

# Hierarchical QoS Routing in Next Generation Optical Networks

Ronghui Hou, King-Shan Lui, Fred Baker, and Jiandong Li

**Abstract**—In this paper, we study the problem of inter-domain routing with two additive QoS constraints in hierarchical optical networks. We develop an inter-domain routing protocol that (1) identifies the QoS supported by the paths, (2) selects an inter-domain path that satisfies the QoS requirement of a connection request, and (3) reserves the wavelength on each link along the path in such a way that the number of wavelength converters needed is minimized. Both formal analyses and extensive simulation experiments show that our inter-domain routing protocol outperforms the existing protocols.

**Index Terms**—Hierarchical optical networks, multiple constraints, topology aggregation, wavelength assignment.

## I. INTRODUCTION

AS the next generation of the Internet, the dense wavelength division multiplexing (DWDM) optical networks has attracted lots of attention in the community. ITU-T proposed the Automatic Optical Network (ASON) as the standard of the next generation transport optical networks. ASON aims at provisioning dynamic resources allocation so as to satisfy the demands of dynamic traffic. With the rapid growth of the network coverage in recent years, scalability is one of the major concerns for designing optical networks. For scalability reason, the current Internet is hierarchically structured into multiple domains (or autonomous systems). Similarly, in the ASON/GMPLS (Generalized Multi-protocol Label Switching) framework, a network is also structured in a hierarchical manner, and each node in a domain has no topology information about other domains. Fig. 1(a) shows a simple network with five domains and each domain has two *border nodes*, which are the nodes that are connected to other domains by a physical fiber. The two border nodes in Domain *B* are *B.1* and *B.2*. A domain itself can be another network. For example, Fig. 1(b) shows the internal structure of Domain *B* in Fig. 1(a) where *e* is *B.1* and *d* is *B.2*. A path going from *A.2* to *T.2* through Domain *B* in Fig. 1(a) has to traverse Domain *B* internally. If two border nodes are in the same domain or are directly connected, they are *border neighbors* of each other. For

instance, in Fig. 1(a), *B.1* and *T.2* are both border neighbors of *B.2*.

As the hierarchical nature of the ASON framework is very similar to the Internet, it seems that we can apply the routing protocol we have been using in the Internet to identify inter-domain paths in an optical network as well. Therefore, the BGP (Border Gateway Protocol) protocol of the Internet has been extended to the OBG (Optical Border Gateway Protocol) protocol [1] for optical networks, so that OBG can convey and signal optical information between OBG neighbors. Studies also show that the path-vector mechanism adopted by BGP is suitable for performing routing in ASON [2]. Nevertheless, a simple extension of BGP cannot satisfy the needs of future optical networks. First, current BGP does not have a good support of *Quality-of-Service (QoS)*. Second, current BGP does not consider the *wavelength-continuity constraint* of optical networks. QoS is important because a request may require the signal quality on a path should not be lower than a certain threshold. On the other hand, wavelength-continuity constraint means the same wavelength should be used on the portion of the path where wavelength conversion is not allowed. In this paper, we study how to find a path that satisfies both the QoS requirement and the wavelength-continuity constraint of a connection request in a hierarchical optical network.

We adopt the network model used in [3], where the border nodes (nodes that connect to other domains) are equipped with the wavelength converters, while the internal nodes are not. Therefore, the wavelength-continuity constraint applies to intra-domain paths, where the wavelength used on each link on a path that goes between two nodes in the same domain must be the same. On the other hand, the intra-domain paths in different domains of an inter-domain path can use different wavelengths. For example, all the links on the path from *e* to *d* in Fig. 1(b) must use the same wavelength. Nevertheless, on the inter-domain path from *S.2* to *T.2* that goes through Domains *A* and *B*, the intra-domain path from *A.1* to *A.2* and the intra-domain path from *B.1* to *B.2* can use different wavelengths, since border nodes *A.2* and *B.1* have the capability of wavelength conversion.

Very often, there are multiple paths connecting two nodes and they offer different QoSes. In this paper, we consider two additive QoS requirements as in [4], [2], such as SNR degradation and (monetary) cost<sup>1</sup>. The cost (or SNR degradation) metric of a path is defined as the sum of the costs (or SNR degradations) of all the links on this path [4]. Each request specifies the cost and SNR degradation it can tolerate. A *feasible path* of a request is

<sup>1</sup>A multiplicative metric, such as reliability, can be treated as an additive one as well after applying logarithm.

Manuscript received December 10, 2009; revised April 22, 2010; accepted May 24, 2010. Date of publication June 14, 2010; date of current version July 21, 2010. This work is supported in part by the Cisco Research Initiative Award.

R. Hou and J. Li are with the State Key Laboratory of ISN, Xidian University, China (e-mail: rhhou@pcn.xidian.edu.cn; jdli@pcn.xidian.edu.cn).

K.-S. Lui is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: kslui@eee.hku.hk).

F. Baker is with Cisco Research Center, San Jose, CA 95134 USA (e-mail: fred@cisco.com).

Digital Object Identifier 10.1109/JLT.2010.2052787

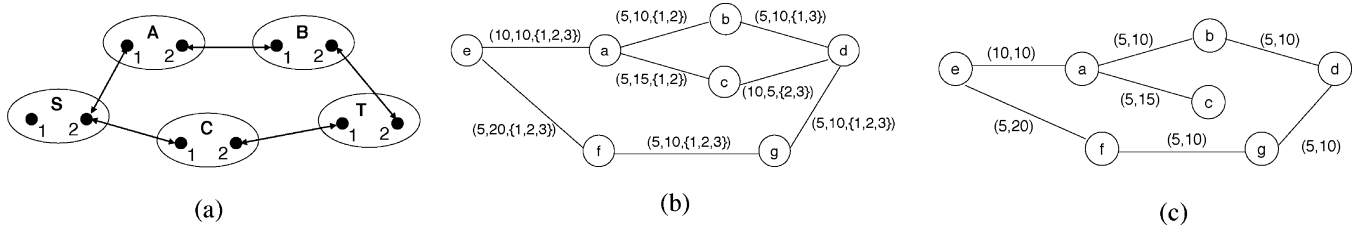


Fig. 1. Illustration for different topologies. (a) A hierarchical network. (b) Domain B. (c) Domain B with wavelength 1.

a path whose cost and SNR degradation do not exceed what the request can tolerate [4]. Although we refer our QoS metrics as cost and SNR degradation, our mechanism can be used for any two independent additive QoS metrics, such as delay, hop count number, or reliability [4].

Our goal is to identify the QoS of inter-domain paths under the path-vector framework adopted by the BGP/OBGP protocol, so that feasible paths for serving requests can be found. In the path-vector approach, each border node computes the (intra-domain) paths information between any two border nodes in the same domain, and then advertises the information to its border neighbors. Based on the received path information, each border node can compute the paths across multiple domains. We now use an example in Fig. 1(a) to illustrate the process of inter-domain path computation. We consider the process of computing the path information from  $S.2$  to the destination domain  $T.2$ .  $B.2$  and  $C.2$  are directly connected to domain  $T$ , and they advertise the path information from themselves to  $T$  to  $B.1$  and  $C.1$ , respectively. Secondly,  $B.1$  and  $C.1$  compute the paths from themselves to domain  $T$ , and advertise them to  $A.2$  and  $S.2$ , respectively. The process continues until  $S.2$  receives the paths information from  $A.1$  to  $T$  and that from  $C.1$  to  $T$ . Based on the received path information,  $S.2$  can compute the paths from itself to domain  $T$ . There are two issues we need to solve in developing a QoS inter-domain routing in an optical network.

