Internet Engineering Task Force (IETF)
Request for Comments: 7926
BCP: 206
Category: Best Current Practice
ISSN: 2070-1721

A. Farrel, Ed.
J. Drake
Juniper Networks
N. Bitar
Nokia
G. Swallow
Cisco Systems, Inc.
D. Ceccarelli
Ericsson
X. Zhang
Huawei
July 2016

Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks

Abstract

In Traffic-Engineered (TE) systems, it is sometimes desirable to establish an end-to-end TE path with a set of constraints (such as bandwidth) across one or more networks from a source to a destination. TE information is the data relating to nodes and TE links that is used in the process of selecting a TE path. TE information is usually only available within a network. We call such a zone of visibility of TE information a domain. An example of a domain may be an IGP area or an Autonomous System.

In order to determine the potential to establish a TE path through a series of connected networks, it is necessary to have available a certain amount of TE information about each network. This need not be the full set of TE information available within each network but does need to express the potential of providing TE connectivity. This subset of TE information is called TE reachability information.

This document sets out the problem statement for the exchange of TE information between interconnected TE networks in support of end-to-end TE path establishment and describes the best current practice architecture to meet this problem statement. For reasons that are explained in this document, this work is limited to simple TE constraints and information that determine TE reachability.

Status of This Memo

This memo documents an Internet Best Current Practice.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on BCPs is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7926.

Copyright Notice

Table of Contents

1. Introduction

Traffic-Engineered (TE) systems such as MPLS-TE [RFC2702] and GMPLS [RFC3945] offer a way to establish paths through a network in a controlled way that reserves network resources on specified links. TE paths are computed by examining the Traffic Engineering Database (TED) and selecting a sequence of links and nodes that are capable of meeting the requirements of the path to be established. The TED is constructed from information distributed by the Interior Gateway Protocol (IGP) running in the network -- for example, OSPF-TE [RFC3630] or ISIS-TE [RFC5305].

It is sometimes desirable to establish an end-to-end TE path that crosses more than one network or administrative domain as described in [RFC4105] and [RFC4216]. In these cases, the availability of TE information is usually limited to within each network. Such networks are often referred to as domains [RFC4726], and we adopt that definition in this document; viz.,

> For the purposes of this document, a domain is considered to be any collection of network elements within a common sphere of address management or path computational responsibility. Examples of such domains include IGP areas and Autonomous Systems (ASes).

In order to determine the potential to establish a TE path through a series of connected domains and to choose the appropriate domain connection points through which to route a path, it is necessary to have available a certain amount of TE information about each domain. This need not be the full set of TE information available within each domain but does need to express the potential of providing TE connectivity. This subset of TE information is called TE reachability information. The TE reachability information can be exchanged between domains based on the information gathered from the local routing protocol, filtered by configured policy, or statically configured.

This document sets out the problem statement for the exchange of TE information between interconnected TE networks in support of end-to-end TE path establishment and describes the best current practice architecture to meet this problem statement. The scope of this document is limited to the simple TE constraints and information (such as TE metrics, hop count, bandwidth, delay, shared risk) necessary to determine TE reachability: discussion of multiple additional constraints that might qualify the reachability can significantly complicate aggregation of information and the stability of the mechanism used to present potential connectivity, as is explained in the body of this document.

Appendix A summarizes relevant existing work that is used to route TE paths across multiple domains.

1.1. Terminology

This section introduces some key terms that need to be understood to arrive at a common understanding of the problem space. Some of the terms are defined in more detail in the sections that follow (in which case forward pointers are provided), and some terms are taken from definitions that already exist in other RFCs (in which case references are given, but no apology is made for repeating or summarizing the definitions here).

1.1.1. TE Paths and TE Connections

A TE connection is a Label Switched Path (LSP) through an MPLS-TE or GMPLS network that directs traffic along a particular path (the TE path) in order to provide a specific service such as bandwidth guarantee, separation of traffic, or resilience between a well-known pair of end points.

1.1.2. TE Metrics and TE Attributes

"TE metrics" and "TE attributes" are terms applied to parameters of links (and possibly nodes) in a network that is traversed by TE connections. The TE metrics and TE attributes are used by path computation algorithms to select the TE paths that the TE connections traverse. A TE metric is a quantifiable value (including measured characteristics) describing some property of a link or node that can be used as part of TE routing or planning, while a TE attribute is a wider term (i.e., including the concept of a TE metric) that refers to any property or characteristic of a link or node that can be used as part of TE routing or planning. Thus, the delay introduced by transmission of a packet on a link is an example of a TE metric, while the geographic location of a router is an example of a more general attribute.

Provisioning a TE connection through a network may result in dynamic changes to the TE metrics and TE attributes of the links and nodes in the network.

These terms are also sometimes used to describe the end-to-end characteristics of a TE connection and can be derived according to a formula from the TE metrics and TE attributes of the links and nodes that the TE connection traverses. Thus, for example, the end-to-end delay for a TE connection is usually considered to be the sum of the delay on each link that the connection traverses.

