possible trunk assignments that would end on trunk T1 and T2 each. When this information is combined with the switching constraint vector at node 5, we obtain the resultant vector as $[0 \;\; 10]$ indicating that there are no possible trunk assignments that could end on trunk $T1$ at node 5.

I. Destination initiated connection establishment

The sections thus far assumed that the connection request is initiated by the source node. However, in some scenarios, such as video-on-demand, the connection request may be initiated by the destination. In destination-initiated communications, the connection establishment may be performed in two ways: (1) three-pass mechanism, or (2) two-pass mechanism. In the three-pass mechanism, the receiver sends a connection request to the source during the first pass. In the second pass, the source node initiates a connection request to the destined receiver. In the third phase, the channel establishment is performed on the reverse path. The connection establishment in such a case could be performed as described in the previous sub-sections. In the two-pass mechanism, the path information from the source to destination is captured when the connection request is sent by the receiver to the source as the first phase. In the second phase, the source decides a path and the channel assignment is performed along the path from the source to destination. It is to be noted that the path information collected in the first pass is for a directed path from the source to destination, although the connection request is sent from the destination to the source.

Consider a path from node i to node k passing through node j. Let V^*_{jk} denote the path information vector at node j for the path from node j to k. The superscript of $*$ denotes that this vector is computed at node j. Let P_{ij} denote the path information matrix from node i to node j. The path information vector at node i from node i to node k, denoted by V^*_{ik}, is computed as:

$$V^*_{ik} = V^*_{jk} \; P^T_{ij} \tag{22}$$

We illustrate the two-pass mechanism here with an example. Consider the network in Fig. 4. Assume that node 5 initiates a connection request. Node 5 initializes its PIV as $V^*_{55} = [1 \; 1]$. Assume that the link information matrices represent the connectivity information and arithmetic operator set $(\times, +)$ is used for computing the PIVs. The PIVs computed at various nodes as the request traverses from node 5 to node 1 are shown in Fig. 11. V^*_{15} denotes the PIV for the path 1–2–3–4–5.

$$\begin{aligned}
V^*_{55} &= \begin{bmatrix} 1 & 1 \end{bmatrix} \\
V^*_{45} &= V^*_{55}.L^T_{45} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \end{bmatrix} \\
V^*_{35} &= V^*_{45}.L^T_{34} = \begin{bmatrix} 2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 0 \end{bmatrix} \\
V^*_{25} &= V^*_{35}.L^T_{23} = \begin{bmatrix} 2 & 2 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 4 \end{bmatrix} \\
V^*_{15} &= V^*_{25}.L^T_{12} = \begin{bmatrix} 4 & 4 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 8 & 8 & 4 \end{bmatrix}
\end{aligned}$$

Fig. 11. Path information vector obtained along the path 1–2–3–4–5 at different nodes when the request is initiated by node 5.

An element p_x of the PIV computed at a node indicates the number of distinct sub-trunk assignments possible on the path from that node to the destination. More than one request may be sent from the destination along a candidate set of paths and the source node may select a path depending on the path selection algorithm employed in the network. Once the path is selected, the sub-trunk assignment starts at the source node and proceeds in the forward direction of the path in a way similar to that explained in Section IV-F. The selection ratio vector at node j for a path from node i to node k through node j is obtained as:

$$\begin{aligned}
R^*_{jk} &= V_{jk} \circ L_{ij}(x_i) && (23) \\
&= \begin{bmatrix} v_1 & \ldots & v_{K_j} \end{bmatrix} \begin{bmatrix} l_{x_i 1} & \ldots & l_{x_i K_j} \end{bmatrix} && (24) \\
&= \begin{bmatrix} v_1 \circ l_{x_i 1} & \ldots & v_{K_j} \circ l_{x_i K_j} \end{bmatrix} && (25)
\end{aligned}$$

where $L_{ij}(x_i)$ is the row vector corresponding to the row x_i of the matrix L_{ij}.

The techniques described in earlier sections to account for blocking nature of the switch at a node and trunk-routing nodes may be combined with the above PIV computation of destination-initiated requests as well.

