Algorithms for Intermediate Waveband Switching in Optical WDM Mesh Networks

Ajay Todimala\textsuperscript{1} and Byrav Ramamurthy\textsuperscript{2}

\textsuperscript{1}OIT-Mid-Atlantic Crossroads, University of Maryland-College Park
College Park MD 20740 U.S.A
ajayt@maxgigapop.net

\textsuperscript{2}Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln NE 68588-0115 U.S.A.
byrav@cse.unl.edu

Abstract—Waveband switching is a technique that allows multiple wavelengths to be switched together as a single unit. Waveband switching technique has been proven to reduce the switch sizes considerably in large networks. Aggregation of wavelengths into wavebands and dis-aggregation of wavebands back to wavelengths can be done at end-nodes or intermediate-nodes. Most of the research on waveband switching has considered source-to-end switching. In intermediate waveband switching, aggregation and/or dis-aggregation can be done at an intermediate node. In the context of intermediate waveband switching, the problem of grooming wavelengths such that the number of ports saved is maximized is non-trivial. Recent research which considered intermediate waveband switching focused on routing and wavelength assignment problem such that the port saving is maximized. In this work we focus on the problem of intermediate waveband switching considering static traffic and assuming that routing and wavelength assignment is known. We define two intermediate waveband grooming policies, intermediate-to-destination waveband switching (ITD-WBS) and both-end-to-intermediate waveband switching (BETI-WBS), depending on where along the path aggregation/disaggregation of wavebands is done. We present greedy algorithms to compute wavebands for the two intermediate waveband grooming polices and analyze their computational complexities.

I. INTRODUCTION

Waveband switching is technique that allows the grouping of multiple wavelengths into a single waveband and switching them as a single unit. Waveband switching reduces the number of ports in the switches. Multiple lightpaths having common segments can be grouped and switched as a single unit. Depending upon where (along the path) the aggregation of wavelengths into wavebands and the disaggregation of a waveband into wavelengths is done, several variations are possible. The source-to-end waveband switching (ETE-WBS) is the simplest form of waveband switching where multiple connections between a source and destination nodes are grouped into a waveband. The intermediated waveband switching can be further classified into several types depending on aggregation and/or dis-aggregation at intermediate nodes. In intermediate-to-destination waveband switching (ITD-WBS) the grooming of wavelengths into a waveband is done at an intermediate node and disaggregation into individual wavelengths is done at the destination node. The intermediate-to-destination waveband switching can be applied to multiple connections having the same destination. In source-to-destination waveband switching (STI-WBS) the grooming is done at the source node and the disaggregation is done at an intermediate node. The second waveband grooming policy is both-end-to-intermediate (BETI) waveband switching where both STI and ITD waveband switching are allowed at the same time. Waveband granularity is defined as the number of wavelengths that can be grouped or aggregated into a waveband. An optical switch is said to support uniform waveband switching if the granularity of all the wavebands is an arbitrary constant $g$. In contrast, for an optical switch that supports non-uniform wavebands, waveband granularities vary in a set i.e., \{\text{$g_1$}, $g_2$, …, $g_k$\}.

Intermediate waveband switching was first studied in [3] where it is shown that it reduces the switch sizes by a factor of two or more. The study of waveband switching in paper [4] showed that it further reduces the number of ports in the switching nodes by at least an order of magnitude. In paper [5], the authors study the problem of intermediate wavelength and waveband switching in a network that supports dynamic traffic requests. The authors of paper [6] addressed the problem of waveband switching focusing mostly on the routing and wavelength assignment problem for maximizing the number of ports saved using waveband switching.
They present a heuristic, its performance analysis and an integer linear programming formulation to the problem. In this work we address the problem of waveband grooming assuming that the routing and wavelength assignment is already known.

The rest of the paper is organized as follows. In Section II we provide the problem definitions and present our notations. In Section III we present the algorithms for solving the uniform/non-uniform waveband switching which allows only intermediate-to-destination grooming. In Section IV we present the algorithms for solving the uniform/non-uniform waveband switching with both intermediate-to-destination and source-to-destination grooming. In Section V we present the conclusions.