- 1) How to compute the path information between two border neighbor nodes (e.g.,  $A.1$  and  $A.2$ ) of the same domain under the wavelength-continuity constraint?
- 2) How to select a path that traverses several domains, e.g.,  $S.2$  to domain  $T$ , and assign the wavelength on the selected inter-domain path?

To the best of our knowledge, our work is the first proposal for studying the inter-domain routing with two additive QoS constraints in optical networks. The contributions of this paper are as follows.

- 1) We propose an algorithm to extract the QoS information supported by all the intra-domain paths between any two border nodes in the same domain.
- 2) We develop the inter-domain path selection mechanism and the wavelength assignment mechanism. Our scheme assigns wavelengths in a way that the number of wavelength converters needed is minimized.

The rest of the paper is organized as follows. In Section II, we present how to compute the intra-domain path information with the wavelength-continuity constraint. Section III describes our inter-domain QoS routing protocol. We describe our simulation experiments in Section IV and related works in Section V.

## II. INTRA-DOMAIN PATHS COMPUTATION

We first present our network model. Each domain is represented by  $(V, E, B, W)$ , where  $V$  is the node set,  $E$  is the edge set, and  $B$  is the border node set in this domain. We thus have  $B \subseteq V$ . The set of wavelengths available is  $W = \{1, \dots, w\}$ . The QoS metric of each link is assumed to be the same for all wavelengths on this link<sup>2</sup>. A link goes from node  $a$  to node  $b$  is represented as  $(a, b)$ . The cost and degradation of  $(a, b)$  are  $c_{(a,b)}$  and  $d_{(a,b)}$ , respectively. We call the tuple  $(c_{(a,b)}, d_{(a,b)})$  the QoS parameter of link  $(a, b)$ . The set of wavelengths available in  $(a, b)$  is  $\lambda(a, b)$ , where  $\lambda(a, b) \subseteq W$ . In Fig. 1(b), link  $(b, d)$  is associated with the tuple  $(5, 10, \{1, 3\})$ , meaning the cost and degradation metrics on this link are 5 and 10, respectively, while the wavelengths 1 and 3 are available on this link. An intra-domain path  $p$  traversing nodes  $v_1, v_2, \dots, v_{h-1}$ , and  $v_h$  is presented as  $\langle v_1, \dots, v_h \rangle$ . Because of the wavelength-continuity constraint, all links on this path,  $(v_i, v_{i+1})$  where  $i = 1, \dots, h-1$ , have to use the same wavelength. The cost, degradation, and wavelengths available for  $p$  are denoted as  $c_p, d_p$ , and  $\lambda(p)$ , respectively, where

$$\begin{cases} c_p = \sum_{i=1}^{h-1} c_{(v_i, v_{i+1})} \\ d_p = \sum_{i=1}^{h-1} d_{(v_i, v_{i+1})} \\ \lambda(p) = \bigcap_{i=1}^{h-1} \lambda(v_i, v_{i+1}) \end{cases} \quad (1)$$

The cost and degradation of the path  $\langle e, a, c, d \rangle$  in Fig. 1(b) are 25 and 30, respectively. Moreover, we can see that  $\lambda(\langle e, a, c, d \rangle) = \{2\}$ .

We assume a request specifies the cost and degradation it can tolerate in the form of  $(c_{\text{req}}, d_{\text{req}})$ . The request is *feasible* if there exists a path  $p$  such that  $c_{\text{req}} \geq c_p$  and  $d_{\text{req}} \geq d_p$ . Therefore, if a node obtains the *supported QoS information* from itself to a destination, it can determine whether any incoming request is feasible. Before describing how to obtain the QoS information, we first introduce some important definitions, which are also defined in [5].

**Definition 1:** An QoS parameter  $(c, d)$  is more representative than another different QoS parameter  $(c', d')$ , denoted as  $(c, d) \prec (c', d')$ , if and only if  $c \leq c'$  and  $d \leq d'$ .

<sup>2</sup>It is worth noting that the QoS metric can also be defined in a *per wavelength* manner. Our inter-domain protocol can be easily extended for this case but we leave out the details for the ease of discussion.

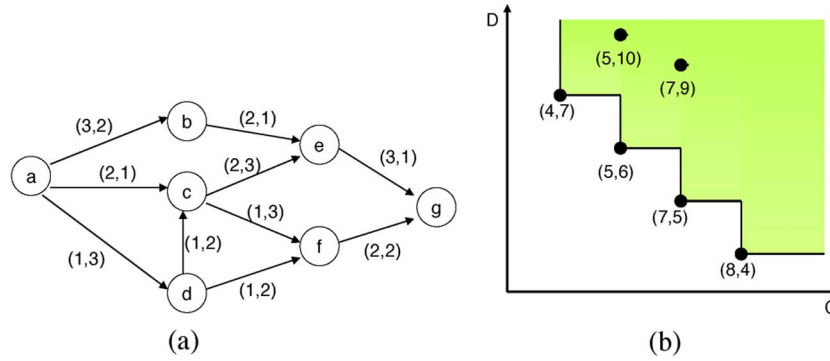


Fig. 2. Illustration for supported QoS. (a) A simple domain topology. (b) Presentation on plane C-D.

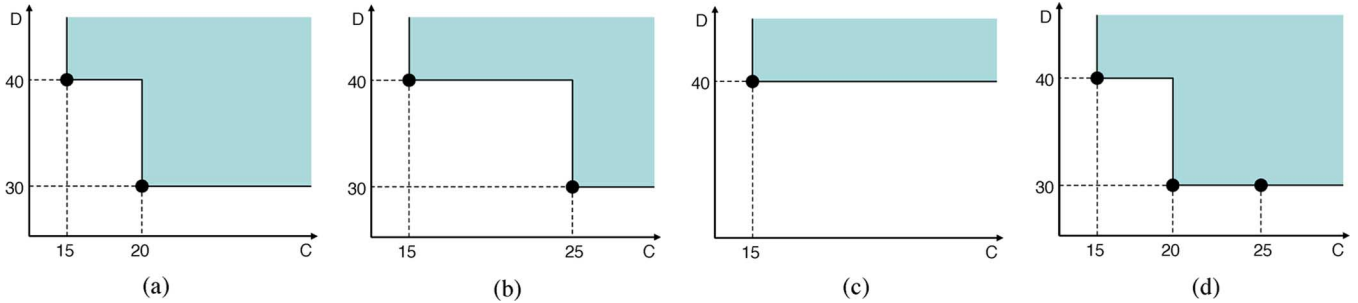


Fig. 3. Example for the supported QoS defined by different wavelengths. (a) Wavelength 1. (b) Wavelength 2. (c) Wavelength 3. (d) All wavelengths.

*Definition 2:* The QoS parameter  $(c_1, d_1)$  of a path from a source  $s$  to  $t$  is a *representative point* if we cannot find another path from  $s$  to  $t$  with the QoS parameter  $(c_2, d_2)$  such that  $(c_2, d_2) \prec (c_1, d_1)$ . The path with the QoS parameter  $(c_1, d_1)$  is called a *non-dominated path*.

