IP-Subnet Aware Routing in WDM Mesh Networks

Swarup Acharya *Bhawna Gupta* *Pankaj Risbood* *Anurag Srivastava*
Network Software Research Department, Optical Networking Division
Bell Laboratories, Lucent Technologies, Inc.
{acharya,bhawna,risbood,anurag}@research.bell-labs.com

Abstract— **We explore the problem of routing bandwidth guaranteed paths in wavelength-routed, WDM optical mesh networks. A WDM mesh network offers great flexibility in dynamically re-configuring the optical core to match the IP layer demands. In this paper, we argue that IP subnets can limit the re-configurability potential of the WDM mesh network. We show that finding the shortest IP-hop path, normally admitting a straightforward polynomial solution on the WDM mesh, is NP-hard in the presence of subnets. We propose a new algorithm called *MöbiTwist* that finds the optimal shortest path when accounting for subnets. We also observe that subnets impose a routing penalty by forcing longer paths for bandwidth demands. Consequently, they create a trade-off between lower network efficiency if subnets are honored (due to longer paths) or, an upfront overhead of dynamically changing subnets to derive shorter paths. We propose the *MöbiFlex* algorithm that attempts to achieve a balance by finding the shortest path given an upper limit on the number of subnet violations acceptable. The inherent hardness of the routing problem due to subnets precludes a solution with low *worst-case* complexity. However, we present performance results that show that both the algorithms proposed are extremely efficient in routing demands, and in practice, do so in polynomial time.**

I. Introduction

We investigate the problem of routing bandwidth guaranteed paths in WDM (Wavelength Division Multiplexing) mesh networks. In a WDM mesh network, also referred to as the IP-over-Optical model, IP routers connect directly to a switched optical core consisting of optical cross-connect switches (OXCs) interconnected via high-speed Dense Wave Division Multiplexing (DWDM) line systems. Unlike, the ring architecture of the SONET-based transport, a mesh is more appropriate for the OXC-based core. The switching capability of the OXCs allow for the creation of end-to-end lightpaths (or, λs) across the optical core which in turn creates a mesh virtual-topology at the IP layer. The routers then route the bandwidth demands along this topology. Consequently, the IP layer paths are multi-hop and thus, traverse multiple λs in the core.

One of the key challenges in any network environment is that of routing bandwidth-guaranteed paths since they form the basis for higher-level QoS dependent services. Their primary application in WDM mesh networks is for MPLS traffic engineering. The WDM mesh networks are attractive for service providers because they provide the appropriate hooks for control and management of MPLS traffic. The rapid λ switching capabilities of the OXCs allow the optical core to be re-configured on-demand. In parallel, protocols such as GMPLS, ASTN and O-UNI enable seamless interoperability within the optical domain and across the different domains on the service provider network. Consequently, integrated *cross-domain*[1] routing that incorporates traffic and topology information from both IP and optical domains in the path selection process, has started to receive serious attention for MPLS traffic engineering [1], [2], [3]. Also referred to as MPLS over WDM, the goal of cross-domain routing of MPLS traffic is to create a more efficient end-to-end network by “discovering” bandwidth that would be wasted if each domain were routed independently. In the next section, we provide a simple example of cross-domain routing to highlight this advantage.

However, one issue that is yet to be addressed in literature is the impact of IP subnets on cross-domain MPLS over WDM routing. Subnets are a mechanism to reduce routing overheads of IP networks by grouping a set of addresses under a single “subnet” identifier [4], [5]. In turn, the routing protocols route to subnets and not to individual hosts. For example, an interface with address 10.3.2.1/24 is on the subnet 10.3.2.0 (i.e., a 24-bit subnet mask). A key restriction of the IP subnet model is that two routers can send packets via their connected interfaces *if and only if the two interfaces are on the same subnet.* [2] Interior gateway routing protocols (IGP) such as OSPF(-TE) [4], [7] strictly follow this constraint when auto-discovering IGP neighbors. This restriction, which we shall refer to as *subnet constraint*, fundamentally impacts MPLS over WDM routing, crippling the very flexibility that WDM networks aim to provide.

In this paper, we highlight the impact of the subnet constraint on cross-domain routing, propose novel routing algorithms and argue that in the emerging network model, subnet management should shift from being a static configuration issue to take on a more real-time, dynamic role.

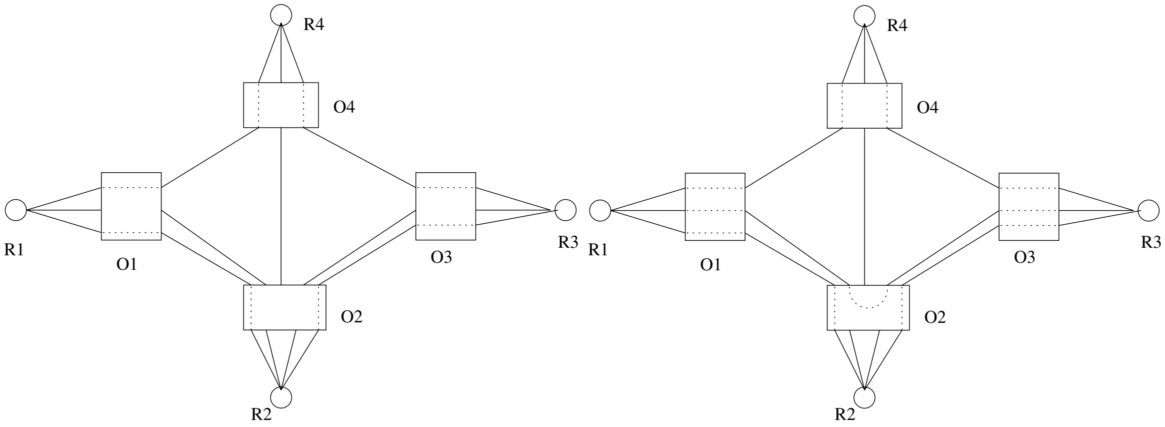
A. Contributions and Outline

To the best of our knowledge, this is the first work that highlights the impact of IP subnets on WDM mesh networks. We focus on the problem of finding bandwidth-guaranteed end-to-end paths in the WDM mesh model. We formally prove that finding the shortest IP hop path, admitting polynomial solutions on the WDM mesh network in the absence of subnets [2], [8], is *NP-hard in the presence of subnets*. We

[1] In this paper, we use the word *domain* interchangeably with layer to refer to technology domains (IP, SONET, ATM etc.) because of their one-to-one relationship with network layers.

[2] IPv6 does not enforce this constraint. [6]

0-7803-7753-2/03/$17.00 (C) 2003 IEEE



(a) Original Network (b) Topology after λ setup

Fig. 1. Example WDM Mesh Network $\mathcal{A}$

show that subnets impose a routing penalty by potentially forcing longer paths in the network in order to honor the subnet constraint. Consequently, we identify three classes of WDM mesh networks: networks planned with no subnets, networks with pre-defined subnets assigned to interfaces and finally, propose a novel network model wherein subnets may be *dynamically re-configured* during route setups in order to minimize the path length penalty. We present efficient shortest path algorithms for these different scenarios. The inherent hardness of the problem clearly precludes a low *worst-case* complexity. However, we show that in practice, those algorithms find the *optimal* shortest-path in polynomial time, comparable to Dijkstra's shortest path algorithm. We present this paper in the context of routing MPLS traffic. However, the contributions of this paper are relevant to any bandwidth guaranteed routing context.

The paper is organized as follows. In the next section we briefly summarize MPLS cross-domain routing and highlight the benefit of a switched optical core. In Section III, we highlight the problems imposed by subnets on routing bandwidth guaranteed paths. Section IV describes relevant work and in Section V, we outline the network model under consideration and provide the hardness proof. In Section VI, we present the *MöbiTwist* algorithm that finds the optimal shortest path in the presence of subnets and in Section VII we present the *MöbiFlex* algorithm that generalizes the problem to accept some limited violation of the subnet constraint. Section VIII provides performance numbers for the algorithms and finally, we conclude.