1.1.3. TE Reachability

In an IP network, reachability is the ability to deliver a packet to a specific address or prefix, i.e., the existence of an IP path to that address or prefix. TE reachability is the ability to reach a specific address along a TE path. More specifically, it is the ability to establish a TE connection in an MPLS-TE or GMPLS sense. Thus, we talk about TE reachability as the potential of providing TE connectivity.

TE reachability may be unqualified (there is a TE path, but no information about available resources or other constraints is supplied); this is helpful especially in determining a path to a destination that lies in an unknown domain or that may be qualified by TE attributes and TE metrics such as hop count, available bandwidth, delay, and shared risk.

1.1.4. Domain

As defined in [RFC4726], a domain is any collection of network elements within a common sphere of address management or path computational responsibility. Examples of such domains include IGP areas and ASes.

1.1.5. Server Network

A Server Network is a network that provides connectivity for another network (the Client Network) in a client-server relationship. A Server Network is sometimes referred to as an underlay network.

1.1.6. Client Network

A Client Network is a network that uses the connectivity provided by a Server Network. A Client Network is sometimes referred to as an overlay network.

1.1.7. Aggregation

The concept of aggregation is discussed in Section 3.5. In aggregation, multiple network resources from a domain are represented outside the domain as a single entity. Thus, multiple links and nodes forming a TE connection may be represented as a single link, or a collection of nodes and links (perhaps the whole domain) may be represented as a single node with its attachment links.

1.1.8. Abstraction

Section 4.2 introduces the concept of abstraction and distinguishes it from aggregation. Abstraction may be viewed as "policy-based aggregation" where the policies are applied to overcome the issues with aggregation as identified in Section 3 of this document.

Abstraction is the process of applying policy to the available TE information within a domain, to produce selective information that represents the potential ability to connect across the domain. Thus, abstraction does not necessarily offer all possible connectivity options, but it presents a general view of potential connectivity according to the policies that determine how the domain’s administrator wants to allow the domain resources to be used.

1.1.9. Abstract Link

An abstract link is the representation of the characteristics of a path between two nodes in a domain produced by abstraction. The abstract link is advertised outside that domain as a TE link for use in signaling in other domains. Thus, an abstract link represents the potential to connect between a pair of nodes.

More details regarding abstract links are provided in Section 4.2.1.

1.1.10. Abstract Node or Virtual Node

An abstract node was defined in [RFC3209] as a group of nodes whose internal topology is opaque to an ingress node of the LSP. More generally, an abstract node is the representation as a single node in a TE topology of some or all of the resources of one or more nodes and the links that connect them. An abstract node may be advertised outside the domain as a TE node for use in path computation and signaling in other domains.

The term "virtual node" has typically been applied to the aggregation of a domain (that is, a collection of nodes and links that operate as a single administrative entity for TE purposes) into a single entity that is treated as a node for the purposes of end-to-end traffic engineering. Virtual nodes are often considered a way to present islands of single-vendor equipment in an optical network.

Sections 3.5 and 4.2.2.1 provide more information about the uses and issues of abstract nodes and virtual nodes.

1.1.11. Abstraction Layer Network

The abstraction layer network is introduced in Section 4.2.2. It may be seen as a brokerage-layer network between one or more server networks and one or more client networks. The abstraction layer network is the collection of abstract links that provide potential connectivity across the server networks and on which path computation can be performed to determine edge-to-edge paths that provide connectivity as links in the client network.

In the simplest case, the abstraction layer network is just a set of edge-to-edge connections (i.e., abstract links), but to make the use of server network resources more flexible, the abstract links might not all extend from edge to edge but might offer connectivity between server network nodes to form a more complex network.

2. Overview of Use Cases

2.1. Peer Networks

The peer network use case can be most simply illustrated by the example in Figure 1. A TE path is required between the source (Src) and destination (Dst), which are located in different domains. There are two points of interconnection between the domains, and selecting the wrong point of interconnection can lead to a suboptimal path or even fail to make a path available. Note that peer networks are assumed to have the same technology type -- that is, the same "switching capability", to use the term from GMPLS [RFC3945].

```
 --------------      --------------
| Domain A     | x1 |     Domain Z |
|   -----      +----+       -----  |
|  | Src |     +----+      | Dst | |
|   -----      | x2 |       -----  |
 --------------      --------------
```

Figure 1: Peer Networks

For example, when Domain A attempts to select a path, it may determine that adequate bandwidth is available from Src through both interconnection points x1 and x2. It may pick the path through x1 for local policy reasons: perhaps the TE metric is smaller. However, if there is no connectivity in Domain Z from x1 to Dst, the path cannot be established. Techniques such as crankback may be used to alleviate this situation, but such techniques do not lead to rapid setup or guaranteed optimality. Furthermore, RSVP signaling creates state in the network that is immediately removed by the crankback procedure. Frequent events of this kind will impact scalability in a non-deterministic manner. More details regarding crankback can be found in Appendix A.2.