Consider the simple domain in Fig. 2(a) where all the links use the same wavelength. There are six paths going from  $a$  to  $g$ . The path  $p_1 = \langle a, c, f, g \rangle$  has the QoS parameter  $(5, 6)$ , while the path  $p_2 = \langle a, d, c, f, g \rangle$  has the QoS parameter  $(5, 10)$ . According to Definition 1, we have  $(5, 6) \prec (5, 10)$ . Any request supported by path  $p_2$  can be supported by  $p_1$  as well. We plot the QoS parameters of all the paths from  $a$  to  $g$  on the C(ost)-D(egradation) plane as shown in Fig. 2(b). We call the shaded area the *feasible region*. If the requirement of a request falls in the feasible region, we can find a path satisfying this request. For example, given a request with the QoS requirement  $(6, 7)$ , path with the QoS parameter  $(5, 6)$  satisfies the requirement of this request, and so this request is feasible. On the other hand, the request with the requirement  $(4, 6)$  is infeasible, since the requirement falls outside the feasible region. By Definitions 1 and 2, the point  $(4, 7)$  is a representative point, and the path with the QoS parameter  $(4, 7)$  is a non-dominated path. The representative points can uniquely define the feasible region, which is also the supported QoS between the nodes.

To find the supported QoS between two nodes on all wavelengths, we first divide the network into several subnetworks, where each subnetwork contains all the links which have the same available wavelength. If there are  $w$  available wavelengths in the network, we can get  $w$  subnetworks. We then compute the supported QoS of each subnetwork. After getting the supported QoS for each wavelength, the union of the supported QoSes is

the total supported QoS between the two nodes. To find the supported QoS between two nodes with a certain wavelength, we need to find all the representative points on the C-D plane as shown in Fig. 2(b). Since both cost and degradation are additive metrics, the problem is in fact NP-complete. We can apply existing approximation algorithms to estimate the supported QoS [4], [6].

For example, Fig. 1(c) illustrates the subnetwork with wavelength 1 which is induced from Fig. 1(b). The supported QoS from  $e$  to  $d$  with the wavelength 1 is defined by the points  $(20, 30)$  and  $(15, 40)$ . We plot these two QoS parameters on the C-D plane, as illustrated in Fig. 3(a), and the shaded area is the *feasible region*. With the same method, the supported QoS from  $e$  to  $d$  with wavelength 2 is defined by  $(25, 30)$  and  $(15, 40)$ , as illustrated in Fig. 3(b), and that with wavelength 3 is defined by  $(15, 40)$ , as illustrated in Fig. 3(c). We can see that the supported QoSes of different wavelengths are different. The union of the supported QoSes for different wavelengths is thus the total supported QoS.

As aforementioned, the border node in each domain is equipped with the wavelength converters. Different domains can apply different wavelengths. This implies that each border node does not have to advertise the available wavelength information of its intra-domain paths. Moreover, to an outside domain, the supported QoS of routing between two border nodes in this domain is the union of the supported QoSes for all wavelengths. Therefore, each border node just needs to advertise the union of the supported QoSes for all wavelengths. Fig. 3(d) illustrates the union of the supported QoSes from  $e$  to  $d$  for all wavelengths 1, 2, and 3. The total supported QoS can be defined by the points  $(15, 40)$  and  $(20, 30)$ . Note that

a certain QoS point can be supported by multiple paths with the same or different wavelengths. For example, the path with the QoS parameter (15, 40) can be provisioned by wavelengths {1, 2, 3}, while the path with the QoS parameter (20, 30) must use wavelength 1.

### III. INTER-DOMAIN QOS ROUTING

#### A. Inter-Domain Path Computation

In this subsection, we describe how to compute the inter-domain supported QoS based on the intra-domain supported QoS. Denote  $R(b_s, b_u) = \{r_1, \dots, r_m\}$  and  $R(b_u, b_d) = \{r'_1, \dots, r'_n\}$  as the supported QoSes (set of representative points) from the border node  $b_s$  to the border neighbor node  $b_u$  and from  $b_u$  to the destination border node  $b_d$ , respectively. To understand how to compute the supported QoS from  $b_s$  to  $b_d$  via  $b_u$ , we first consider the QoS of the concatenation of paths  $p_1$  and  $p_2$ , where  $p_1$  is a non-dominated path from  $b_s$  to  $b_u$  and  $p_2$  is a non-dominated path from  $b_u$  to  $b_d$ . We let  $p$  be the concatenated path and  $p = p_1 \oplus p_2$ . We further let the QoS of  $p_1$  be  $r = (r.c, r.d)$  and the QoS of  $p_2$  be  $r' = (r'.c, r'.d)$ . As both cost and degradation are additive, the QoS of  $p$ , denoted  $r \oplus r'$ , is  $(r.c + r'.c, r.d + r'.d)$ . Therefore, to find the supported QoS from  $b_s$  to  $b_d$  via  $b_u$ , we can first find out all possible  $r_i \oplus r'_j$  where  $r_i \in R(b_s, b_u)$  and  $r'_j \in R(b_u, b_d)$ . There are  $m \cdot n$  possible points in total. Then we identify the representative points among these points.

We now present an example to illustrate how to find out the supported QoS across several domains. In Fig. 1(a), assume the QoS parameter of the link ( $B.2, T.2$ ) is (5, 5). First,  $B.2$  advertises (5, 5) to  $B.1$ . The supported QoS from  $B.1$  to  $B.2$ , as shown in Fig. 1(c), is  $\{(15, 40), (20, 30)\}$ .  $(15, 40) \oplus (5, 5) = (15 + 5, 40 + 5) = (20, 45)$  and  $(20, 30) \oplus (5, 5) = (25, 35)$ . Thus,  $B.1$  knows that the supported QoS from itself to  $T.2$  is  $\{(20, 45), (25, 35)\}$ .  $B.1$  then advertises  $\{(20, 45), (25, 35)\}$  to  $A.2$ . Now suppose the supported QoS from  $A.2$  to  $T.2$  is  $\{(30, 55), (35, 45)\}$ , and the supported QoS from  $A.1$  to  $A.2$  is  $\{(10, 30), (20, 25)\}$ .  $A.1$  can obtain four points (40, 85), (50, 80), (45, 75), (55, 70). However, (50, 80) is not a representative point since  $(45, 75) \prec (50, 80)$ . As a result, the supported QoS from  $A.1$  to  $T.2$  is defined by three representative points  $\{(40, 85), (45, 75), (55, 70)\}$ . We can see that as the number of domains increases, the number of representative points increases, and so the advertisement overhead increases. The work in [5] presents a supported QoS information aggregation method which uses a constant number of points to approximately represent the inter-domain supported QoS. Nevertheless, error will be introduced when an approximation is used. How to balance the tradeoff is outside the scope of this paper. We refer readers to [5] for detailed discussion.

A node may have several neighbors, and the QoSes to a certain destination through different neighbors would be different. For example,  $S.2$  in Fig. 1(a) has two neighbors leading to  $T.2$  using different paths. The supported QoS from  $A.1$  to  $T.2$  and the supported QoS from  $C.1$  to  $T.2$  may not be the same. When  $S.2$  advertises the supported QoS from itself to  $S.1$ , it should advertise the union of the supported QoSes of the two paths. In general, if  $A_b(b_s)$  is the set of the border neighbors of  $b_s$ , the

TABLE I  
THE INTRA-DOMAIN ROUTING TABLE OF NODE  $B.1$  IN FIG. 1

Destination	QoS to Destination		
	QoS parameter	Wavelength	Node List
$B.2$	(15, 40)	{1,2,3}	< $B.1, f, g, B.2$ >
	(20, 30)	{1}	< $B.1, a, c, B.2$ >

supported QoS from  $b_s$  to  $b_d$  is the representative points in the set  $\bigcup_{b_u \in A_b(b_s)} \{r \oplus r' \mid r \in R(b_s, b_u) \text{ and } r' \in R(b_u, b_d)\}$ .

#### B. Inter-Domain Path Selection

Apart from computing the QoS to a certain destination based on the advertisements from neighbors, a border node should also construct the routing table for packet forwarding. In this section, we first describe the routing table of each border node, and then illustrate how to identify an inter-domain path for a request based on the routing table.