II. Cross-Domain Routing for MPLS Traffic Engineering

MPLS traffic engineering over WDM routed networks promises to be the next big step in the service provider network evolution. The ability to configure explicit routes for the Label Switched Paths (LSPs) and rapidly re-configure the virtual topology to mirror this traffic makes this combination highly effective. In the absence of a switched core, the virtual topology is static, limiting traffic engineering to the connectivity it provides. This is overly restrictive and makes incomplete use of network resources.

Consequently, integrated cross-domain routing was proposed that uses topology and traffic information from both IP and optical domains to determine routes [1], [2]. We define a cross-domain route as one that uses a combination of existing IP connectivity in the virtual topology and new optical λs (that further creates new IP connectivities in the virtual topology) for the route. In the example below, we highlight the advantage that a cross-domain route provides over routing independently on each domain. In fact, we submit that the real utility of the WDM mesh network is in the improved efficiency that cross-domain routing provides.

Consider Figure 1(a). It shows a network of four IP routers (`R1`-`R4`) connected to an optical core consisting of four OXCs (`O1`-`O4`). We denote this network as $\mathcal{A}$. The dotted lines inside the OXCs represent cross-connects and the solid lines interconnecting them is the WDM line system network. Each router is running an IGP with appropriate traffic engineering hooks such as OSPF-TE, and the virtual topology for each router is reflected by the IGP's view of the network connectivity.

Assume that IP links `R1`-`R4` and `R1`-`R2` are at full capacity. If a new MPLS LSP request arrives for some specific bandwidth from `R1` to `R3`, it would be denied by `R1` since no free path to `R3` exists in virtual-topology. In response, on a switched optical core, one might trigger an UNI to create additional bandwidth at the IP layer. For example, a UNI request from `R1` to `R3` would produce a 2-hop optical λ (`O1`-`O2`-`O3`) making `R1` and `R3` IP neighbors via their free interfaces. This is shown in Figure 1(b). The MPLS request can now be routed over this new `R1`-`R3` connection on the IP virtual-topology.

It may not be always possible to UNI a one-hop IP path through the optical core. Consider a slightly modified network $\mathcal{B}$ that is similar to $\mathcal{A}$ but with no free interface on router `R3`. In this case, even the UNI approach will fail.

However, there is enough capacity in $\mathcal{B}$ if one takes an integrated *cross-domain view* of the network. Figure 2 shows

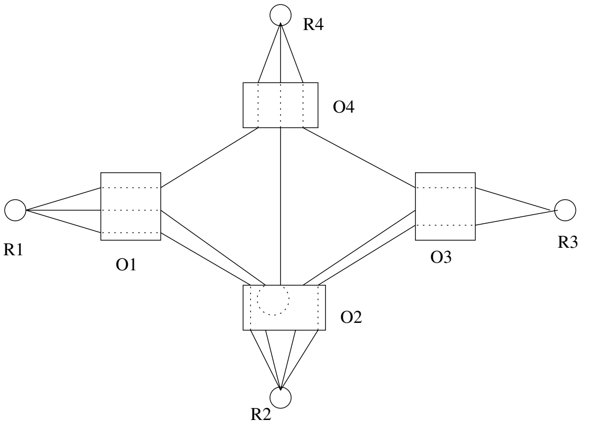


Fig. 2. Network $\mathcal{B}$ topology after cross-domain route

the cross-domain path for the demand on $\mathcal{B}$. The cross-domain route requires provisioning a λ from R1 to R4 (instead of R3) and using a pre-existing virtual-topology path from R4 to R3. The MPLS tunnel is then provisioned via R1-R4-R3.

Thus, cross-domain routing creates a more efficient end-to-end network by "discovering" available network capacity that would be wasted if each domain was routed in isolation. Clearly, it is the flexibility of the switched core that enables this novel form of routing. Moreover, the various signaling and interoperability protocols ensure its applicability in practice.

III. Impact of Subnets on WDM mesh networks

Subnet and interface address assignment are a key component of IP network design and require thoughtful planning. It has typically been considered as a one-time configuration issue since the topology was never expected to be re-configurable. However, the switched nature of the optical core can potentially create new IP connectivities, or, re-configure the existing virtual-topology as optical λs are setup and torndown. Thus, the subnet constraint that requires both ends of an IP link to be on the same subnet *can fundamentally limit the re-configurability potential of a WDM mesh.* For example, in Figure 1(b), UNI-ing a λ from R1 to R3 is of value only if the free interfaces on both the routers are on the same subnet. Otherwise, even though the two routers may have link layer connectivity via the λ, OSPF will fail to recognize the two routers as IGP neighbors.

A. Impact of Subnets on MPLS over WDM Routing

Recall the examples in Figures 1(a), (b) and Figure 2 where the shortest path was found to satisfy a bandwidth demand from R1 to R3. In those cases, there was no restriction placed on the router connectivities; allowing any two interfaces to be connected via the optical core.

Now assume that each IP interface in network $\mathcal{A}$ is assigned an address and thus, is on a specific subnet. In OSPF terminology, these would be numbered interfaces [4]. Figure 3(a) shows the network with each interface colored based on their subnets.[3] Thus, two interfaces can be connected by an optical λ only if they have the same color. Consider the same problem of creating a MPLS tunnel from R1 to R3. The UNI solution of Figure 1(b) is ineffective since the two free interfaces on R1 and R3 are on the different subnets (red and magenta respectively).

In order to find the shortest-hop path honoring the subnet constraint on the virtual-topology, one must find a sequence of virtual links that add up to create an end-to-end path such that the following two constraints are met on all the the pair-wise neighboring interfaces along the path: a) both interfaces are on the same subnet and b) there are free optical resources to signal a λ between the two, i.e., they are optically reachable. For the example network, the shortest path requires two λs to be dialed as shown in Figure 3(b) — one from R1 to R2 on the red interfaces and a second from R2 to R4 on the blue interfaces, creating two new connections on the virtual-topology. The MPLS tunnel is then routed in three hops – on the new IP links R1-R2 and R2-R4 and the existing R4-R3 virtual link.

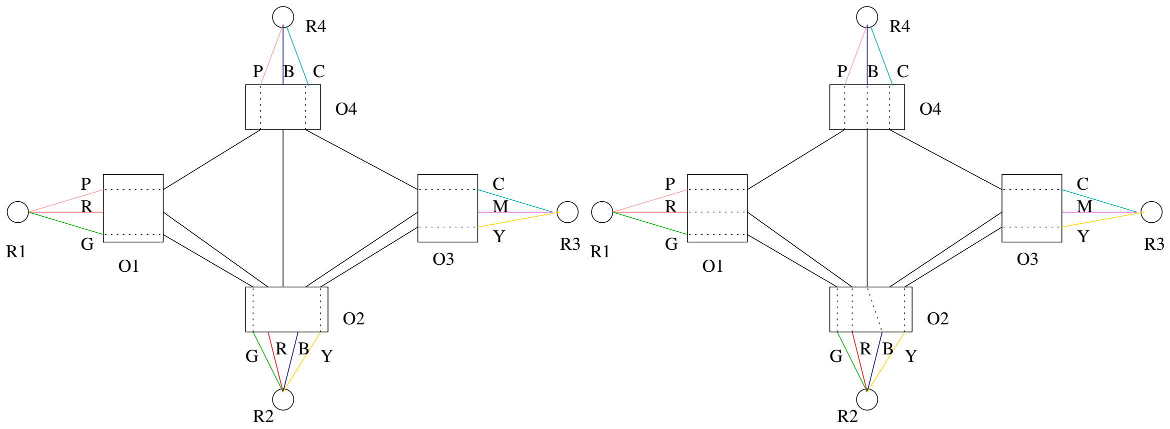
B. Making Subnets Re-configurable

The examples thus far highlight an interesting dilemma for cross-domain routing, namely, that enforcing the subnet constraint forces longer network routes. For example, if subnets are ignored, a one IP-hop path exists between routers R1 and R3 in network $\mathcal{A}$. However, this path works in practice only if one or both the interfaces are reconfigured to be on the same subnet. On the other hand, a three hop path is the shortest possible when accounting for subnets. It requires two new λs to be provisioned but no reconfiguration of the interfaces. Note that it may not always be possible to find a one-hop path on the virtual-topology for every source-destination pair, particularly as the network size increases. For example, in Figure 2, the shortest path ignoring subnets was 2-hop. In such cases, honoring the subnet constraint creates even longer shortest paths.