J. Multicast tree establishment

Multicast connection establishment is accomplished using the destination initiated request approach mentioned above. Consider the network shown in Fig. 12 in which node 0 is the multicast source and nodes 5 and 7 are the destination. Fig. 13 shows the expanded view of the network.

The multicast tree that needs to be established spans all the links and requires node 1 to route the incoming traffic along two paths. In order to facilitate multicast connections, the incoming signal must be copied and transmitted along two different paths. In order to provide a cost-effective solution, the copying may be restricted within a trunk. In such a case, the trunk in which the connection terminates at node 1 must reach all the destinations. In MICRON, such a multicast path is established in three steps: (1) The multicast request is forwarded along the tree, (2) the path information vector on the reverse path is obtained, and (3) the sub-trunk assignment is made along the forward path.

In Step 1, node 0 sends the request along the tree to nodes 5 and 7. Node 1 is aware of the branching, hence will merge the two path information vectors for the two destinations. The PIV for the path 1–2–3–4–5 at node 1 is obtained as shown in Fig. 11 as $V^*_{15} = [8 \;\; 8 \;\; 4]$. Similarly, the PIV for the path 1–6–7 at node 1 is obtained as $V^*_{17} = [1 \;\; 0 \;\; 1]$. It has to be noted that as the PIV is computed at node 1, the vectors along different paths will be of the same dimension (same number of columns – corresponding to the number of trunks at node 1) although the destination nodes may have different switch architectures. From the PIVs it is clear that node 5 may be reached from

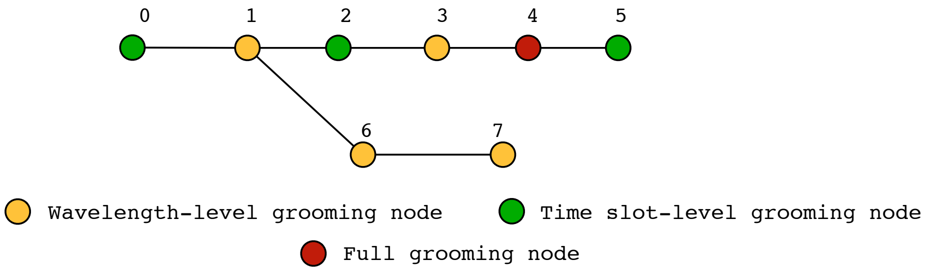

Fig. 12. An example network where node 0 is the source of a multicast connection that is destined to nodes 5 and 7

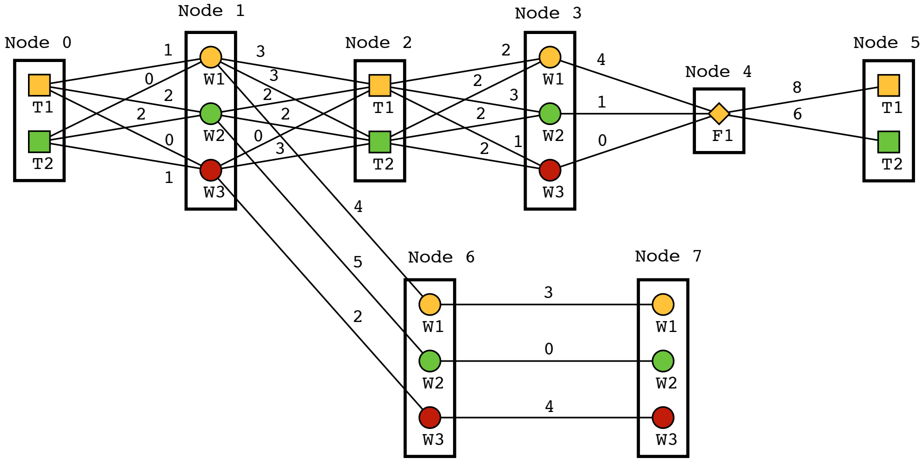

Fig. 13. Expanded view of the network where node 0 is the source of a multicast connection that is destined to nodes 5 and 7.

node 1 by starting on any of the trunks. However, node 7 may be reached only if the connection at node 1 is routed along trunk $W1$ or $W3$.