Each border node keeps two routing tables: *inter-domain routing table* and *intra-domain routing table*. The inter-domain routing table provides information for border nodes selection in the inter-domain path, while intra-domain routing table is responsible for the intra-domain path selection between any two border nodes in the same domain.

In the intra-domain routing table, for each border node within the same domain, we keep a certain number of the non-dominated intra-domain paths leading to that node. The number of the non-dominated paths can be an adjustable parameter in practical implementation. It is obvious that the larger the number of non-dominated paths, the more accurate the supported QoS. For each non-dominated path, the QoS parameter and the available wavelength information are recorded. Table I illustrates the intra-domain routing table of border node  $B.1$ . There are two non-dominated paths from  $B.1$  to  $B.2$ . In each entry of the intra-domain routing table, "QoS parameter" denotes the two additive metrics of this path, "Wavelength" keeps the information about the available wavelength resources, and "Node List" specifies the set of nodes on this path. For example, according to the first entry, we know that path  $\langle B.1, f, g, B.2 \rangle$  has the QoS parameter (15, 40) and the available wavelengths {1, 2, 3}. The path information can be obtained by a link-state protocol, such as OSPF. If a distance-vector protocol is used instead, we keep the next hop information instead of the whole path under "Node List".

The inter-domain routing table keeps the QoS information of routing to the border nodes in other domains. Each entry for each destination represents an inter-domain path. In each entry of the inter-domain routing table, as illustrated in Table II, "QoS parameter" specifies the two additive metrics of the path from this border node to the destination, and "Next border" determines the next border node on this inter-domain path. When the next border node is a border node in the same domain, the "Intra-domain Path" field keeps the QoS information leading to this border node, which is the QoS parameter of a certain intra-domain path. If the next border node is a node in another domain, we do not have to specify the intra-domain path field, because the QoS information leading to this border in another domain is the QoS parameter of a fiber link. Table II illustrates

TABLE II  
THE INTER-DOMAIN ROUTING TABLE OF NODE *A.1* IN FIG. 1

Destination	QoS to Destination		
	QoS parameter	Next border	Intra-domain Path
<i>T.2</i>	(40, 85)	<i>A.2</i>	(10,30)
	(45, 75)	<i>A.2</i>	(10,30)
	(55, 70)	<i>A.2</i>	(20,25)

the inter-domain routing table of border node *A.1*. There are three non-dominated paths from *A.1* to *T.2* and they all go through *A.2*, another border node in Domain *A*. Therefore, the QoS information from *A.1* to *A.2* of all three non-dominated paths are specified in the Intra-domain Path field. Note that the QoS parameters of the intra-domain paths for the first two entries are both (10, 30) because the two corresponding inter-domain paths go through the same intra-domain path from *A.1* to *A.2*.

A node considers updating its inter-domain routing table when it receives an advertisement from its border neighbor. The advertisement contains the QoS information of the non-dominated path from the border neighbor to a destination. The node can then identify the non-dominated paths from itself to the destination via the border neighbor based on the mechanism described in Section III.A. It updates the inter-domain routing table and advertises the information if necessary.

Based on the inter-domain routing table, a node can determine immediately whether a request with the QoS requirements is feasible. For example, suppose *A.1* receives a request to go to *T.2* with QoS requirements (45, 80). Since  $(45, 75) \prec (45, 80)$ , *A.1* knows that there is a feasible path for the request. Now, we discuss how to identify an inter-domain path satisfying the QoS requirements. When the source wants to identify an inter-domain path for a feasible connection request, it generates a *REV\_REQ* packet. The *REV\_REQ* packet carries the QoS requirements imposed by the connection request for routing from the current border node to a destination. When a border node receives the *REV\_REQ* packet, it finds the next border neighbor based on its inter-domain routing table. If it cannot find a feasible inter-domain path, it will drop the *REV\_REQ* packet, and informs the source that a path cannot be established for the request. If there are multiple feasible paths, the border node randomly selects one.

If the next border neighbor is in the same domain, the border node will identify the intra-domain path to the next border neighbor based on its intra-domain routing table. No matter the next border node is within the same domain or not, the current border node should update the QoS requirements in *REV\_REQ* before sending it out. If the QoS parameter of routing from the current border node to the next border node is  $(c, d)$  and the original QoS requirement in *REV\_REQ* is  $(c', d')$ , the QoS requirements in *REV\_REQ* is updated to  $(c' - c, d' - d)$ . This process continues until the destination receives *REV\_REQ*. If the destination receives *REV\_REQ*, the connection establishment succeeds. Algorithm 1 presents the pseudocodes of our mechanism. In Algorithm 1, *REV\_REQ* packet is represented by the tuple  $((c_{\text{req}}, d_{\text{req}}), s, t, b_c, r_s, \lambda)$ .  $(c_{\text{req}}, d_{\text{req}})$  denotes the QoS requirements of routing from the current border node to

the destination, while *s* and *t* are the source and the destination of the whole path, respectively.  $b_c$  is the next intended recipient of the *REV\_REQ* packet,  $r_s$  is a certain border node this packet has traversed, and  $\lambda$  denotes the common available wavelengths on the route from  $r_s$  to  $b_c$  over which this packet has traversed. We will describe how to update  $b_c, r_s$ , and  $\lambda$  in the next section.

---

#### Algorithm 1 Path Section for border node $b_c$

---

##### parameter

$((c_{\text{req}}, d_{\text{req}}), s, d, b_c, r_s, \lambda)$ : *REV\_REQ* packet initiated by *s* and received by  $b_c$   
 $(c_{\text{req}}, d_{\text{req}})$ : The current QoS requirements of routing from  $b_c$  to *d*

```

1: if  $b_c = d$  then
2:   Perform the wavelength reservation procedure.
3:   The path selection process terminates.
4: else
5:   Feasible  $\leftarrow$  false
6:   for each entry E in the inter-domain routing table do
7:      $r \leftarrow$  the QoS parameter of E
8:     if  $r \prec (c_{\text{req}}, d_{\text{req}})$  then
9:       Feasible  $\leftarrow$  true
10:    break
11:   if Feasible then
12:      $u \leftarrow$  Next border of E
13:     if u is in the same domain as  $b_c$  then
14:        $p \leftarrow$  the Intra-domain path of E
15:        $c_{\text{req}} \leftarrow c_{\text{req}} - c_p$ 
16:        $d_{\text{req}} \leftarrow d_{\text{req}} - d_p$ 
17:        $\lambda' \leftarrow \lambda \cap \lambda(p)$ 
18:       else
19:          $c_{\text{req}} \leftarrow c_{\text{req}} - c_{(b_c, u)}$ 
20:          $d_{\text{req}} \leftarrow d_{\text{req}} - d_{(b_c, u)}$ 
21:          $\lambda' \leftarrow \lambda \cap \lambda(b_c, u)$ 
22:       if  $\lambda' = \emptyset$  then
23:         Performs the wavelength reservation procedure.
24:        $r_s \leftarrow b_c$ 
25:        $\lambda \leftarrow \lambda(p)$ 
26:       else
27:          $\lambda \leftarrow \lambda'$ 
28:       send  $((c_{\text{req}}, d_{\text{req}}), s, d, u, r_s, \lambda)$  to u
29:       else
30:         drop  $((c_{\text{req}}, d_{\text{req}}), s, d, b_c, r_s, \lambda)$ 
31:       inform the previous node that the route request is rejected
    
```

#### C. Wavelength Assignment

Apart from finding a feasible path, the *REV\_REQ* packet also allows border nodes to allocate appropriate wavelengths on the links on the path. To reduce the cost, we would like to minimize the number of wavelength converters used.  $\lambda$  in *REV\_REQ* carries the wavelengths available on all the links on a path from  $r_s$  to  $b_c$ . When  $b_c$  receives this *REV\_REQ* packet. It should inform its next hop *u* the wavelengths available from  $r_s$  to *u*. No matter  $b_c$  and *u* are separated by an intra-domain path or a fiber link,  $b_c$  knows the set of available wavelengths  $(\lambda(b_c, u))$  for going to *u*.  $b_c$  can obtain the set of available wavelengths from  $r_s$  to

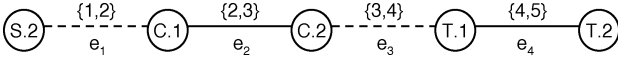


Fig. 4. An inter-domain path.