Clearly, there is a routing trade-off between honoring subnet constraints and the length of the shortest path possible. A longer path is wasteful of network resources (router interfaces, OXC ports etc.) and makes the network inefficient over time. On the other hand, changing subnets (and interface addresses) involves an upfront reconfiguration overhead while creating a more efficient network. Given this trade-off, how can the two conflicts be balanced? More specifically, assuming the network operator is willing to accept some network reconfiguration, the issue is in balancing the costs: how much one-time re-configuration overhead is acceptable to save a routing hop? Numerically answering this trade-off is beyond the scope of this paper and would likely vary by network and the specific demand being satisfied.

From an algorithmic viewpoint, however, one can formulate the problem as follows — given an end-to-end demand for

[3] This paper is best viewed in color. However, to make it readable in gray scale, each interface has been marked using the first letter of its color.



(a) Network $\mathcal{A}$ with Subnets

(b) Topology after λ setup

Fig. 3. Cross-domain routing with subnets

some specific bandwidth between two routers, if t is the length of the shortest path honoring the subnet constraints and s is the length of the shortest-path ignoring subnets and thus, requiring c subnet modifications to route, is it possible to find a path P of some specific length p, $s \leq p \leq t$ and requiring at most $d, d \leq c$ subnet modifications? In other words, assuming the operator has an upper bound on the upfront overhead (e.g., at most d subnet changes), what is the shortest path one can find on the virtual-topology?

In Section VII-B, we present the *MöbiFlex* algorithm that aims to solve this problem. We also briefly highlight some of the challenges in dynamically changing interface addresses and subnets on a live network and how one can minimize this overhead. Suffice it to say that operators do re-configure networks on the field. However, it is a costly process and thus, often driven by specific events such as addition of new routers, network failures etc. or, as part of periodic network engineering. An outcome of the inherent flexibility of the WDM mesh optical core is that it has the potential to make this re-configuration process much more frequent, particularly if the goal were to create a very efficient, highly utilized network.

IV. Related Work

Routing problems have been studied since the fifties and numerous variations such as shortest path, constrained shortest path and multi-constraint path have been proposed in the IP context [9]. Recently, there has been a renewed focus on optimizing the virtual-topology to best meet the IP traffic needs [10], [11]. These papers are complementary in the sense that they are more focussed on optimizing the IP and optical topologies while our focus is to route a single demand efficiently.

To the best of our knowledge, this is the first paper to address the problem of IP-optical integrated routing in the context of IP subnets. This is largely attributable to the fact that, once deployed, the network topology was expected to be fairly static. However, as motivated earlier, with the deployment of switched optical elements, this assumption is becoming increasingly obsolete.

The problem of integrated cross-domain IP-optical routing, was first introduced in [2]. The paper motivated the need for integrated routing and proposed an efficient routing algorithm. Our work is closest to that paper in terms of the high-level goal, but with some fundamental differences. That paper aimed to provide minimum interference routing in the presence of OXC nodes that allow wavelength conversion and those that do not. However, the algorithm did not account for subnets and thus, its applicability is limited in practice. In this paper, our goal is to route accounting for subnet constraints. As we show below, even the simplest case of shortest path routing in the presence of subnets is computationally hard.

The other related work is in the area of constrained shortest path routing. The original work on application of Lagrangian relaxation technique to this problem was by Jaffe [12]. The generalized algorithm *MöbiFlex* is based on work on multi-constraint routing proposed in [13].

V. Problem Definition and Complexity

A. Network Model

In this section, we describe the network model. A WDM mesh network consists of IP routers surrounding a OXC-based switched optical core. We assume that the OXCs are wavelength conversion capable [14], i.e., any input port can be cross-connected to any output port. While orthogonal to the routing algorithm, we recognize the need for some signaling mechanism to automate the path provisioning. On the IP layer, this requires the availability of MPLS for switching and RSVP or CR-LDP for resource reservation [15]. At the optical layer, λs may be signaled via a centralized NMS or, a GMPLS/ASTN control plane. In order for complete automation, a router may require an O-UNI style signaling mechanism to the optical core [16]. We assume all bandwidth guaranteed path requests are MPLS LSP requests. Additionally, we expect routers to be running OSPF-TE though this work is applicable to any other IGP with appropriate traffic engineering extensions. Consequently, we assume that the routing engine has up-to-date knowledge about the available link bandwidth extracted from the IGP. OSPF allows IP router interfaces to be marked in one of two ways. They can be *numbered*, i.e., are assigned a physical IP address, or, be *unnumbered* [4]. Not being assigned an IP address, an unnumbered interface is not bound by the subnet constraint. We assume we only know the demand being requested with no knowledge of the future.

B. Problem Statement

We formally define below the three routing problems we explore in this paper.

1) Integrated Shortest Path ($\mathcal{ISP}$): Given a WDM mesh network as outlined in section V-A, a source-destination router pair and a bandwidth to be routed, find the shortest IP-hop path that satisfies the bandwidth requirement without considering the subnet constraints. The solution found will be acceptable if either all interfaces are marked as unnumbered or one is willing to change subnet numbers at as many interfaces as required.

A polynomial time algorithm for this problem was proposed in [2] and the interested reader is referred there. We simply present the problem for completeness and will not explore it further. Henceforth, we assume that we know how to determine the shortest-hop path if the subnet constraint is ignored.

2) Subnet Shortest Path ($\mathcal{SSP}$): Given a WDM mesh network as outlined in Section V-A, a source-destination router pair and a bandwidth to be routed, find a *subnet feasible* shortest IP-hop path that satisfies the bandwidth requirement. A $\mathcal{SSP}$ path is subnet feasible if every pair of neighboring routers along the path satisfies the subnet constraint.

The *MöbiTwist* algorithm is presented in Section VI-B as a solution to the $\mathcal{SSP}$ problem.

3) Generalized Subnet Shortest Path ($\mathcal{GSP}$): Given the preconditions in the $\mathcal{SSP}$ problem and a number *K*, find a feasible shortest IP-hop path that satisfies the bandwidth requirement. A $\mathcal{GSP}$ path is feasible if the number of subnet constraint violations along the path is less than *K*. A violation is defined as a link in the virtual-topology, whose end interfaces are not on the same subnet. A dual of this problem is, given the maximum acceptable path length, find the path with minimum number of subnet violations.

The *MöbiFlex* algorithm is presented in Section VII-B as a solution to the $\mathcal{GSP}$ problem.

Note that $\mathcal{ISP}$ and $\mathcal{SSP}$ are special cases of $\mathcal{GSP}$. In particular, *K* is infinite for $\mathcal{ISP}$ and zero for $\mathcal{SSP}$.