II. PROBLEM STATEMENT AND NOTATIONS

A. Problem Statement

In this work we consider the following problem. We are given a routing and wavelength assignment of a set of static connection requests in a graph \( G = (V, E) \). The problem is to groom wavelengths into wavebands such that the number of ports saved is maximized. Two variations of the problem exist depending upon the type of waveband switching supported in the network i.e., uniform or non-uniform waveband switching. The problem also has variations depending upon the waveband aggregation and dis-aggregation allowed in the network. In this work we consider ITD-WBS and STI-WBS. We formalize each of the variations of the network. In this work we consider ITD-WBS and present our notations. In Section III we present the algorithms for solving the uniform/non-uniform waveband switching only occurs among paths within a

B. Notations

A waveband \( B \) of granularity \( g \) is denoted by \( (Q, s, d, g) \) where \( Q = \{b_1, b_2, \ldots, b_m\} \) is a set of routed-demand \( (p_i, b_i) \), where \( p_i \) is a routed-demand between a pair of nodes and \( b_i \) is the number of units (lightpaths) of the routed-demand. Nodes \( s \) and \( d \) are aggregating and disaggregating nodes. The sum of \( b_i \) of all the routed-demands in the waveband \( B \) of granularity \( g \) is at most the waveband granularity \( g \) i.e., \( \sum_{1 \leq i \leq m} b_i \leq g \). The number of units of demand of the routed-demand \( p \), which is yet to be routed, after aggregating \( b \) units into a waveband \( B \) is called the residual demand \( r \) where \( c = b + r \).

The number of wavelength ports used by a waveband of length \( L \) and granularity \( g \) is \( 4g + 2(L + 1) \). The number of ports required for routing \( g \) wavelength level connections of length \( L \) is \( 2g(L + 1) \). Therefore the number of ports saved by a waveband route of length \( L \) and granularity \( g \) is \( 2(L + 1)(g - 1) - 4g \).

III. ALGORITHM FOR ITD-UWBS PROBLEM

In this section we present algorithms for the ITD-UWBS problem.
A. Initialization

In this section we present an algorithm to transform an instance of the ITD-UWBS problem into a destination-rooted capacitated tree \( T \). The input to the algorithm consists of a set of \( \kappa \) paths \( P_d = \{p_1, p_2, \ldots, p_m\} \) in the graph \( G \) all having the same destination \( d \) and their corresponding path capacities \( C = \{c_1, c_2, \ldots, c_m\} \). The algorithm for initialization is outlined by Algorithm 1. The algorithm computes a sub-graph \( T \) using paths in \( P_d \). Let the edges of the sub-graph \( T \) be directed from destination node \( d \) to corresponding source nodes as leaf nodes. The graph \( T \) is now transformed into a tree by deleting cycles in \( T \) and modifying the paths accordingly. Let us illustrate the transformation into tree using an example instance shown in Figure 1. Each node \( u \) in the tree \( T \) is associated with a variable \( R_u \) that represents the sum of the residual capacities \( r_j \) of all the paths \( p_j \) that have a common segment from node \( u \) to destination node \( d \). The Algorithm 1 computes height \( h_i \) of each node in the tree \( T \) where the root node \( d \) has height 0. The Algorithm 1 now initializes the residual capacity of each leaf node \( j \) of the tree \( T \) with the capacity of the path \( p_i \) where \( j \) is the source node of the path \( p_i \). The algorithm then computes the residual capacity of each intermediate node in the tree as the sum of the residual capacities of its child nodes. Fig. 2 shows the tree computed and initialized by the Algorithm 1 for an instance of the ITD-UWBS problem.

Algorithm 1

1: Input: \((G, P_d, C)\)
2: Output: Destination-rooted capacitated tree \( T \)
3: compute graph \( T \) using paths in the set \( P_d \)
4: transform \( T \) into a tree by deleting cycles in \( T \) and modifying paths accordingly
5: compute the height \( h_i \) for each node \( i \) in the tree \( T \)
6: initialize the residual capacity \( R_i \) of the leaf node \( j \) to \( n_i \) where \( j \) is the source node of the path \( p_i \)
7: compute the residual capacity \( R_u \) of each intermediate node \( i \) as the sum of the residual capacities of its child nodes

B. Algorithm to solve the ITD-UWBS Problem

The algorithm for solving the ITD-UWBS problem is a greedy algorithm and is outlined by Algorithm 2. The input to the algorithm consists of a set of \( \kappa \) paths \( P_d = \{p_1, p_2, \ldots, p_m\} \) all having the same destination \( d \) and their corresponding path capacities \( C = \{c_1, c_2, \ldots, c_m\} \) and an integer \( g \) representing waveband granularity. The algorithm outputs a set of wavebands \( B = \{B_1, B_2, \ldots\} \).