The tree information vector (TIV) at node 1, denoted by $V_1^*\{5,7\}$ is obtained by combining the path information vectors as:

$$V_1^*\{5,7\} = V_{15}^* \circ V_{17}^* \tag{26}$$

where $\circ$ represents an element-wise operation. When the nodes employ intra-trunk copying facility, the element-wise operator must satisfy one property: if any of the element is zero, the result must be zero. There is more than one operator that may satisfy this constraint. In this example, we use integer multiplication. The TIV at node 1 is obtained as $[8 \quad 0 \quad 4]$. The TIV at node 0 is computed as:

$$\begin{aligned} V_{0\{5,7\}}^* &= V_{1\{5,7\}}^* . L_{01}^T \\ &= \left[\begin{array}{ccc} 8 & 0 & 4 \end{array} \right] \left[\begin{array}{ccc} 1 & 2 & 0 \\ 0 & 2 & 1 \end{array} \right] \\ &= \left[\begin{array}{cc} 8 & 8 \end{array} \right] \end{aligned}$$

The TIV at node 1 indicates that the multicast tree may be established starting at either trunk. However, the selection of a trunk will affect the selection ratio vector computed at node 1. For example, if trunk $T1$ is selected at node 0, the selection ratio vector at node 1 will be $R_{1\{5,7\}}^* = [8 \quad 0 \quad 0]$, thus forcing the multicast connection to be established through trunk W_1 at node 1. Similarly, if trunk $T2$ is chosen at node 0, the multicast connection will be established using trunk W_3 at node 1.

The techniques for modeling blocking switches and trunk-routing switches may be combined with the multicast connection establishment techniques as well.

K. Reductions to simpler networks

The MICRON framework applied to a WDM grooming network where all the nodes in the network are wavelength-level grooming nodes results in the link information matrix being a diagonal matrix. In such a scenario, the matrix can be reduced to a vector, thus simplifying the operations to be performed to obtain path information. Such a reduction of the MICRON framework has been adopted in [26] and [27]. In case of networks where all nodes are equipped with full-grooming capability, hence every node views each link as one trunk, the MICRON framework can further be reduced to a single metric. The proposed methodology then reduces to the well-known information aggregation methods established in traditional networks.

The framework may be easily extended with other metrics such as delay, cost, etc. to implement a plethora of routing and channel assignment algorithms based on the information collected using link-state protocols in WDM grooming networks with heterogeneous grooming capability.

V. CONCLUSIONS

In this paper, we develop a framework for connection establishment in a WDM grooming network with heterogeneous switching architectures. We illustrate with examples the various information that could be collected from the links and various operators that could be used to obtain information along a path. The information collected may be used to select a path dynamically depending on the network status. We complete the framework by providing a generic channel assignment procedure that could be employed to implement different channel assignment schemes.

REFERENCES

[1] D. K. Hunter and D. G. Smith, "New architectures for optical TDM switching," *IEEE/OSA Journal of Lightwave Technology*, vol. 11, no. 3, pp. 495–511, March 1993.

[2] H. F. Jordan, D. Lee, K. Y. Lee, and S. V. Ramanan, "Serial array time slot interchangers and optical implementations," *IEEE Transactions on Computers*, vol. 43, no. 11, pp. 1309–1318, November 1994.

[3] J. Yates, J. Lacey, and D. Everitt, "Blocking in multiwavelength TDM networks," in *4th International Conference on Telecommunication Systems, Modeling, and Analysis*, Nashville, Tennessee, USA, March 1996, pp. 535–541.

[4] A. L. Chiu and E. H. Modiano, "Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks," *Journal of Lightwave Technology*, vol. 18, no. 1, pp. 2–12, January 2000.

[5] O. Gerstel, R. Ramaswami, and G. H. Sasaki, "Cost-effective traffic grooming in WDM rings," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 618–630, October 2000.