There are countless more complicated examples of the problem of peer networks. Figure 2 shows the case where there is a simple mesh of domains. Clearly, to find a TE path from Src to Dst, Domain A must not select a path leaving through interconnect x1, since Domain B has no connectivity to Domain Z. Furthermore, in deciding whether to select interconnection x2 (through Domain C) or interconnection x3 through Domain D, Domain A must be sensitive to the TE connectivity available through each of Domains C and D, as well as the TE connectivity from each of interconnections x4 and x5 to Dst within Domain Z. The problem may be further complicated when the source domain does not know in which domain the destination node is located, since the choice of a domain path clearly depends on the knowledge of the destination domain: this issue is obviously mitigated in IP networks by inter-domain routing [RFC4271].

Of course, many network interconnection scenarios are going to be a combination of the situations expressed in these two examples. There may be a mesh of domains, and the domains may have multiple points of interconnection.

```
                           --------------
                          |     Domain B |
                          |              |
                          |              |
                          /--------------
                         /
                        /x1
         --------------/                       --------------
        | Domain A     |                      |     Domain Z |
        |              |    --------------    |              |
        |  -----       | x2|     Domain C | x4|       -----  |
        | | Src |      +---+              +---+      | Dst | |
        |  -----       |   |              |   |       -----  |
        |              |    --------------    |              |
         --------------\                      /--------------
                        \x3                  /
                         \                  /
                          \                /x5
                           \--------------/
                           |     Domain D |
                           |              |
                           |              |
                            --------------
```

Figure 2: Peer Networks in a Mesh

2.2. Client-Server Networks

Two major classes of use case relate to the client-server relationship between networks. These use cases have sometimes been referred to as overlay networks. In both of these classes of use case, the client and server networks may have the same switching capability, or they may be built from nodes and links that have different technology types in the client and server networks.

The first group of use cases, shown in Figure 3, occurs when domains belonging to one network are connected by a domain belonging to another network. In this scenario, once connectivity is formed across the lower-layer network, the domains of the upper-layer network can be merged into a single domain by running IGP adjacencies and by treating the server-network-layer connectivity as links in the higher-layer network. The TE relationship between the domains (higher and lower layers) in this case is reduced to determining what server network connectivity to establish, how to trigger it, how to route it in the server network, and what resources and capacity to assign within the server network layer. As the demands in the higher-layer (client) network vary, the connectivity in the server network may need to be modified. Section 2.4 explains in a little more detail how connectivity may be requested.

```
       ----------------                          ----------------
      | Client Network |                        | Client Network |
      |   Domain A     |                        |   Domain B     |
      |                |                        |                |
      |  -----         |                        |         -----  |
      | | Src |        |                        |        | Dst | |
      |  -----         |                        |         -----  |
      |                |                        |                |
       ----------------\                        /----------------
                        \x1                  x2/
                         \                    /
                          \                  /
                           \----------------/
                           | Server Network |
                           |     Domain     |
                           |                |
                            ----------------
```

Figure 3: Client-Server Networks

The second class of use case relating to client-server networking is for Virtual Private Networks (VPNs). In this case, as opposed to the former one, it is assumed that the client network has a different address space than that of the server network, where non-overlapping IP addresses between the client and the server networks cannot be guaranteed. A simple example is shown in Figure 4. The VPN sites comprise a set of domains that are interconnected over a core domain (i.e., the provider network) that is the server network in our model.

Note that in the use cases shown in Figures 3 and 4 the client network domains may (and, in fact, probably do) operate as a single connected network.

```
          --------------                         --------------
         | Domain A     |                       |     Domain Z |
         | (VPN site)   |                       |   (VPN site) |
         |              |                       |              |
         |  -----       |                       |       -----  |
         | | Src |      |                       |      | Dst | |
         |  -----       |                       |       -----  |
         |              |                       |              |
          --------------\                       /--------------
                         \x1                 x2/
                          \                   /
                           \                 /
                            \---------------/
                            |  Core Domain  |
                            |               |
                            |               |
                            /---------------\
                           /                 \
                          /                   \
                         /x3                 x4\
          --------------/                       \--------------
         | Domain B     |                       |     Domain C |
         | (VPN site)   |                       |   (VPN site) |
         |              |                       |              |
         |              |                       |              |
          --------------                         --------------
```

Figure 4: A Virtual Private Network

Both use cases in this section become "more interesting" when combined with the use case in Section 2.1 -- that is, when the connectivity between higher-layer domains or VPN sites is provided by a sequence or mesh of lower-layer domains. Figure 5 shows how this might look in the case of a VPN.

```
        ------------                                   ------------
       | Domain A   |                                 |   Domain Z |
       | (VPN site) |                                 | (VPN site) |
       |  -----     |                                 |     -----  |
       | | Src |    |                                 |    | Dst | |
       |  -----     |                                 |     -----  |
       |            |                                 |            |
        ------------\                                 /------------
                     \x1                           x2/
                      \                             /
                       \                           /
                        \----------     ----------/
                        | Domain X |x5 | Domain Y |
                        | (core)   +---+ (core)   |
                        |          |   |          |
                        |          +---+          |
                        |          |x6 |          |
                        /----------     ----------\
                       /                           \
                      /                             \
                     /x3                           x4\
        ------------/                                 \------------
       | Domain B   |                                 |   Domain C |
       | (VPN site) |                                 | (VPN site) |
       |            |                                 |            |
        ------------                                   ------------
```