$u$  by computing  $\lambda' = \lambda \cap \lambda(b_c, u)$ . If  $\lambda' \neq \emptyset$ ,  $b_c$  updates the information carried in REV\_REQ to  $\lambda'$ , and sends the message to  $u$ . We do not know which wavelength to use yet because we do not know which wavelength is also available from  $u$  to the destination. If  $\lambda' = \emptyset$ , it means that the path from  $r_s$  to  $b_c$  does not share any wavelength with that from  $b_c$  to  $u$ . In this case, no matter which wavelength is used from  $r_s$  to  $b_c$ ,  $b_c$  has to convert the wavelength. Therefore,  $b_c$  can select a wavelength in  $\lambda$ , and informs  $r_s$  over the selected path.  $\lambda$  carried in REV\_REQ is then updated to  $\lambda(b_c, u)$ . When the destination receives REV\_REQ, it can select a wavelength for the last portion of the path.

We now illustrate the process based on the network in Fig. 4. To simplify our discussion, we only describe fields  $b_c$ ,  $r_s$ , and  $\lambda$  in REV\_REQ packets in the format of  $\langle b_c, r_s, \lambda \rangle$ . S.2 creates the first REV\_REQ packet  $\langle C.1, S.2, \{1, 2\} \rangle$  and sends it to C.1. After receiving the packet, C.1 computes  $\{1, 2\} \cap \{2, 3\}$ , the wavelength available from S.2 to its neighbor C.2. As  $\{1, 2\} \cap \{2, 3\} = \{2\} \neq \emptyset$ , C.1 sends  $\langle C.2, S.2, \{2\} \rangle$  to C.2. C.2 realizes that  $\{2\} \cap \{3, 4\} = \emptyset$ , and so it can inform C.1 and S.2 that wavelength 2 should be used from S.2 to C.2. Besides, it sends  $\langle T.1, C.2, \{3, 4\} \rangle$  to T.1. T.1 sends  $\langle T.2, C.2, \{4\} \rangle$  to T.2. T.2 can then inform T.1 and C.2 that wavelength 4 should be used from C.2 to T.2. In this example, only one converter is needed, which is optimal.

Note that a path using  $k$  converters can be partitioned into  $k + 1$  subpaths that are separated by the converters. To formally prove our mechanism can always minimize the number of wavelength converters needed on a path  $\langle 1, 2, \dots, N \rangle$ , we first develop the following property of our scheme.

*Property 1:* If our algorithm assigns nodes  $i_1, i_2, \dots, i_k$  as converters,  $\bigcap_{m=i_j}^{i_{j+1}-1} \lambda(m, m+1) \neq \emptyset$  but  $\bigcap_{m=i_j}^{i_{j+1}} \lambda(m, m+1) = \emptyset$  for  $j = 0, \dots, k-1$  and  $i_j = 1$ .

For example, refer to Fig. 4,  $\lambda(S.2, C.1) \cap \lambda(C.1, C.2) \neq \emptyset$  but  $\lambda(S.2, C.1) \cap \lambda(C.1, C.2) \cap \lambda(C.2, T.1) = \emptyset$ .

*Lemma 1:* Our wavelength assignment mechanism is optimal in minimizing number of wavelength converters.

*Proof:* Assume to the contrary that our mechanism is not optimal. That is, on a particular path, there is an optimal wavelength assignment using  $k'$  converters while our algorithm finds a solution using  $k$  converters where  $k > k'$ . Suppose that the optimal solution applies converters at nodes  $i'_1, i'_2, \dots, i'_k$ , while our solution applies converters at nodes  $i_1, i_2, \dots, i_k$ . According to the optimal solution,  $\bigcap_{m=1}^{i'_1-1} \lambda(m, m+1) \neq \emptyset$ . It suggests that  $i_1 \geq i'_1$  by Property 1. On the other hand,  $k > k'$  implies there exists  $j$  such that  $i_j < i'_j$  while  $i_{j-1} \geq i'_{j-1}$ , as shown in Fig. 5. Note that  $\langle i_{j-1}, \dots, i_j \rangle$  is a subpath of  $\langle i'_{j-1}, \dots, i'_j \rangle$ . No converter is needed in  $\langle i'_{j-1}, \dots, i'_j \rangle$  implies  $\bigcap_{m=i'_{j-1}}^{i'_j-1} \lambda(m, m+1) \neq \emptyset$ . On the other hand, our algorithm applies a converter at  $i_j$  implies  $\bigcap_{m=i_{j-1}}^{i_j} \lambda(m, m+1) = \emptyset$  according to Property 1. It leads to contradiction. ■



Fig. 5. Proof for Lemma 1.

#### D. Network State Update Policy

The wavelength resources information changes after the network accepts a new connection or an existing traffic terminates. The change of the wavelength resources may cause the supported QoS between any two border nodes to change as well. If we re-compute the supported QoS whenever a change occurs, the computational complexity is huge. We call a routing mechanism where the QoS information supported by the network is computed dynamically based on the current link information *adaptive routing* [7]. Our inter-domain routing protocol applies the combination of *adaptive routing* and *fixed-alternate routing* mechanisms.

After a domain accepts a connection request, a wavelength, say  $w_0$ , on all the links of the selected intra-domain path should be reserved for this connection. Then, the supported QoS with wavelength  $w_0$  should be updated, and the total supported QoS may be changed subsequently. Our intra-domain supported QoS information is updated after a new connection is established, or an existing connection terminates, or an interior node (or a fiber link) failure happens, and so we call our intra-domain routing the *adaptive routing*.

If the intra-domain supported QoS of a certain domain changes, theoretically, the border node should advertise the newly-computed supported QoS to the border neighbors in other domains, so as to update the inter-domain supported QoS. However, if the inter-domain supported QoS is updated in a per-connection basis, the advertisement overhead in the network and the computational overhead of each router will be huge. To reduce the overhead, we apply the *fixed-alternate routing* to compute the inter-domain supported QoS. The inter-domain supported QoS in the whole network is updated only if the network topology changes, such as the difference between the newly-computed intra-domain supported QoS and the old one exceeds a certain threshold. Because we do not always advertise new inter-domain QoS information, the source may determine an infeasible request to be feasible. In our path selection process, the intermediate node may detect the failure of the connection establishment. In this case, the wavelength reserved by all the predecessors should be released. On the other hand, it is possible for a source to reject a feasible connection request due to the inaccuracy of the inter-domain supported QoS, which will increase the blocking probability. In our simulation results, we use the blocking probability as the evaluation metric to test the performance of our routing protocol.