[6] J. Wang, W. Cho, V. Rao Vemuri, and B. Mukherjee, "Improved approaches for cost-effective traffic grooming in WDM ring networks: ILP formulations and single-hop and multihop connections," *Journal of Lightwave Technology*, vol. 19, no. 11, pp. 1645–1653, November 2001.

[7] A. Jarray and B. Jaumard, "Exact ILP solution for the grooming problem in WDM ring networks," in *Proceedings of IEEE International Conference on Communications (ICC)*, Seoul, Korea, May 2005, vol. 3, pp. 1708–1712.

[8] H. Huang and J. A. Copeland, "Hybrid wavelength and sub-wavelength routed optical networks," in *Proceedings of IEEE GLOBECOM*, San Antonio, TX, USA, November 2001, vol. 4, pp. 25–29.

[9] P. Prathombutr, J. Stch, and E. K. Park, "An algorithm for traffic grooming in WDM optical mesh networks," in *Proceedings of IEEE International Conference on Computer Communications and Networks (ICCCN)*, Dallas, TX, USA, October 2003, pp. 405–411.

[10] J. Hu and B. Leida, "Traffic grooming, routing and wavelength assignment in optical WDM mesh networks," in *Proceedings of IEEE INFOCOM*, Hong Kong, March 2004, pp. 495–501.

[11] C. Xin, C. Qiao, and S. Dixit, "Traffic grooming in mesh WDM optical networks – performance analysis," *IEEE Journal of Selected Areas in Communications*, vol. 22, no. 9, pp. 1658–1669, November 2004.

[12] W. Yao and B. Ramamurthy, "Rerouting schemes for dynamic traffic grooming in optical WDM mesh networks," in *Proceedings of IEEE GLOBECOM*, Dallas, TX, USA, November-December 2004, vol. 3, pp. 1793–1797.

[13] A. Mokhtar and M. Azizoglu, "Adaptive wavelength routing all-optical networks," *IEEE Transactions on Networking*, vol. 6, no. 2, pp. 197–206, April 1998.

[14] H. Zang, J. Jue, L. Sahasrabuddhe, R. Ramamurthy, and B. Mukherjee, "Dynamic lightpath establishment in wavelength-routed WDM networks," *IEEE Communications*, pp. 100–108, September 2001.

[15] G. Li and R. Simha, "On the wavelength assignment problem in multifiber wdm star and ring networks," *IEEE/ACM Transactions on Networking*, vol. 9, pp. 60–68, February 2001.

[16] B. Wen and K. M. Sivalingam, "Routing, wavelength and time-slot assignment in time division multiplexed wavelength-routed optical WDM networks," in *Proceedings of IEEE INFOCOM'02*, New York, NY, USA, June 2002, pp. 1442–1450.

[17] K. Zhu, H. Zhu, and B. Mukherjee, "Traffic engineering in multi-granularity heterogeneous wdm mesh networks through dynamic traffic grooming," *IEEE Network*, vol. 17, no. 2, pp. 8–15, March-April 2003.

[18] H. Zhu, K. Zang, K. Zhu, and B. Mukherjee, "A novel generic graph model for traffic grooming in heterogeneous wdm mesh networks," *IEEE/ACM Transactions on Networking*, vol. 11, no. 2, pp. 285–299, April 2003.

[19] L. Li and A.K. Somani, "Fiber requirement in multifiber WDM networks with alternate-path routing," in *Proceedings of the International Conference on Computer Communications and Networks*, Boston, MA, USA, October 1999, pp. 338–343.

[20] E. Karasan and E. Ayanoglu, "Effects of wavelength routing and selection algorithms on wavelength conversion gain in WDM optical networks," *IEEE/ACM Transactions on Networking*, vol. 6, no. 2, pp. 186–196, April 1998.

[21] C.-F. Hsu, T.-L. Liu, and N.-Fu. Huang, "Performance of adaptive routing strategies in wavelength-routed networks," in *Proceedings of IEEE International Conference on Performance, Computing, and Communications*, Phoenix, AZ, USA, April 2001, pp. 163–170.