Figure 5: A VPN Supported over Multiple Server Domains

2.3. Dual-Homing

A further complication may be added to the client-server relationship described in Section 2.2 by considering what happens when a client network domain is attached to more than one domain in the server network or has two points of attachment to a server network domain. Figure 6 shows an example of this for a VPN.

```
                               ------------
                              | Domain B   |
                              | (VPN site) |
       ------------           |  -----     |
      | Domain A   |          | | Src |    |
      | (VPN site) |          |  -----     |
      |            |          |            |
       ------------\           -+--------+-
                    \x1         |        |
                     \        x2|        |x3
                      \         |        |              ------------
                       \--------+-      -+--------     |   Domain C |
                       | Domain X | x8 | Domain Y | x4 | (VPN site) |
                       | (core)   +----+ (core)   +----+     -----  |
                       |          |    |          |    |    | Dst | |
                       |          +----+          +----+     -----  |
                       |          | x9 |          | x5 |            |
                       /----------      ----------\     ------------
                      /                            \
                     /                              \
                    /x6                            x7\
       ------------/                                  \------------
      | Domain D   |                                  |   Domain E |
      | (VPN site) |                                  | (VPN site) |
      |            |                                  |            |
       ------------                                    ------------
```

Figure 6: Dual-Homing in a Virtual Private Network

2.4. Requesting Connectivity

The relationship between domains can be entirely under the control of management processes, dynamically triggered by the client network, or some hybrid of these cases. In the management case, the server network may be asked to establish a set of LSPs to provide client network connectivity. In the dynamic case, the client network may make a request to the server network exerting a range of controls over the paths selected in the server network. This range extends from no control (i.e., a simple request for connectivity), through a set of constraints (latency, path protection, etc.), up to and including full control of the path and resources used in the server network (i.e., the use of explicit paths with label subobjects).

There are various models by which a server network can be asked to set up the connections that support a service provided to the client network. These requests may come from management systems, directly from the client network control plane, or through an intermediary broker such as the Virtual Network Topology Manager (VNTM) [RFC5623].

The trigger that causes the request to the server network is also flexible. It could be that the client network discovers a pressing need for server network resources (such as the desire to provision an end-to-end connection in the client network or severe congestion on a specific path), or it might be that a planning application has considered how best to optimize traffic in the client network or how to handle a predicted traffic demand.

In all cases, the relationship between client and server networks is subject to policy so that server network resources are under the administrative control of the operator or the server network and are only used to support a client network in ways that the server network operator approves.

As just noted, connectivity requests issued to a server network may include varying degrees of constraint upon the choice of path that the server network can implement.

- "Basic provisioning" is a simple request for connectivity. The only constraints are the end points of the connection and the capacity (bandwidth) that the connection will support for the client network. In the case of some server networks, even the bandwidth component of a basic provisioning request is superfluous because the server network has no facility to vary bandwidth and can offer connectivity only at a default capacity.

- "Basic provisioning with optimization" is a service request that indicates one or more metrics that the server network must optimize in its selection of a path. Metrics may be hop count, path length, summed TE metric, jitter, delay, or any number of technology-specific constraints.

- "Basic provisioning with optimization and constraints" enhances the optimization process to apply absolute constraints to functions of the path metrics. For example, a connection may be requested that optimizes for the shortest path but in any case requests that the end-to-end delay be less than a certain value.

Equally, optimization may be expressed in terms of the impact on the network. For example, a service may be requested in order to leave maximal flexibility to satisfy future service requests.

- "Fate diversity requests" ask the server network to provide a path that does not use any network resources (usually links and nodes) that share fate (i.e., can fail as the result of a single event) as the resources used by another connection. This allows the client network to construct protection services over the server network -- for example, by establishing links that are known to be fate diverse. The connections that have diverse paths need not share end points.

- "Provisioning with fate sharing" is the exact opposite of fate diversity. In this case, two or more connections are requested to follow the same path in the server network. This may be requested, for example, to create a bundled or aggregated link in the client network where each component of the client-layer composite link is required to have the same server network properties (metrics, delay, etc.) and the same failure characteristics.

- "Concurrent provisioning" enables the interrelated connection requests described in the previous two bullets to be enacted through a single, compound service request.

- "Service resilience" requests that the server network provide connectivity for which the server network takes responsibility to recover from faults. The resilience may be achieved through the use of link-level protection, segment protection, end-to-end protection, or recovery mechanisms.

2.4.1. Discovering Server Network Information

Although the topology and resource availability information of a server network may be hidden from the client network, the service request interface may support features that report details about the services and potential services that the server network supports.