C. Routing Hardness

We analyze the complexity of the three routing problems. We prove that the $\mathcal{SSP}$ problem is NP-hard. Since $\mathcal{GSP}$ is a generalized version of $\mathcal{SSP}$ (for *K*=0), it follows that $\mathcal{GSP}$ is also NP-hard. $\mathcal{ISP}$ has been shown to admit a polynomial solution in [2].

We prove the hardness of $\mathcal{SSP}$ by reducing the well known NP-complete problem, Constrained Shortest Path ($\mathcal{CSP}$) problem [17] to it. Broadly, the $\mathcal{CSP}$ problem requires determining the shortest path while additionally minimizing one or more constraints (e.g., delay). We formally define the $\mathcal{CSP}$ problem as follows. Consider an arbitrary directed graph $G(V,A)$, where V is the set of nodes and A is the set of links with two cost metrics associated with them, i.e., each link $l_i(v_j,v_k)$ has a cost tuple (d_{i1},d_{i2}). Given a source and a destination node (s,t), find a path P from s to t such that:

$\sum_i d_{i2} < C, \quad \forall\ l_i \in P$, for some C and
$\sum_i d_{i1}$ is minimum.

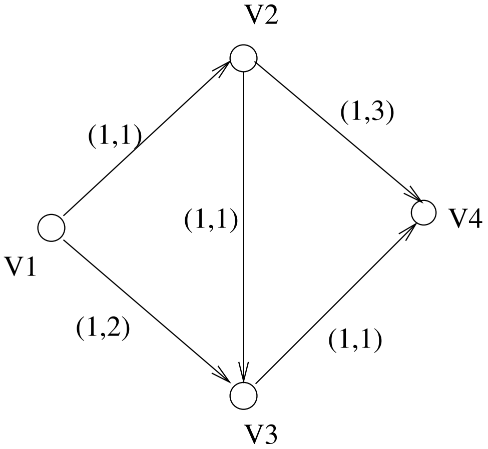


Fig. 4. Example $\mathcal{CSP}$ graph

We will now map the $\mathcal{CSP}$ to the $\mathcal{SSP}$ problem. Assume that d_{i2} is a natural number for all $l_i \in A$. Figure 4 shows an example graph with the cost pair alongside each edge. We will use this graph as a running example to demonstrate the mapping construction. The construction is in two steps. In the first step, we convert Figure 4 to an intermediate graph and in the second step, we transform this intermediate graph into a WDM mesh network with subnets.

D. Transformation I

Let $n = \sum_i d_{i2}$. Create two sets of nodes $V_a = \{v_{11}, ..., v_{1n}\}$ and $V_b = \{v_{21}, ..., v_{2n}\}$ such that $|V_a| = |V_b| = n$. Connect a link each between v_{1i} and v_{2i} for all nodes in V_a and V_b. Call the link set L.

Now, for each link $l_i(v_j,v_k)$ in the original network with cost (d_{i1},d_{i2}), create $(d_{i2}-1)$ nodes subject to $d_{i2} \geq 1$. These are shown as filled black nodes in Figure 5. Connect these nodes to $(d_{i2}-1)$ nodes of V_a and $(d_{i2}-1)$ nodes of V_b as shown in the figure. This construction uses up d_{i2} links of set L and creates a multi-hop path of length $3d_{i2}-2$. Let the end points of this path (shown in bold in figure 5) now created be $v_{1a} \in V_a$ and $v_{2b} \in V_b$. Add link between v_j, v_{1a} and v_k, v_{2b}, each with cost $d_{i1}/2$. All other links have a cost of zero. This path from v_j to v_k is now used to replace the link $l_i(v_j,v_k)$. Note that cost of going from v_j to v_k via this path is d_{i1} and it uses d_{i2} links of set L. Not also that $\sum_i d_{i2}$ links in set L are sufficient for the whole construction.

Our construction has resulted in a network with $2|A| + 2\sum_i(d_{i2}-1) + \sum_i d_{i2}$ links and $|V| + \sum_i(d_{i2}-1) + 2\sum_i d_{i2}$ nodes. If d_{i2} is bounded i.e., $d_{i2} < d\ \forall\ l_i(v_1,v_2) \in A$, then $\sum_i d_{i2}$ is $O(A)$ and thus construction is polynomial.

Consider the problem of finding least cost path from some node s to t in this network that uses no more than C links of

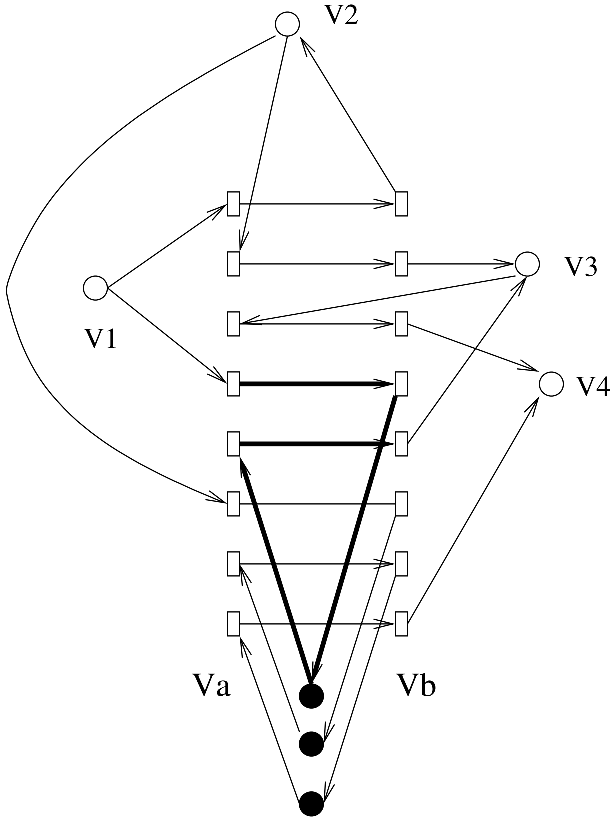


Fig. 5. Figure (4) after Transformation 1

set L. If such a path is found, then the sequence of original nodes in this path is the solution to two constraint shortest path in the original network with $\sum_i d_{i2} < C$. Thus, this problem is NP-hard.

E. Transformation II

We now transform the network generated by Transformation I into a WDM mesh network of routers and OXCs. For each link in L in Figure 5 mark two adjacent links (there will be no more than two) by same color. Replace all nodes of set V_a by a single node V_{o1} and similarly for V_b by V_{o2}. Add C links between two nodes. Now the problem can be posed as finding a shortest path (i.e., possibly with loops), from s to t with the constraint that the ingress color on V_{o1} should be the same as the egress color on V_{o2}. Figure 6 shows the network after this construction for the network in Figure 5. If such a path is found, the sequence of nodes along the path will solve the original $\mathcal{CSP}$ problem.

This construction results in a special case of a WDM mesh network of routers (v_i nodes and the filled black nodes) connected to a network of OXCs. In this particular case, there are just two OXCs. Thus, the original $\mathcal{SSP}$ problem is NP-hard. Note that same argument holds if links are bidirectional.

VI. *MöbiTwist* ALGORITHM FOR $\mathcal{SSP}$

In this section, we describe *MöbiTwist*, a cross-domain routing algorithm that accounts for IP subnets and solves

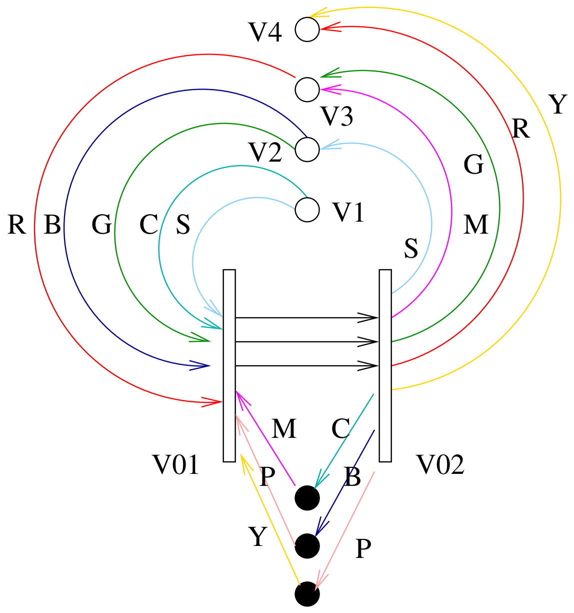


Fig. 6. Figure 5 after Transformation 2

the $\mathcal{SSP}$ problem. *MöbiTwist* works by first converting the WDM mesh network into a network graph on which it runs a path selection algorithm. We first describe the transformation process and then, present the algorithm.