The Algorithm 2 constructs a destination-rooted capacitated tree \( T \) using the Algorithm 1 and input \( P_d \) and \( C \). The Algorithm 2 iterates with decreasing tree height. In each iteration with height \( i \) the algorithm computes all possible wavebands of length \( i \). To compute a waveband, it selects a node \( u \) of height \( i \) and residual granularity at least \( g \). Now, if \( (\{S_u = 2(i+1)(g-1) - 4g > 0\} \) i.e., the number of ports saved by forming a waveband \( B = (Q, u, d, g) \) from node \( u \) to node \( d \) of granularity \( g \) is non-negative then we form a waveband \( B \) and add it to the set of wavebands \( B \). We update residual capacities of all the nodes along each of the path \( p \in Q \).

Let us illustrate the working of the Algorithm 2 on the instance shown in Fig. 2. Fig. 3(a) shows the waveband \( B_1 = (Q_1, s_1, d, 4) \) where \( Q_1 = \{\{p_1, 4\}\} \), computed by the Algorithm 2 in the first iteration. The residual capacities of all the nodes along the waveband \( B_1 \) are updated accordingly and are shown in Fig. 3(b). In the second iteration, the Algorithm 2 computes the waveband \( B_2 = (Q_2, s_1, d, 4) \) where \( Q_2 = \{\{p_2, 4\}\} \). The Algorithm 2 iterates until there exists no node in the tree with residual granularity greater than \( 4 \) and a non-negative \( S \) value. The Algorithm 2 computes five wavebands as shown in Fig. 3.

Algorithm 2

1: Input: \((G, P_d, C, g)\)
2: Output: Waveband set \( B \)
3: run Algorithm 2 on input \((G, P_d, C)\)
4: \(\text{for } i = h; i \leq 2; i --\) do
5: \(\text{for each } u \text{ where } h_u = i \text{, and } R_u \geq g \) do
6: \(\text{if } (\{S_u = 2(i+1)(g-1) - 4g > 0\} \text{ then})\)
7: \(\text{form waveband } B = (Q, u, d, g) \text{ from node } u \text{ to node } d \text{ and add to } B\)
8: \(\text{update the residual capacities } R_u \text{ of all the nodes along the paths included in the waveband}\)
9: \(\text{end if}\)
10: \(\text{end for}\)
11: \(\text{end for}\)

Let us discuss the complexity of the Algorithm 2. The time taken to compute the tree \( T \) by Algorithm 1 assuming that the graph computed by the Algorithm 1 in Step 3 does not have cycles is \( O(nm) \) where \( n \) is the number of nodes and \( m \) is the number of paths in the set \( P \). Consider the following simple modification to Algorithm 2 where instead of forming a single waveband in Step 7 from node \( u \) to node \( d \), we form \( R_u/g \) number of wavebands. With this modification each node in the tree is examined at most once. After forming the waveband, the algorithm needs
to update the residual capacities of the nodes along the paths in the wavebands. Therefore for each node examined we need to update residual capacities of at most $O(m \log n)$ nodes. Therefore the complexity of the Algorithm 2 is $O(mn \log n)$. In the next section we present algorithms for waveband switching where both intermediate-to-destination and source-to-intermediate grooming is allowed.

IV. ALGORITHM FOR BETI WAVEBAND SWITCHING

In this section we present an algorithm for a variation of intermediate waveband switching where aggregation and dis-aggregation can be done at both the end nodes i.e., source and destination nodes. We consider uniform wavebands. Using the Algorithm 3, we transform an instance of the BETI problem into a destination-rooted capacitated tree $T_s$ and a source-rooted capacitated tree $T_t$. All the edges in the graph $T_t$ (correspondingly, $T_s$) are directed from destination to source nodes (correspondingly, from source to destination nodes) of the paths in $P$.

Algorithm 3 The Initialization Algorithm for the BETI problem.