- Reporting of path details, service parameters, and issues such as path diversity of LSPs that support deployed services allows the client network to understand to what extent its requests were satisfied. This is particularly important when the requests were made as "best effort".

- A server network may support requests of the form "If I were to ask you for this service, would you be able to provide it?" -- that is, a service request that does everything except actually provision the service.

3. Problem Statement

The problem statement presented in this section is as much about the issues that may arise in any solution (and so have to be avoided) and the features that are desirable within a solution, as it is about the actual problem to be solved.

The problem can be stated very simply and with reference to the use cases presented in the previous section.

> A mechanism is required that allows TE path computation in one domain to make informed choices about the TE capabilities and exit points from the domain when signaling an end-to-end TE path that will extend across multiple domains.

Thus, the problem is one of information collection and presentation, not about signaling. Indeed, the existing signaling mechanisms for TE LSP establishment are likely to prove adequate [RFC4726] with the possibility of minor extensions. Similarly, TE information may currently be distributed in a domain by TE extensions to one of the two IGPs as described in OSPF-TE [RFC3630] and ISIS-TE [RFC5305], and TE information may be exported from a domain (for example, northbound) using link-state extensions to BGP [RFC7752].

An interesting annex to the problem is how the path is made available for use. For example, in the case of a client-server network, the path established in the server network needs to be made available as a TE link to provide connectivity in the client network.

3.1. Policy and Filters

A solution must be amenable to the application of policy and filters. That is, the operator of a domain that is sharing information with another domain must be able to apply controls to what information is shared. Furthermore, the operator of a domain that has information shared with it must be able to apply policies and filters to the received information.

Additionally, the path computation within a domain must be able to weight the information received from other domains according to local policy such that the resultant computed path meets the local operator’s needs and policies rather than those of the operators of other domains.

3.2. Confidentiality

A feature of the policy described in Section 3.1 is that an operator of a domain may desire to keep confidential the details about its internal network topology and loading. This information could be construed as commercially sensitive.

Although it is possible that TE information exchange will take place only between parties that have significant trust, there are also use cases (such as the VPN supported over multiple server network domains described in Section 2.2) where information will be shared between domains that have a commercial relationship but a low level of trust.

Thus, it must be possible for a domain to limit the shared information to only that which the computing domain needs to know, with the understanding that the less information that is made available the more likely it is that the result will be a less optimal path and/or more crankback events.

3.3. Information Overload

One reason that networks are partitioned into separate domains is to reduce the set of information that any one router has to handle. This also applies to the volume of information that routing protocols have to distribute.

Over the years, routers have become more sophisticated, with greater processing capabilities and more storage; the control channels on which routing messages are exchanged have become higher capacity; and the routing protocols (and their implementations) have become more robust. Thus, some of the arguments in favor of dividing a network into domains may have been reduced. Conversely, however, the size of networks continues to grow dramatically with a consequent increase in the total amount of routing-related information available. Additionally, in this case, the problem space spans two or more networks.

Any solution to the problems voiced in this document must be aware of the issues of information overload. If the solution was to simply share all TE information between all domains in the network, the effect from the point of view of the information load would be to create one single flat network domain. Thus, the solution must deliver enough information to make the computation practical (i.e., to solve the problem) but not so much as to overload the receiving domain. Furthermore, the solution cannot simply rely on the policies and filters described in Section 3.1 because such filters might not always be enabled.

3.4. Issues of Information Churn

As LSPs are set up and torn down, the available TE resources on links in the network change. In order to reliably compute a TE path through a network, the computation point must have an up-to-date view of the available TE resources. However, collecting this information may result in considerable load on the distribution protocol and churn in the stored information. In order to deal with this problem even in a single domain, updates are sent at periodic intervals or whenever there is a significant change in resources, whichever happens first.

Consider, for example, that a TE LSP may traverse ten links in a network. When the LSP is set up or torn down, the resources available on each link will change, resulting in a new advertisement of the link’s capabilities and capacity. If the arrival rate of new LSPs is relatively fast, and the hold times relatively short, the network may be in a constant state of flux. Note that the problem here is not limited to churn within a single domain, since the information shared between domains will also be changing. Furthermore, the information that one domain needs to share with another may change as the result of LSPs that are contained within or cross the first domain but that are of no direct relevance to the domain receiving the TE information.

In packet networks, where the capacity of an LSP is often a small fraction of the resources available on any link, this issue is partially addressed by the advertising routers. They can apply a threshold so that they do not bother to update the advertisement of available resources on a link if the change is less than a configured percentage of the total (or, alternatively, the remaining) resources. The updated information in that case will be disseminated based on an update interval rather than a resource change event.

In non-packet networks, where link resources are physical switching resources (such as timeslots or wavelengths), the capacity of an LSP may more frequently be a significant percentage of the available link resources. Furthermore, in some switching environments, it is necessary to achieve end-to-end resource continuity (such as using the same wavelength on the whole length of an LSP), so it is far more desirable to keep the TE information held at the computation points up to date. Fortunately, non-packet networks tend to be quite a bit smaller than packet networks, the arrival rates of non-packet LSPs are much lower, and the hold times are considerably longer. Thus, the information churn may be sustainable.