A. Network Transformation

We present the graph model on which the routing algorithm operates. We shall refer to the WDM mesh network as the Original Network ($\mathcal{ON}$) and the graph model as the Transformed Network ($\mathcal{TN}$). Let S be the total number of subnets in the network.

The mapping of routers and OXCs in $\mathcal{ON}$ to nodes in $\mathcal{TN}$ is done as follows. For every router R_i in the $\mathcal{ON}$, we create a corresponding node $R_i^{'}$ in $\mathcal{TN}$. For each OXC O_a in $\mathcal{ON}$, we create a set of S nodes $O_{a1}^{'}, ..., O_{as}^{'}$, each representing a logical OXC for each subnet.

The procedure to map links from $\mathcal{ON}$ to $\mathcal{TN}$ is as follows. All links in $\mathcal{ON}$ are bidirectional and are mapped as two unidirectional links. There are two types of links in $\mathcal{ON}$ – *external* links, that connect an IP interface to an OXC port and *internal* links that connect ports in neighboring OXCs. Consider an external link connecting an interface on subnet t on router R_i to a port in OXC O_a. If the interface is not part of the live network (i.e., not in the virtual-topology), then we connect $R_i^{'}$ to $O_{at}^{'}$ in $\mathcal{TN}$. If the interface features in the virtual-topology, i.e., it is interconnected via a λ to another router, say R_j, then, it does so through a sequence of internal links. We map the two external links and the internal links that form the connection in $\mathcal{ON}$ to a single link in $\mathcal{TN}$ that directly connects $R_i^{'}$ to $R_j^{'}$. We call this a *virtual short-circuit link* since such a link also exists in the virtual-topology. A similar construct was used in [2] to represent such logical links. Any internal links left over after accounting for all virtual links are then free (i.e., they are not cross-connected). Consider OXCs

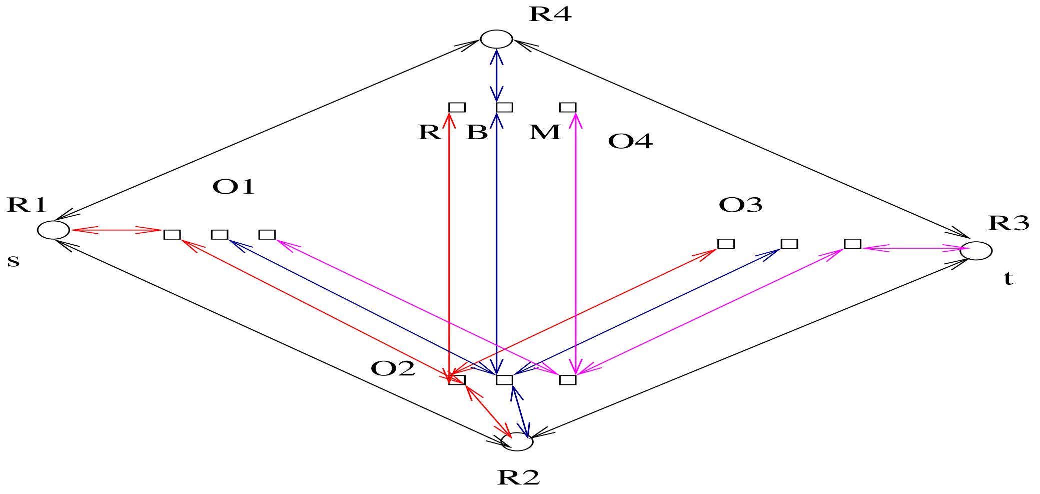


Fig. 7. Network in Figure 3(a) after Network Transformation

O_a and O_b. If there exists at least one free link between them (i.e., O_a and O_b are not disconnected in the residual optical network), we create S links in $\mathcal{TN}$, each connecting $O^{'}_{ai}$ and $O^{'}_{bi}$, $i \in \{1..S\}$.

Figure 7 shows the $\mathcal{TN}$ for the network in Figure 3 (a). Note that this transformation is essentially the reverse of Transformation 2 in Section V-E. In effect, we are exploding each OXC node into S smaller nodes, one for each subnet. Thus, if there are less than S links between a pair of OXCs in the real network, a path may be found in $\mathcal{TN}$ that may in fact be infeasible on the real network. This is an outcome of the inherent hardness of the problem. We will call such a path *optically infeasible* and will account for it in our routing algorithm. Conversely, note that even if there are greater than S links between two OXCs in $\mathcal{ON}$, they are reduced to S links between the logical smaller OXCs representing the subnets. We prove below that shrinking the size of the network in this fashion has no impact on the routing.

We now assign costs to each link. Recall that our goal in this work is to find the shortest IP hop path. Thus, we assign unit cost when we cross an IP router. Thus all the free external links between a router and a OXC are assigned a cost of 1/2 and all virtual short-circuit links are assigned a cost of 1. The free links in optical network are assigned a small cost such that the algorithm would prefer a logical link over creating any new IP connectivity and that two paths of equal hops are distinguished by their optical resource utilization.

We note that the algorithm is independent of the cost assignment and depending on the requirement, the costs may be assigned differently (for example, to give preference to optical resources). We have chosen this scheme to realize our goal of optimizing IP network performance.

B. *MöbiTwist Algorithm*

The *MöbiTwist* algorithm is shown above. It operates on the $\mathcal{TN}$ to find the shortest hop path accounting for the subnet constraints. However, the path found may be optically infeasible if it requires more wavelengths between a pair of OXCs than actually available on the real network. Recall that, the $\mathcal{TN}$ construction scales the capacity of each OXC to S, the number of subnets and thus, can create infeasibility if there are less than S free links between two OXCs. *MöbiTwist* cycles through various paths, starting from the shortest on, testing for infeasibility. If after some user defined parameter N for number of tries, it fails to find a path, it terminates without finding a path. Of course, one can continue this process until an optically feasible path is found. N is simply an upper bound to reduce the run-time complexity since in the worst case there exist exponential number of paths between any two nodes in a graph. However, as we show in Section VIII, in practice the algorithm terminates in very few iterations (often, one) making a very cheap solution to implement in practice.

MöbiTwist Algorithm Pseudocode

```
1) Convert original network of IP routers and OXCs to a
   transformed network as outlined in section VI-A.
2) Find $N$ shortest paths in transformed network between
   source and destination using Lawler's algorithm [18].
3) $I \leftarrow 1$.
4) Let $P$ be the $I$th shortest path in the transformed
   network from source to destination
5) If $P$'s analogous path in original network is feasible,
   then it is the optimal feasible path that satisfies subnet
   constraints. Terminate.
6) Else $I \leftarrow I + 1$;
7) If $I > K$, exit; else Goto Step 4.
```

C. *Optimality*

To prove that the algorithm indeed finds the shortest IP hop path, it suffices to prove that the K^{th} $(K \leq N)$ shortest path in $\mathcal{TN}$, which is optically feasible, is the shortest feasible path in $\mathcal{ON}$ if all previous $K-1$ paths are infeasible. Before we prove the correctness, we first make two observations about our construction.