#### IV. PERFORMANCE EVALUATION

We conducted simulation experiments in several multi-domain networks to evaluate the performance of our routing pro-

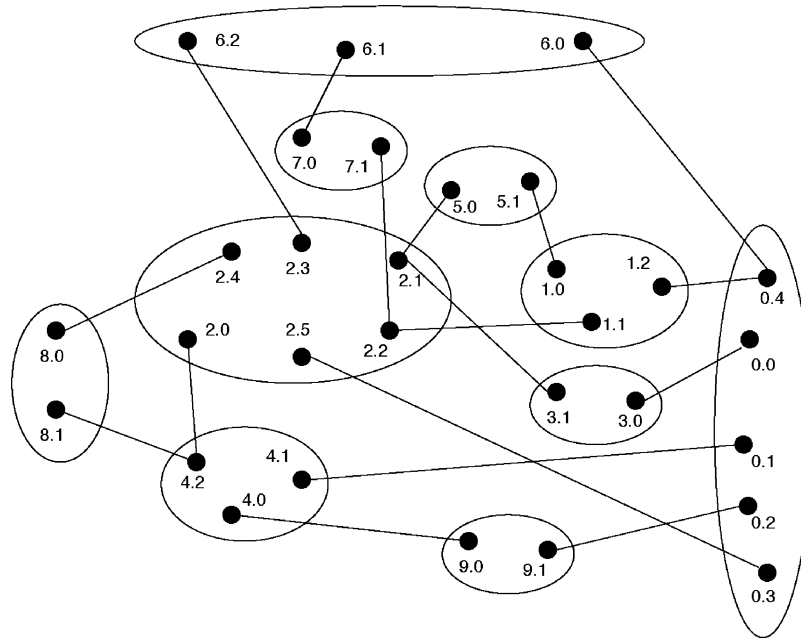


Fig. 6. Our Internet topology.

tol. The work in [3] selects the minimum hop count inter-domain path for the connection request. The intra-domain path is also selected based on the minimum hop count. Although the work in [8] considers two additive QoS metrics, inter-domain hop count and bit error rate (BER), during path selection, only one metric is considered. Therefore, both works in [3] and [8] in essence consider one additive metric. We compare our mechanism with [8] in this paper.

We studied the performance of the mechanisms on three different kinds of networks. The first two were those studied in [3] and [8]. We denote them as TOPO1 (Fig. 4 in [3]) and TOPO2 (Fig. 4 in [8]). Since the hierarchical optical network is considered as the next generation Internet, testing the current Internet topologies is desirable. We apply the BRITE [9] generator to generate the network as shown in Fig. 6. BRITE software is widely applied by the research community in the Internet. The intra-domain topology is generated based on the Waxman’s model, and the inter-domain topology is generated based on the Barabasi-Albert model. Our BRITE network contains 10 domains, and each domain contains 20 nodes. Fig. 6 only shows the border nodes and the inter-domain links among border nodes. There are 30 border nodes and 34 directed inter-domain links in total. Links are asymmetric. The cost and degradation of the intra-domain links in the three networks follow the uniform distribution  $U[1, 10]$ , while those of inter-domain links follow the uniform distribution  $U[5, 20]$ .

In our simulation experiments, the connection requests arrive according to a Poisson process with call holding time being negatively exponentially distributed. Sources and destinations are randomly selected among the border nodes in the network. Given a source  $bs$  and a destination  $bd$ , denote  $(c_l, d_u)$  and  $(c_u, d_l)$  as the minimum cost and degradation paths, respectively. We can easily verified that  $c_u$  and  $d_l$  are the maximum cost and degradation in all the representative points. The cost and degradation constraints of the connection request from  $bs$

to  $bd$  follow  $U[c_l, fc_u]$  and  $U[c_d, fd_u]$ , respectively, where  $f$  is a parameter in the simulation. We test the *blocking probability* (BP) produced by the different protocols under different scenarios. We define the *blocking probability as the ratio of the number of connection requests rejected to the total number of connection requests*. A connection request may be rejected at the source node because it is an infeasible request according to the routing tables. A request can also be rejected during the path setup process in allocating wavelength. In this case, the source has determined it to be feasible but it turns out that it is not. This happens because we do not update inter-domain path information whenever the QoS information changes, which introduces inaccuracies in the routing tables. Smaller BP implies better performance.

For each topology, there are 20 000 connection requests with the randomly selected source and destination from all the border nodes in the networks. For each topology, 10 different configurations for the connections are generated, and so the result is the average of 10 different points. Only intra-domain path information is updated during the simulation time, but not inter-domain path QoS. We study the results of using different  $f$  values. Larger  $f$  would facilitate the generation of more feasible requests. Each border node keeps several alternate paths to all other border nodes. These paths are for carrying the data flow between two border nodes. In TOPO2, each border node keeps at most 8 intra-domain paths to nodes in the same domain and 8 inter-domain paths to a border node in another domain. In TOPO1 and BRITE topologies, at most 12 intra-domain paths and 20 inter-domain paths are kept.

We first study the blocking probabilities with different network loads when the number of wavelengths per link is 8. The network load (Erlang) is calculated by the average connection holding time divided by the average new connection arrival interval. For instance, if the average connection holding time is 200 s, and a new connection will arrive in an average of 2 s,