3.5. Issues of Aggregation

One possible solution to the issues raised in other subsections of this section is to aggregate the TE information shared between domains. Two aggregation mechanisms are often considered:

- Virtual node model. In this view, the domain is aggregated as if it was a single node (or router/switch). Its links to other domains are presented as real TE links, but the model assumes that any LSP entering the virtual node through a link can be routed to leave the virtual node through any other link (although recent work on "limited cross-connect switches" may help with this problem [RFC7579]).

- Virtual link model. In this model, the domain is reduced to a set of edge-to-edge TE links. Thus, when computing a path for an LSP that crosses the domain, a computation point can see which domain entry points can be connected to which others, and with what TE attributes.

Part of the nature of aggregation is that information is removed from the system. This can cause inaccuracies and failed path computation. For example, in the virtual node model there might not actually be a TE path available between a pair of domain entry points, but the model lacks the sophistication to represent this "limited cross-connect capability" within the virtual node. On the other hand, in the virtual link model it may prove very hard to aggregate multiple link characteristics: for example, there may be one path available with high bandwidth, and another with low delay, but this does not mean that the connectivity should be assumed or advertised as having both high bandwidth and low delay.

The trick to this multidimensional problem, therefore, is to aggregate in a way that retains as much useful information as possible while removing the data that is not needed. An important part of this trick is a clear understanding of what information is actually needed.

It should also be noted in the context of Section 3.4 that changes in the information within a domain may have a bearing on what aggregated data is shared with another domain. Thus, while the data shared is reduced, the aggregation algorithm (operating on the routers responsible for sharing information) may be heavily exercised.

4. Architecture

4.1. TE Reachability

As described in Section 1.1, TE reachability is the ability to reach a specific address along a TE path. The knowledge of TE reachability enables an end-to-end TE path to be computed.

In a single network, TE reachability is derived from the Traffic Engineering Database (TED), which is the collection of all TE information about all TE links in the network. The TED is usually built from the data exchanged by the IGP, although it can be supplemented by configuration and inventory details, especially in transport networks.

In multi-network scenarios, TE reachability information can be described as "You can get from node X to node Y with the following TE attributes." For transit cases, nodes X and Y will be edge nodes of the transit network, but it is also important to consider the information about the TE connectivity between an edge node and a specific destination node. TE reachability may be qualified by TE attributes such as TE metrics, hop count, available bandwidth, delay, and shared risk.

TE reachability information can be exchanged between networks so that nodes in one network can determine whether they can establish TE paths across or into another network. Such exchanges are subject to a range of policies imposed by the advertiser (for security and administrative control) and by the receiver (for scalability and stability).

4.2. Abstraction, Not Aggregation

Aggregation is the process of synthesizing from available information. Thus, the virtual node and virtual link models described in Section 3.5 rely on processing the information available within a network to produce the aggregate representations of links and nodes that are presented to the consumer. As described in Section 3, dynamic aggregation is subject to a number of pitfalls.

In order to distinguish the architecture described in this document from the previous work on aggregation, we use the term "abstraction" in this document. The process of abstraction is one of applying policy to the available TE information within a domain, to produce selective information that represents the potential ability to connect across the domain.

Abstraction does not offer all possible connectivity options (refer to Section 3.5) but does present a general view of potential connectivity. Abstraction may have a dynamic element but is not intended to keep pace with the changes in TE attribute availability within the network.

Thus, when relying on an abstraction to compute an end-to-end path, the process might not deliver a usable path. That is, there is no actual guarantee that the abstractions are current or feasible.

Although abstraction uses available TE information, it is subject to policy and management choices. Thus, not all potential connectivity will be advertised to each client network. The filters may depend on commercial relationships, the risk of disclosing confidential information, and concerns about what use is made of the connectivity that is offered.

4.2.1. Abstract Links

An abstract link is a measure of the potential to connect a pair of points with certain TE parameters. That is, it is a path and its characteristics in the server network. An abstract link represents the possibility of setting up an LSP, and LSPs may be set up over the abstract link.

When looking at a network such as the network shown in Figure 7, the link from CN1 to CN4 may be an abstract link. It is easy to advertise it as a link by abstracting the TE information in the server network, subject to policy.

The path (i.e., the abstract link) represents the possibility of establishing an LSP from client network edge to client network edge across the server network. There is not necessarily a one-to-one relationship between the abstract link and the LSP, because more than one LSP could be set up over the path.

Since the client network nodes do not have visibility into the server network, they must rely on abstraction information delivered to them by the server network. That is, the server network will report on the potential for connectivity.

4.2.2. The Abstraction Layer Network

Figure 7 introduces the abstraction layer network. This construct separates the client network resources (nodes C1, C2, C3, and C4, and the corresponding links) and the server network resources (nodes CN1, CN2, CN3, and CN4, and the corresponding links). Additionally, the architecture introduces an intermediary network layer called the abstraction layer. The abstraction layer contains the client network edge nodes (C2 and C3), the server network edge nodes (CN1 and CN4), the client-server links (C2-CN1 and CN4-C3), and the abstract link (CN1-CN4).