Theorem 1: The shortest subnet feasible path between any pair of routers in $\mathcal{ON}$ will traverse through any OXC at most S times.

Proof: We prove by contradiction. Assume a shortest path travels through an OXCs more than S times. Each traversal of an OXC is always associated with a single subnet since subnets can change only at the routers. Thus, for a path to traverse an OXC more than S times, at least one subnet associated with the traversal should be repeated. We can represent such a path as

$P = P_{si} <> X(a,b) <> P_{ii} <> X(c,d) <> P_{it}$,

where $<>$ is the concatenation operator, P_{si} is the sub-path from source s to intermediate OXC i and $X(a,b)$ is a cross-connect setup at OXC between port a and b. By the above argument, the subnet traversed on $X(a,b)$ is same as subnet traversed on $X(c,d)$. Since port a and d are on the same subnet we can replace this path by

$P^{'} = P_{si} <> X(a,d) <> P_{it}$, and,

$|P^{'}| < |P|$

$P^{'}$ is a shorter path between the same source and destination and it also satisfies subnet constraint. Thus, we can always replace a path traversing an OXC more than S times by a shorter path. Hence shortest subnet feasible path traverses an OXC at most S times. ■

Corollary 1: The shortest subnet feasible path between a pair of routers will never require more than S wavelengths between a pair of OXCs.

Theorem 2: For every shortest subnet feasible path P_{on} between a source-destination router pair in $\mathcal{ON}$ there exist an equivalent path P_{tn} in $\mathcal{TN}$ such that $cost(P_{on}) = cost(P_{tn})$.
Proof: Consider the path *P* in the transformed network that is formed by choosing links, which are transformed equivalents of links in P_{on}. For IP virtual short-circuit links and external links, there is a clear one-to-one mapping between the links and thus, their costs are the same (by cost assignment). For internal links between OXCs, choose the links corresponding to the subnet. By construction and cost assignment of these links in $\mathcal{TN}$, the path is well defined and $Cost(P_{on}) = Cost(P)$.

Note that by Corollary 1, P_{on} will not use more than S wavelengths between any pair of OXCs, thus scaling the set of links between a pair of OXCs to S links is sufficient even when there are more than S wavelengths between the pair of nodes in $\mathcal{ON}$. ■

Theorem 3: The path in $\mathcal{ON}$ corresponding to the Kth shortest path in $\mathcal{TN}$ is the shortest path in $\mathcal{ON}$, if all previous $K-1$ paths are infeasible.
Proof: We prove by contradiction. Let P^k_{on} be the path in $\mathcal{ON}$ corresponding to P^k_{tn}, the Kth shortest path in $\mathcal{TN}$. Let's assume by contradiction that P^k_{on} is different from P_{on}, the shortest path in $\mathcal{ON}$. Clearly, $Cost(P_{on}) < Cost(P^k_{on})$. Now, consider the path corresponding to P_{on} in $\mathcal{TN}$, say P_{tn}. By theorem 2, P_{tn} exists and

$Cost(P_{tn}) = Cost(P_{on})$ and hence

$Cost(P_{tn}) < Cost(P^k_{tn})$

Thus, P_{tn} should have been one of previous $K-1$ paths found by the algorithm. But since none of the $K-1$ paths were feasible, no such P_{on} can exist which is different from P^k_{on}. Hence, the *MöbiTwist* algorithm finds the shortest subnet feasible path in the network. QED. ■

D. Complexity

It is important to note that if number of free wavelengths between every pair of connected OXCs, is at least equal to number of subnets, then, the first path found will be the shortest feasible path. In the worst case, the complexity of algorithm is exponential as there can be exponential number of paths in a network. However, as we show in Section VIII, in practice, the algorithm typically terminates in polynomial time. If the Kth simple path turns out to be the optimal subnet feasible shortest path, then the complexity is as below:

Assume that there are R routers and N OXCs in $\mathcal{ON}$, with A_1 unused internal link and A_2 λs established resulting in A_2 virtual short circuit links and A_3 links connecting IP routers and OXCs. If there are total of S subnets then the $\mathcal{TN}$ has $(R + N * S)$ nodes and $(S * A_1 + A_2 + A_3)$ links. The cost of finding optimal shortest path is thus:

C = cost of finding K shortest paths +
cost of checking for infeasibility $*$ K

$C = O(K * (R + N * S)^3) + O(K * (R + N * S - 1))$

$C = O(K(R + N * S)^3)$.

This complexity is assuming use of Lawler's algorithm [18]. If one optimistically assumes the shortest path in $\mathcal{TN}$ will also be the optimal shortest path, then one can run the algorithm in two steps — run Dijkstra shortest path first and only on failure, run the Lawler's algorithm. If so, the best case complexity will be as in Dijkstra [8], namely, $O((S * A_1 + A_2 + A_3)log((R + N * S))$. As the performance numbers show in section VIII, this lower value turns out to be the complexity in practice.

VII. *MöbiFlex* ALGORITHM FOR $\mathcal{GSP}$

The $\mathcal{GSP}$ problem relaxes the constraints of $\mathcal{SSP}$ by allowing upto K subnet violations on the path. Recall that, as motivated in Section III-B, overhead to re-configure network interfaces and their subnets may be acceptable, if it improves the routing in return. Thus, a solution for $\mathcal{GSP}$ lies midway between solutions for $\mathcal{SSP}$ and $\mathcal{ISP}$.

We present the *MöbiFlex* algorithm next that finds the shortest path with at most K subnet violations. As in *MöbiTwist* case, we create a transformed network on which the algorithm operates.

A. Network Transformation

The model on which *MöbiFlex* operates on is very similar to that presented in the *MöbiTwist* case except for free links between an IP router and an OXC. We replace all of the free links between an IP router and an OXC by S links, each connecting the router to one of the S virtual OXC nodes created (one for each of the S subnets). This construction

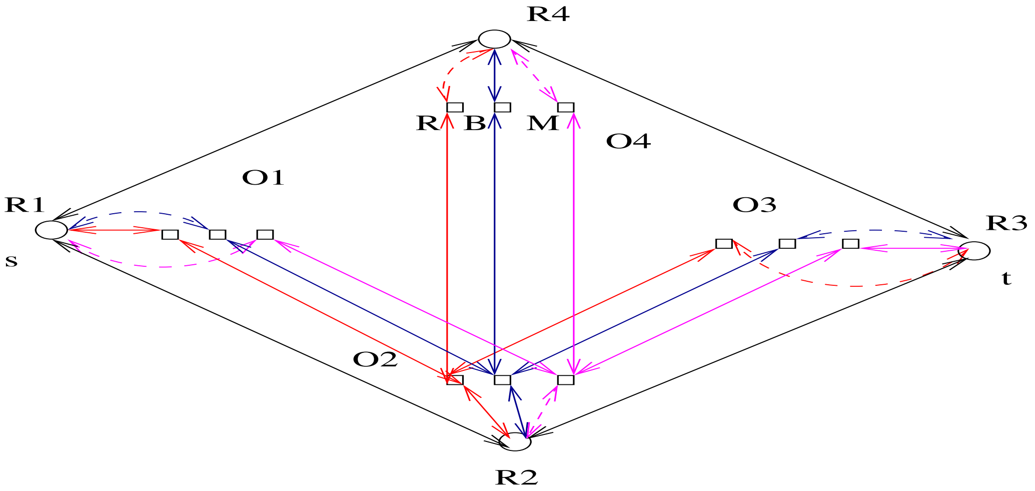


Fig. 8. *MöbiFlex* Network Transformation

increases the number of links emanating from the router on the $\mathcal{TN}$ compared to the *MöbiTwist* case. We associate two costs with them. The first cost is the same as in the *MöbiTwist* case. The second cost of a link is either 1 or 0. All the links that connect to virtual OXCs whose subnets are present in the router have a value 0, else it is 1. This construction models the cost of subnet violations by forcing a higher cost on links that cause it. Figure 8 shows the network after applying the transformation. Note that all dotted links between the router nodes and virtual OXC nodes are the ones which are newly added to model and have a non zero second cost (cost of modifying subnet) associated with them. It is important to note that the scaling of number of free external links does not introduce any new infeasibility since a router is never traversed twice on a shortest path. Thus, as long as there is one external link in the $\mathcal{ON}$, any shortest route in $\mathcal{TN}$ will be feasible. Note that the optical infeasibility still remains.