1. **Input:** $(G, P, C)$
2. compute graphs $T_t$ and $T_s$ using paths in the set $P$
3. add super destination node $d$ and super source node $s$ to trees $T_t$ and $T_s$ respectively
4. add edges from node $d$ to all the destination nodes in tree $T_t$
5. add edges from node $s$ to all the source nodes in tree $T_s$
6. transform $T_t$ and $T_s$ into a trees by deleting cycles in $T_t$ and $t_s$ and modifying paths accordingly
7. compute the height $h_i$ for each node $i$ in the tree $T$
8. initialize the residual capacity $R_j$ of the leaf node $s_i$ of the tree $T_1$ to $n_i$ where $s_i$ the source node of the path $p_i$
9. initialize the residual capacity $R_j$ of the leaf node $t_j$ of the tree $T_2$ to $n_i$ where $t_j$ the source node of the path $p_i$
10. compute the residual capacity $R_i$ of each intermediate node of the trees $T_t$ and $T_s$ as the sum of the residual capacities of its child nodes

A. Algorithm to solve the BETI Problem

The outline of the BETI Algorithm is given by Algorithm 4. The input to the algorithm is an instance of the BETI Problem and it outputs a set of wavebands $B = \{B_1, B_2, \ldots\}$.

Let us discuss the working of BETI Algorithm in detail. The algorithm constructs a destination-rooted and a source-rooted capacitated tree $T_t$ and $T_s$ respectively using the Initialization algorithm with input $P$ and $C$. Let $h_d$ and $h_s$ be the heights of the trees $T_d$ and $T_s$ respectively. Let $h$ be the maximum of the heights of both the trees. The Algorithm 4 iterates with decreasing height and computes wavebands in each iteration. In an iteration with height $i$, the algorithm selects a node $u$ of
Algorithm 4 The BETI Algorithm for computing the wavebands.

1: **Input:** \((G, P, C)\)
2: **Output:** Waveband set \(B\)
3: run Algorithm 3 on input \((G, P, C)\) to compute trees \(T_t\) and \(T_s\)
4: let \(h_t\) and \(h_s\) be the heights of the trees
5: let \(h\) be the maximum of the heights \(h_d\) and \(h_t\)
6: for \(i = h; i \leq 2; i--\) do
7:   for each \(u\) in \(T_t\) and \(T_s\) where \(h_u = i\) in the corresponding tree, and \(R_u \geq g\) do
8:     if \((S_u = 2(i+1)(g-1) - 4g > 0)\) then
9:         form waveband \(B = (Q, u, d, g)\) from node \(u\) to root node \(d/s\) corresponding to tree \(T_t/T_s\) and add to \(B\)
10:    update the residual capacities \(R_u\) of all the nodes along the paths in included in the waveband in both the trees \(T_t\) and \(T_s\)
11: end if
12: end for
13: end for

height \(i\) from either of the trees with residual capacity at least \(g\). Note that the same node may have different heights in both the trees. We form the waveband \(i\) if and only if \((S_u = 2(i+1)(g-1)-4g > 0)\) holds i.e., number of ports saved is non-negative. Now, if we select a node \(u\) in the tree \(T_d\) then we form a waveband \(B = (Q, u, d, g)\) from node \(u\) to node \(d\) of granularity \(g\). But if we select a node \(u\) in tree \(T_s\) then we form a waveband \(B = (Q, s, u, g)\) from node \(s\) to node \(u\) of granularity \(g\). We add the waveband \(b\) to the set of wavebands \(B\). We update residual capacities of all the nodes along each of the paths \(p \in Q\) in both the trees \(T_d\) and \(T_s\). The algorithm continues to form wavebands of length \(i\) until no node \(u\) exists with \(R_u \geq g\) and \((S_u = 2(i+1)(g-1)-4g > 0)\) after which it continues with the next iteration decrementing the height.

V. CONCLUSIONS

In this paper we studied the problem of intermediate waveband switching assuming that routing and wavelength assignment are already known. Three types of grooming policies exist when considering intermediate waveband switching, namely, destination-to-intermediate, source-to-intermediate and intermediate-to-intermediate grooming. We presented algorithms for uniform and non-uniform waveband switching where either destination-to-intermediate or source-to-intermediate grooming is allowed. We also presented algorithms for uniform and non-uniform waveband switching where both destination-to-intermediate and source-to-intermediate grooming (source-to-intermediate grooming) is allowed. The waveband switching where intermediate-to-intermediate grooming is allowed is a hard problem. In future we plan to study intermediate-to-intermediate waveband grooming and its performance in comparison to source-to-intermediate grooming.

REFERENCES