The client network is able to operate as normal. Connectivity across the network can be either found or not found, based on links that appear in the client network TED. If connectivity cannot be found, end-to-end LSPs cannot be set up. This failure may be reported, but no dynamic action is taken by the client network.

The server network also operates as normal. LSPs across the server network between client network edges are set up in response to management commands or in response to signaling requests.

The abstraction layer consists of the physical links between the two networks, and also the abstract links. The abstract links are created by the server network according to local policy and represent the potential connectivity that could be created across the server network and that the server network is willing to make available for use by the client network. Thus, in this example, the diameter of the abstraction layer network is only three hops, but an instance of an IGP could easily be run so that all nodes participating in the abstraction layer (and, in particular, the client network edge nodes) can see the TE connectivity in the layer.

```
    --    --                                  --    --
   |C1|--|C2|                                |C3|--|C4|   Client Network
    --   |  |                                |  |   --
         |  |                                |  |  . . . . . . . . . . .
         |  |                                |  |
         |  |                                |  |
         |  |    ---                  ---    |  |          Abstraction
         |  |---|CN1|================|CN4|---|  |         Layer Network
          --    |   |                |   |    --
                |   |                |   |   . . . . . . . . . . . . . .
                |   |                |   |
                |   |                |   |
                |   |   ---    ---   |   |                Server Network
                |   |--|CN2|--|CN3|--|   |
                 ---    ---    ---    ---

    Key
    --- Direct connection between two nodes
    === Abstract link
```

Figure 7: Architecture for Abstraction Layer Network

When the client network needs additional connectivity, it can make a request to the abstraction layer network. For example, the operator of the client network may want to create a link from C2 to C3. The abstraction layer can see the potential path C2-CN1-CN4-C3 and can set up an LSP C2-CN1-CN4-C3 across the server network and make the LSP available as a link in the client network.

Sections 4.2.3 and 4.2.4 show how this model is used to satisfy the requirements for connectivity in client-server networks and in peer networks.

4.2.2.1. Nodes in the Abstraction Layer Network

Figure 7 shows a very simplified network diagram, and the reader would be forgiven for thinking that only client network edge nodes and server network edge nodes may appear in the abstraction layer network. But this is not the case: other nodes from the server network may be present. This allows the abstraction layer network to be more complex than a full mesh with access spokes.

Thus, as shown in Figure 8, a transit node in the server network (here, the node is CN3) can be exposed as a node in the abstraction layer network with abstract links connecting it to other nodes in the abstraction layer network. Of course, in the network shown in Figure 8, there is little if any value in exposing CN3, but if it had other abstract links to other nodes in the abstraction layer network and/or direct connections to client network nodes, then the resulting network would be richer.

```
    --    --                                     --    --     Client
   |C1|--|C2|                                   |C3|--|C4|    Network
    --   |  |                                   |  |   --
         |  |                                   |  |  . . . . . . . . .
         |  |                                   |  |
         |  |                                   |  |
         |  |   ---          ---          ---   |  |       Abstraction
         |  |--|CN1|========|CN3|========|CN5|--|  |      Layer Network
          --   |   |        |   |        |   |   --
               |   |        |   |        |   |  . . . . . . . . . . . .
               |   |        |   |        |   |
               |   |        |   |        |   |                 Server
               |   |   ---  |   |  ---   |   |                 Network
               |   |--|CN2|-|   |-|CN4|--|   |
                ---    ---   ---   ---    ---
```

Figure 8: Abstraction Layer Network with Additional Node

It should be noted that the nodes included in the abstraction layer network in this way are not "abstract nodes" in the sense of a virtual node described in Section 3.5. Although it is the case that the policy point responsible for advertising server network resources into the abstraction layer network could choose to advertise abstract nodes in place of real physical nodes, it is believed that doing so would introduce significant complexity in terms of:

- Coordination between all of the external interfaces of the abstract node.

- Management of changes in the server network that lead to limited capabilities to reach (cross-connect) across the abstract node. There has been recent work on control-plane extensions to describe and operate devices (such as asymmetrical switches) that have limited cross-connect capabilities [RFC7579] [RFC7580]. These or similar extensions could be used to represent the same type of limitations, as they also apply in an abstract node.

4.2.3. Abstraction in Client-Server Networks

Figure 9 shows the basic architectural concepts for a client-server network. The nodes in the client network are C1, C2, CE1, CE2, C3, and C4, where the client edge (CE) nodes are CE1 and CE2. The core (server) network nodes are CN1, CN2, CN3, and CN4. The interfaces CE1-CN1 and CE2-CN4 are the interfaces between the client and server networks.