Thus, the $\mathcal{GSP}$ problem is now equivalent to finding a shortest path based on first link cost, such that the sum of second link cost is less than K. It then reduces to a Constraint Shortest Path problem on the transformed network.

B. MöbiFlex Algorithm

As motivated above, the *MöbiFlex* algorithm attempts to find a solution midway between a solution to the $\mathcal{ISP}$ and $\mathcal{SSP}$ problem. It starts with solutions provided by the shortest path algorithm which ignores subnets ($\mathcal{ISP}$) and the *MöbiTwist* algorithm for $\mathcal{SSP}$ as the two extremes. It then attempts to narrow down the range of available options using a Lagrangian relaxation technique [12], [13]. Note that since there are two metrics being optimized (path length and number of subnet violations), the relaxation requires mapping the two variable into one. For *MöbiFlex*, we choose a linear transform function. The interested reader is referred to research on Lagrangian relaxation on the general technique and other alternative polynomial transformation functions one may use [12], [13].

MöbiFlex Algorithm Pseudocode

```
1)  Find the best solution to the $\mathcal{ISP}$ problem. This is the shortest path on the integrated network ignoring subnets. Let the path found be $P$.
2)  If $P$ is not found, terminate. There exists no path from source to destination on this network.
3)  If $d_2(P) < K$, terminate. $P$ is the required solution.
4)  Find the $\mathcal{SP}(d_{i1} + \infty d_{i2})$ and let the path found be $Q$. This is the path corresponding to no subnet violations.
5)  If no path is found (i.e., no solution honoring subnet constraints), find $\mathcal{SP}(d_{i1} + \theta d_{i2})$ for very large value of $\theta$.
6)  If $d_2(Q) > K$ then there is no solution terminate.
7)  $\gamma = \frac{d_1(Q)-d_1(P)}{d_2(P)-d_2(Q)}$.
8)  Find $\mathcal{SP}(d_{i1} + \gamma d_{i2})$ and let the path found be $R$.
9)  If $d_2(R) \leq K$ then $Q \leftarrow R$.
10) Else $P \leftarrow R$.
11) Check new $Q$ for optical feasibility and if feasible, record the route as current best known solution.
12) If terminate() then terminate.
13) Else Goto Step 7.
```

For the following discussion, we follow certain shorthand notation for convenience. For any path P in the network obtained after applying the transformation of section VII-A, we define $d_1(P) = \sum_i d_{i1} \ \forall \ l_i \in P$ and $d_2(P) = \sum_i d_{i2} \ \forall \ l_i \in P$. We also assume a routing engine that given a transformed network and the cost of various links, will compute the shortest path. Calls to this engine are denoted by $\mathcal{SP}(func(d_{i1}, d_{i2}))$, where $func(d_{i1}, d_{i2})$ is the cost function which represents the cost of link l_i.

The *MöbiFlex* algorithm is given above. It starts with solutions at two extremes from the $\mathcal{SSP}$ and $\mathcal{ISP}$ problem (algorithms *MöbiTwist* and [2] respectively) and continually narrows the range until the desired conditions are met. The correctness of the *MöbiFlex* algorithm is based on the following two results from [13]. Given a network:

- Let P be the shortest cost path for each link cost l_i set

to $d_{i1} + \alpha d_{i2}$ and Q the shortest cost path for each link cost l_i set to $d_{i1} + \beta d_{i2}$. If $\alpha > \beta$ then
$d_1(P) \geq d_1(Q)$ and $d_2(P) \leq d_2(Q)$.

- Let $\gamma = \frac{d_1(Q)-d_1(P)}{d_2(P)-d_2(Q)}$. Let R be the shortest cost path for each link cost l_i set to $d_1 + \gamma d_2$ then
$d_1(P) \geq d_1(R) \geq d_1(Q)$ and
$d_2(P) \leq d_2(R) \leq d_2(Q)$.

Every iteration of the algorithm reduces the operating range and if the current path is optically feasible, *MöbiFlex* makes it the best path seen so far. When the scope for further improvement is minimal, the algorithm terminates and outputs the best known solution. The termination condition is captured by the `terminate()` subroutine and we leave it upto the user to specify what the conditions may be. There could be various factors that influence when to terminate the search. For example, $d_2(Q) = K$ since no valid solution can be found by further search. Another criterion could be if the value $(d_1(P)-d_1(Q))$ becomes less than some acceptable threshold.

It is instructive to note that the dual problem, where hop count is bounded and intent is to minimize the number of subnet violations can be solved using same procedure by interchanging the two costs.

C. Practical Implications of Subnet Changes

While orthogonal to the paper, we briefly discuss the impact of subnet violations. Once the algorithm returns a path, the links with subnet violations have to be re-configured. This may require one interface to switch its subnet to the other, or, both to move to a new subnet. Note that there is an additional constraint imposed by routers which disallows more than one interface to be on the same subnet. The details of how one may "sanitize" the network to ensure the validity of the route is beyond the scope of this paper. However a simple technique would be to use an unused pool of /30 subnets. Changing interface addresses may additionally lead to OSPF routing table updates. However, this overhead can be reduced by appropriate planning. For example, an interface that is not "live" (a requirement to be able to UNI to it) can be engineered to not generate OSPF link state updates and thus, changing its address has minimal overhead.

VIII. Performance Figures

In this section, we show the performance of the *MöbiTwist* algorithm. We do not present any numbers for *MöbiFlex* since it uses the same underlying shortest path technique as *MöbiTwist* and its performance follows similarly. The complexity of *MöbiTwist* clearly depends on N, i.e., how often the sequence of paths found by the algorithm turn out to be optically infeasible. We present results of extensive tests over a large number of randomly generated WDM mesh networks, varying the number of subnets in the network and number of interfaces available on IP routers.

The Figures 9, 10 and 11 present the results of one such test. In this case, we used a network of 15 IP routers and 15 OXCs. The number of subnets were varied from 5 to 20 keeping the maximum number of interfaces per router limited to 4 with an average of 2 interfaces per router. Requests among routers were generated for all source destination pairs, and the test involved computing 10 shortest paths that satisfy the subnet constraints for each request.

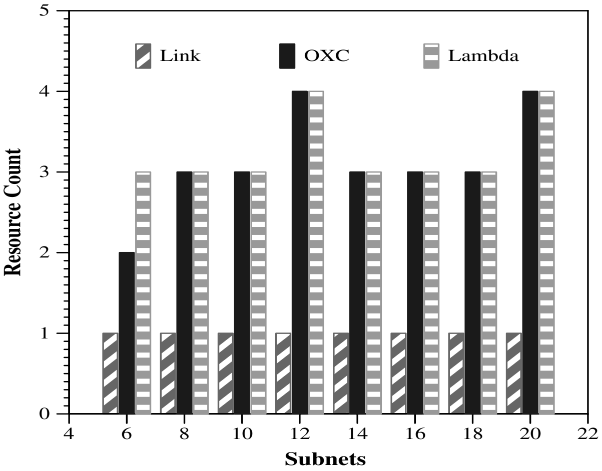


Fig. 9. Max resource requirement for shortest path among all pairs

For all such computed paths, we tabulated the resources required in the core to make the path optically feasible. For each path, we consider the following three parameters that are presented in the graphs:

1) The maximum number of wavelengths(links) required between a pair of OXCs to make the path feasible (*Link*).
2) The maximum number of times any OXC features on the path (*OXC*), and,
3) Number of new λs that need to be dialed to setup this path (*Lambda*).