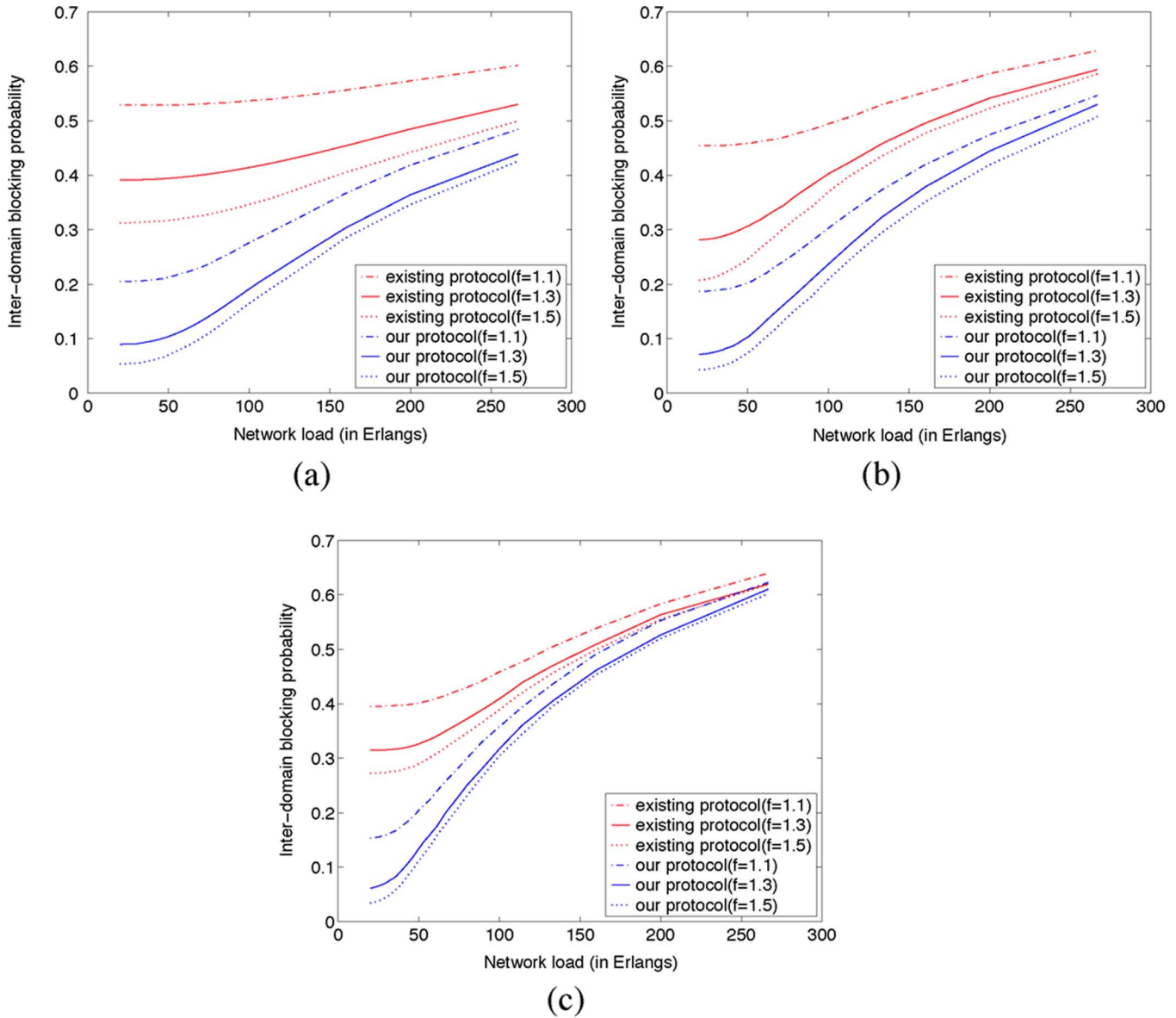


Fig. 7. Performance with varying traffic load when there are 8 wavelengths on each link. (a) BRITE topology. (b) TOPO1. (c) TOPO2.

then, the network load is 100 Erlang. The simulation results of the three topologies are shown in Fig. 7.

Generally speaking, our routing protocol performs better than the existing protocol, since the blocking probability of our protocol is lower than that of the existing protocol. As expected, Fig. 7 shows that the BPs of both protocols grow with the increase of the network load. When the network load is light, it is likely that a feasible path exists for a connection request. However, whether the connection is blocked would also depend on whether the routing mechanism can identify the path. The existing protocol in [8] selects the inter-domain paths according to either metric but not both, while our protocol compares two paths based on the two metrics simultaneously. For instance, given three paths with the QoS parameters (2, 8), (3, 9), and (4, 5). The existing protocol will select the two QoS parameters (2, 8) and (3, 9), but our protocol will choose (2, 8) and (4, 5). It can be easily verified that the supported QoS region found by our protocol is larger than that of the existing protocol. There-

fore, some connections rejected by the existing protocol would be accepted by our protocol. That is why the blocking probability of the existing protocol is much higher than that of our protocol when the network load is small. When the network load is large, it is more likely that there is no wavelength in the network to support a newly arrived connection request. Both protocols cannot identify a feasible path in this case. That is why the gap between the blocking probabilities of the existing protocol and our protocol reduces as the network load increases.

We then analyze the performance of our protocol when there are different numbers of wavelengths available on the links. Fig. 8 shows the blocking probabilities when the network load is 150 Erlangs. Fig. 8(a) shows the blocking probabilities with different numbers of wavelengths of the BRITE topology. As the number of wavelengths increases from 8 to 14, the blocking probability of our protocol with  $f = 1.3$  reduces quickly. If each link only has 8 wavelengths, many connections which are determined as feasible by the source are rejected due to unavailability



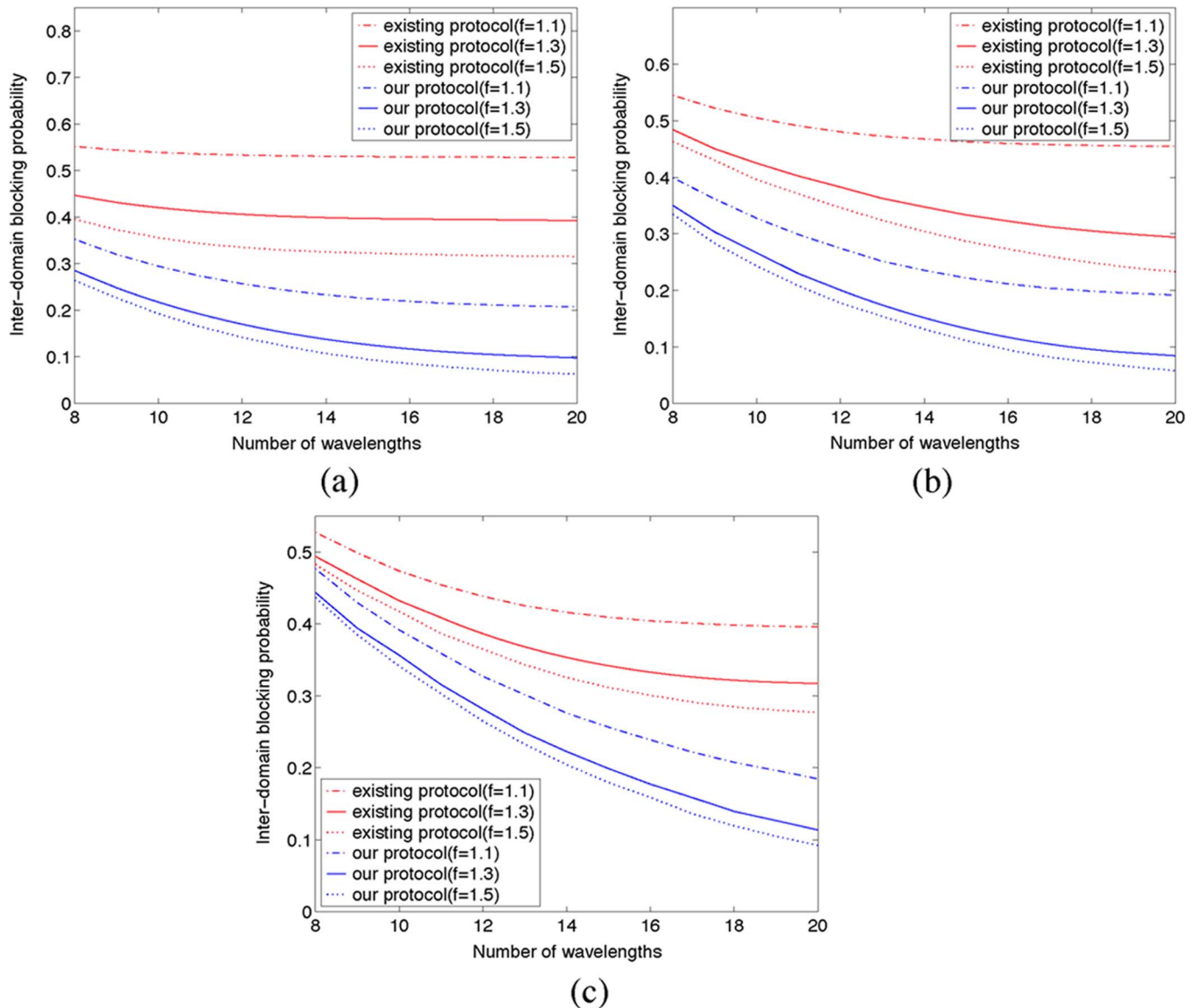


Fig. 8. Performance with varying number of wavelengths when the traffic load is 150 Erlangs. (a) BRITE topology. (b) TOPO1. (c) TOPO2.

of wavelength on the link(s) of the selected path. Therefore, when increasing the wavelength resources, more connections can be accepted, and the blocking probability reduces significantly. When each link has 14 wavelengths, all the connections determined as feasible by the source can almost be supported by the network. In this case, the blocking probability mainly depends on the ratio of the number of infeasible connections (no path satisfies the QoS requirement) to the total number of connections. Under this situation, increasing the number of wavelengths cannot help a lot for reducing the blocking probability. We can observe that the curve for the blocking probability of the existing protocol with  $f = 1.1$  is almost flat. This is because many connections are determined as infeasible by the source. We observe that even each link has only 8 wavelengths, the connections accepted by the source cannot saturate the network. In this case, for improving the blocking probability performance, an efficient algorithm for finding the supported QoS information should be applied. The simulation results for TOPO1 and TOPO2 topologies show the similar characteristics as those in

the BRITE topology. In summary, the simulation results show that our protocol offers significant improvement over the existing protocol in terms of blocking probability in diversified network configurations.