[22] R. Srinivasan and A. K. Somani, "A generalized framework for analyzing time-space switched optical networks," *IEEE Journal of Selected Areas in Communications: Special Issue on WDM-based Network Architectures*, pp. 202–215, January 2002.

[23] R. Perlman, *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*, Addison-Wesley, 2nd edition, 1999.

[24] R. Srinivasan, "A generalized framework for analyzing time-space switched optical networks," *PhD Dissertation, Iowa State University*, 2002.

[25] V. Elek, A. Fumagalli, and G. Wedzinga, "Photonic slot routing: a cost effective approach to designing all-optical access and metro networks," *IEEE Communications Magazine*, vol. 39, pp. 164–172, November 2001.

[26] R. Srinivasan and A. K. Somani, "Request-specific routing in WDM grooming networks," in *Proceedings of IEEE International Conference on Communications (ICC 2002)*, New York, NY, USA, April 2002, pp. 2876–2880.

[27] M. Sridharan, R. Srinivasan, and A. K. Somani, "Dynamic routing with partial information in mesh-restorable optical newtorks," in *Proceedings of Sixth Working Conference on Optical Networks Design and Modelling*, February 2002, pp. 327–346.

Srinivasan Ramasubramanian (SM'97–M'02) received the B.E. (Hons.) degree in Electrical and Electronics Engineering from Birla Institute of Technology and Science (BITS), Pilani, India, in 1997, and the Ph.D. degree in Computer Engineering from Iowa State University, Ames, in 2002. He is a co-developer of the Hierarchical Modeling and Analysis Package (HIMAP), a reliability modeling and analysis tool, which is currently being used at Boeing, Honeywell, and several other companies and universities. His research interests include architectures and algorithms for optical, wireless, and sensor networks, multipath routing, fault tolerance, system modeling, and performance analysis. He has served as the TPC Co-Chair of BROADNETS 2005 conference and is an editor of the Springer Wireless Networks Journal.

Arun K. Somani (M'83–SM'88–F'99) received the M.S.E.E. and Ph.D. degrees in electrical engineering from McGill University, Montreal, Canada, in 1983 and 1985, respectively. He is currently Jerry R. Junkins Endowed Chair Professor, Distinguished Professor, and Department Chair of Electrical and Computer Engineering at Iowa State University, Ames. He worked as Scientific Officer for the Government of India, New Delhi, from 1974 to 1982 and as a faculty member at the University of Washington, Seattle, from 1985 to 1997 in the electrical engineering and computer science and engineering departments, where he was promoted to Full Professor in September 1995. His research interests are in the area of design of fault tolerant computing and networking systems, computer interconnection networks, developing scalable architectures and algorithms for access and traffic grooming, and to manage, control, and deliver dependable service efficiently for network employing WDM-based optical fiber technology, parallel computer system architecture, and applications of sensors in critical infrastructure protection such as power grid. He is the chief architect of an antisubmarine warfare system (developed for Indian navy) and Meshkin fault tolerant computer system architecture (developed for the Boeing Company). He has also developed several robust interconnection topologies, architected, designed, and implemented a 46-node multi-computer cluster-based system, Proteus, using a large grain message-passing model and separate data and control planes networks that use fiber optic communication links. Prof. Somani has served on numerous program committees of various conferences in his research areas. He has served as an Editor of IEEE Transactions on Computers, Associate Editor off IEEE/ACM Transactions on Networking and Elsevier Computer Networks Journal. He was the General Chair of IEEE Fault Tolerant Computing Symposium 1997, Technical Program Committee Chair of the International Conference on Computer Communications and Networks 1999 and OPTICOMM 2003, and General Chair of BroadNets 2005. He has served as an IEEE distinguished visitor and IEEE distinguished tutorial speaker. He was elected as a Distinguished Engineer of ACM in 2006 and has been elected a Fellow of IEEE for his contributions to theory and applications of computer networks.