As shown earlier, if the number of available wavelengths between a pair of OXCs exceeds the number of subnets, every path found by the algorithm will be feasible on the original network. However, in practice having fewer free wavelengths also suffices and the first parameter aims to capture that.

Figure 9 and 10 present the data for the first shortest path among all source destination pairs. Figure 9 plots maximum values while Figure 10 shows the average value of evaluation parameters. Figure 11 depicts the maximum value of the parameters required such that all 10 paths between any source destination pair are feasible.

Figure 9 and 10 show that the algorithm produces feasible paths as long as there are at least 2 free wavelengths. Since typical OXCs have upwards of 128 ports, the network has to be extremely loaded for this to be not true. In fact, Figure 11 shows than the numbers do not change much even if 10 shortest paths were considered. Thus, the number of iterations required will be very small (often, just 1) and hence, in practice *MöbiTwist* not only finds the optimal path but does so extremely efficiently.

IX. Conclusions and Future Work

In this paper, we investigated the problem of routing bandwidth guaranteed paths in a WDM mesh network. We

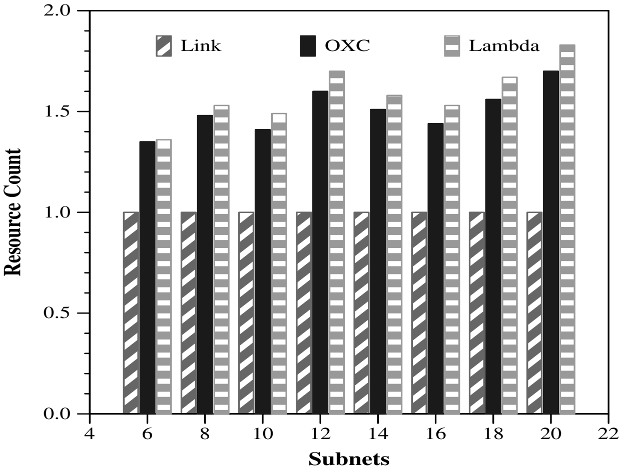


Fig. 10. Average requirement for shortest path among all pairs

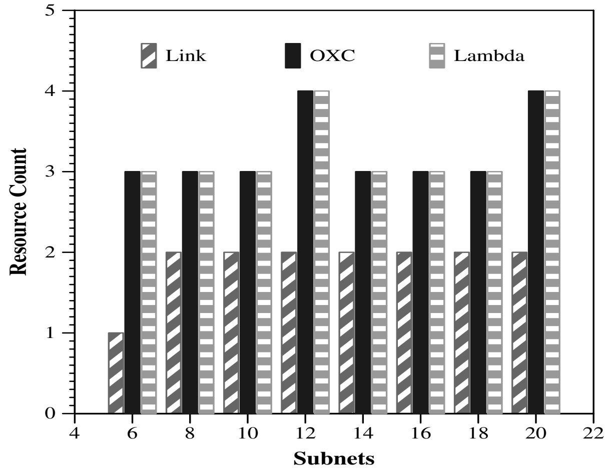


Fig. 11. Max requirement for first 10 shortest paths over all pairs

showed that the constraint imposed by IP subnets on the link connectivity threatens the flexibility and re-configurability that a WDM mesh network aims to provide. We proved that having to honor the subnet constraint transforms the problem of finding the shortest IP hop path, admitting a polynomial solution on a WDM mesh, into a NP-hard problem. Furthermore, we showed that honoring the subnet constraints imposes a penalty of longer paths. Consequently, subnets create a new trade-off of having to make the network less efficient by honoring subnets (due to longer paths) or, accepting an upfront overhead of dynamically changing subnets in order to get more efficient routing paths.

We proposed two shortest path algorithms, *MöbiTwist* and *MöbiFlex*, to route in the presence of subnets. The *MöbiTwist* algorithm finds the optimal shortest hop path honoring the subnet constraints. The *MöbiFlex* algorithm works in the dynamic subnet case wherein given an upper bound on the subnet violations an operator is willing to tolerate, it finds a shorter path than *MöbiTwist* subject to the violation constraint. The algorithm provides a good balance between large-scale network re-configuration and good quality shortest paths. Both the algorithms are efficient and in practice, find the shortest path in polynomial time making them attractive to implement.

This paper has focussed on shortest path routing in the presence of subnets. Over the last few years, there has been a considerable interest in enhanced routing algorithms in WDM meshes such as for shared and dedicated protection. However, these have been considered in the absence of subnets and this paper provides a launching pad to extend those works in the context of subnets.

Till date, there are no mechanisms to automatically change the subnets assigned to interfaces and any updates require manual intervention. We submit that being able to automate the above process will be a key driver for IP-Optical interoperability in WDM mesh networks. Similarly, optimal network design and planning in view of the WDM re-configurability is an open issue. Thus, we believe that going forward management of subnets, currently a one-time configuration issue, will be an active area for research and standards activity.

ACKNOWLEDGMENT

We would like to thank Krishnan Kumaran for initial discussions on the algorithms. We also acknowledge the feedback from the anonymous referees that helped improve the paper.

REFERENCES

[1] S. Acharya et al, "Architecting Self-Tuning Optical Networks," *European Conference on Optical Communication*, 2002.

[2] M. Kodialam and T. V. Lakshman, "Integrated Dynamic IP and Wavelength Routing in IP over WDM Networks," *IEEE infocom*, 2001.

[3] S. Salsano et al, "Off-line Configuration of a MPLS over WDM Network under Time-Varying Offered Traffic," *IEEE Infocom*, 2002.

[4] J. T. Moy, *OSPF: Anatomy of an Internet Routing Protocol*. Addison-Wesley Pub Co, 1998.

[5] R. Perlman, *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*. Addison-Wesley, 1999.

[6] J. Moy et al, "OSPF for IPv6," *IETF RFC 2740*, 1999.

[7] D. Katz and D. Yeung, "Traffic Engineering extensions to OSPF," *IETF draft*.

[8] E. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, 1959.

[9] M. Schwartz and T.E. Stern, "Routing Techniques used in Computer Communication Networks," *IEEE Trans. on Comm.*, vol. COM-28, 1980.

[10] A. Gencata and B. Mukherjee, "Virtual-Topology Adaptation for WDM Mesh Networks Under Dynamic Traffic," *IEEE Infocom*, 2002.

[11] A. Narula-Tam and E. Midiano, "Dynamic load balancing for WDM based packet networks," *IEEE Infocom*, 2000.

[12] J. M. Jaffe, "Algorithms for finding paths with multiple constraints," *Networks*, vol. 14, 1984.

[13] S. Chen and K. Nahrstedt, "On finding multi-constrained paths," *ICC*, 1998.

[14] T. E. Stern and K. Bala, *Multiwavelength Optical Networks: A Layered Approach*. Wesley Longman, Inc., 1999.

[15] D. Awduche et al, "Extension to RSVP for LSP tunnels," *IETF draft*.

[16] B. Rajagopalan, G. Bernstein and D. Spears, "OIF UNI 1.0 Controlling optical networks," *Optical Internetworking Forum, White Paper*, 2001.

[17] M. R. Garey and D. S. Johnson, *Computers and intractability - A Guide to the Theory of NP-Completeness*. Freeman, California, USA, 1979.

[18] E. L. Lawler, "A Procedure for computing the K best solutions to discrete optimization problems and its application to shortest path problem," *Management Science*, 1972.