## V. RELATED WORKS

The work in [2] gives the survey for the inter-domain routing in optical networks. It recommends the path-vector based routing model for provisioning the inter-domain routing. This work mentions that the constraint-based routing (providing QoS) is desirable for the multi-domain optical networks. The works in [10], [11] are also extended from the path-vector routing model. Both works focus on designing the framework for inter-domain routing in optical networks but not how to identify a route for a connection request.

A related problem is topology aggregation which studies how to use a simplified topology to represent the original domain. [12] provides a detailed discussion of topology aggregation. The works in [13]–[15] propose different structures for aggregating

a large network. In this work, we apply the full-mesh topology aggregation. That is, each domain topology is aggregated into a small topology which only contains all the border nodes. There is a logical link between any two border nodes in the same domain. For example, in Fig. 1(a), the aggregated topology for domain *A* contains two nodes, *A.1* and *A.2*, and there is a logical link between *A.1* and *A.2*.

The works in [3], [16], [8], [17] consider the routing problem in multi-domain optical networks. The work in [3] just considers one QoS metric (hop count), while the work in [16] considers the hop count and the number of available wavelengths of a path. This means that the work in [16] assumes that each node in the whole network is capable of wavelength conversion, which is not practical. Moreover, the proposed inter-domain routing algorithms in [3], [16] require centralized computation, and cannot be implemented in the path-vector based routing model. The work in [8] is based on the path-vector routing model and considers two additive QoS metrics. However, the path selection of this work just considers one of the QoS constraints imposed by the connection request. The work in [17] considers the source-based inter-domain routing but not the path-vector based routing, so that the proposed mechanism cannot be incorporated in the OBGp framework. This work [17] considers two metrics, delay and the number of available wavelength, and so this work also assumes that each node in the network is capable of converting wavelength. [4] considers the routing with multiple QoS constraints in optical networks. This work applies the flooding-based method to find all the candidate paths between the source and the destination, so as to find the feasible path for a connection request. This work does not consider the wavelength-continuity constraint, and also only considers the routing in a single domain but not in the hierarchical networks.

The *Sparse Switch-Output Conversion* was proposed to reduce costs by limiting the number of wavelength converters at each node. Therefore, the traffic should be routed in a manner that minimizes the use of wavelength converters [3], [18]. After selecting a path for a new connection, each link may have several available wavelengths. An efficient wavelength assignment mechanism is then needed to specify a wavelength on each link to be reserved for the new connection, such that the number of the wavelength converters on this path is minimized. The work in [3] tries all the possible combinations to find the optimal solution. This method requires centralized execution and introduces exponential computational complexity. Our mechanism, on the other hand, is distributed with a polynomial complexity.

## VI. CONCLUSION

This paper discussed several fundamental issues for developing an inter-domain QoS routing protocol in optical networks. We described how to compute the supported QoS between any two border nodes in the same domain and in different domains, and then discussed how to select an inter-domain path for a

given inter-domain connection request with QoS requirement. Our inter-domain routing protocol applies a combination of the fixed-alternate routing policy and the adaptive routing policy to reduce the computational overhead. Moreover, a polynomial wavelength assignment algorithm was proposed to minimize the number of used wavelength converters. Our simulation experiments show that our routing protocol outperforms the existing protocols and is promising for the next generation optical networks.

## REFERENCES

- [1] M. Blanchet, F. Parent, and B. St-Arnaud, "Optical bgp (obgp): Interas lightpath provisioning," [Online]. Available: <http://tools.ietf.org/id/draft-parent-obgp-01.txt>
- [2] M. Chamania and A. Jukan, "A survey of inter-domain peering and provisioning solutions for the next generation optical networks," *IEEE Commun. Surv. Tutorials*, pp. 33–51, First quarter 2009.
- [3] Q. Liu, M. A. Kok, N. Ghani, and A. Gumaste, "Hierarchical interdomain routing and light-path provisioning in optical networks," *J. Opt. Netw.*, vol. 5, no. 10, pp. 764–774, 2006.
- [4] A. Jukan and G. Franzl, "Path selection methods with multiple constraints in service-guaranteed wdm networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 1, pp. 59–72, Feb. 2004.
- [5] R. Hou, K.-S. Lui, K.-C. Leung, and F. Baker, "Routing with qos information aggregation in hierarchical networks," *Proc. IWQoS'09*, pp. 1–9, Jul. 2009.
- [6] R. Hou, K.-S. Lui, K.-C. Leung, and F. Baker, "An approximation algorithm for qos routing with two additive constraints," *Proc. ICNP'08*, pp. 147–156, Oct. 2008.
- [7] H. Zang, J. P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical wdm networks," *SPIE Opt. Netw. Mag.*, vol. 1, no. 1, pp. 47–60, Jan. 2000.
- [8] X. Yang and B. Ramamurthy, "Interdomain dynamic wavelength routing in the next-generation translucent optical internet," *J. Opt. Netw.*, vol. 3, no. 3, pp. 169–187, Mar. 2004.
- [9] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: An approach to universal topology generation," in *Proc. Int. Workshop Modeling, Analysis Simul. Comp. Telecommun. Syst. MASCOTS'01*, Aug. 2001, pp. 346–353.
- [10] O. Yu, "Inter-carrier interdomain control plane for global optical networks," *Proc. IEEE ICC'04*, pp. 1679–1683, Jun. 2004.
- [11] A. Hafid, A. Maach, M. G. Khair, and J. Drissi, "Advance lightpath provisioning in interdomain optical networks," *J. Opt. Netw.*, vol. 4, no. 11, pp. 714–736, Nov. 2005.
- [12] S. Uludag, K. Lui, K. Nahrstedt, and G. Brewster, "Analysis of topology aggregation techniques for QoS routing," *ACM Comput. Surv.*, vol. 39, no. 3, 2007.
- [13] G. Maier, C. Busca, and A. Pattavina, "Multi-domain routing techniques with topology aggregation in ason networks," in *Proc. Int. Conf. Optical Netw. Design Modeling (ONDM)*, Mar. 2005, pp. 1–6.
- [14] L. Lei and Y. Ji, "A spanning tree-based qos aggregation algorithm in hierarchical ason," *IEEE Commun. Lett.*, vol. 9, no. 5, pp. 459–461, May 2005.
- [15] P. Wan, W. Jiao, and X. Wu, "A novel topology aggregation method for hierarchical routing in ason network," *Proc. IEEE Globecom'07*, pp. 2275–2279, Nov. 2007.
- [16] L. Wang, H. Zhang, and X. Zheng, "Inter-domain routing based on simulated annealing algorithm in optical mesh networks," *Opt. Exp.*, vol. 12, no. 4, pp. 3095–2107, Jul. 2004.
- [17] X. Masip, S. Sanchez, J. Sol, J. Domingo, and E. Marfn, "Hierarchical routing with qos constraints in optical transport networks," in *Proc. Netwo.*, May 2004.
- [18] J. Iness and B. Mukherjee, "Sparse wavelength conversion in wavelength-routed wdm optical networks," *Photon. Netw. Commun.*, vol. 1, no. 3, pp. 183–205, Nov. 2004.

**Author biographies not included at authors' request due to space constraints.**