The technologies (switching capabilities) of the client and server networks may be the same or different. If they are different, the client network traffic must be tunneled over a server network LSP. If they are the same, the client network LSP may be routed over the server network links, tunneled over a server network LSP, or constructed from the concatenation (stitching) of client network and server network LSP segments.

```
                      :                            :
      Client Network  :       Server Network       :  Client Network
                      :                            :
     --    --    ---                                  ---    --    --
    |C1|--|C2|--|CE1|................................|CE2|--|C3|--|C4|
     --    --   |   |    ---                 ---     |   |   --    --
                |   |===|CN1|================|CN4|===|   |
                |   |---|   |                |   |---|   |
                 ---    |   |   ---    ---   |   |    ---
                        |   |--|CN2|--|CN3|--|   |
                         ---    ---    ---    ---

     Key
     --- Direct connection between two nodes
     ... CE-to-CE LSP tunnel
     === Potential path across the server network (abstract link)
```

Figure 9: Architecture for Client-Server Network

The objective is to be able to support an end-to-end connection, C1-to-C4, in the client network. This connection may support TE or normal IP forwarding. To achieve this, CE1 is to be connected to CE2 by a link in the client network. This enables the client network to view itself as connected and to select an end-to-end path.

As shown in the figure, three abstraction layer links are formed: CE1-CN1, CN1-CN2, and CN4-CE2. A three-hop LSP is then established from CE1 to CE2 that can be presented as a link in the client network.

The practicalities of how the CE1-CE2 LSP is carried across the server network LSP may depend on the switching and signaling options available in the server network. The CE1-CE2 LSP may be tunneled down the server network LSP using the mechanisms of a hierarchical LSP [RFC4206], or the LSP segments CE1-CN1 and CN4-CE2 may be stitched to the server network LSP as described in [RFC5150].

Section 4.2.2 has already introduced the concept of the abstraction layer network through an example of a simple layered network. But it may be helpful to expand on the example using a slightly more complex network.

Figure 10 shows a multi-layer network comprising client network nodes (labeled as Cn for n = 0 to 9) and server network nodes (labeled as Sn for n = 1 to 9).

```
                                              --      --
                                             |C3|---|C4|
                                             /--      --\
             --     --     --     --      --/           \--
            |C1|---|C2|---|S1|---|S2|----|S3|           |C5|
             --    /--     --\    --\     --\           /--
                  /           \--    \--     \--     --/    --
                 /            |S4|   |S5|----|S6|---|C6|---|C7|
                /             /--     --\    /--    /--     --
             --/    --     --/    --     \--/    --/
            |C8|---|C9|---|S7|---|S8|----|S9|---|C0|
             --     --     --     --      --     --
```

Figure 10: An Example Multi-Layer Network

If the network in Figure 10 is operated as separate client and server networks, then the client network topology will appear as shown in Figure 11. As can be clearly seen, the network is partitioned, and there is no way to set up an LSP from a node on the left-hand side (say C1) to a node on the right-hand side (say C7).

```
                                    --     --
                                   |C3|---|C4|
                                    --     --\
                    --     --                 \--
                   |C1|---|C2|                |C5|
                    --    /--                 /--
                         /                 --/    --
                        /                 |C6|---|C7|
                       /                  /--     --
                    --/    --          --/
                   |C8|---|C9|        |C0|
                    --     --          --
```

Figure 11: Client Network Topology Showing Partitioned Network

For reference, Figure 12 shows the corresponding server network topology.

```
                          --     --      --
                         |S1|---|S2|----|S3|
                          --\    --\     --\
                             \--    \--     \--
                             |S4|   |S5|----|S6|
                             /--     --\    /--
                          --/    --     \--/
                         |S7|---|S8|----|S9|
                          --     --      --
```

Figure 12: Server Network Topology

Operating on the TED for the server network, a management entity or a software component may apply policy and consider what abstract links it might offer for use by the client network. To do this, it obviously needs to be aware of the connections between the layers (there is no point in offering an abstract link S2-S8, since this could not be of any use in this example).

In our example, after consideration of which LSPs could be set up in the server network, four abstract links are offered: S1-S3, S3-S6, S1-S9, and S7-S9. These abstract links are shown as double lines on the resulting topology of the abstraction layer network in Figure 13. As can be seen, two of the links must share part of a path (S1-S9 must share with either S1-S3 or S7-S9). This could be achieved using distinct resources (for example, separate lambdas) where the paths are common, but it could also be done using resource sharing.

```
                                            --
                                           |C3|
                                           /--
                   --     --            --/
                  |C2|---|S1|==========|S3|
                   --     --\\          --\\
                             \\            \\
                              \\            \\--     --
                               \\            |S6|---|C6|
                                \\            --     --
                   --     --     \\--     --
                  |C9|---|S7|=====|S9|---|C0|
                   --     --       --     --
```

Figure 13: Abstraction Layer Network with Abstract Links

That would mean that when both paths S1-S3 and S7-S9 carry client-edge-to-client-edge LSPs, the resources on path S1-S9 are used and might be depleted to the point that the path is resource constrained and cannot be used.

The separate IGP instance running in the abstraction layer network means that this topology is visible at the edge nodes (C2, C3, C6, C9, and C0) as well as at a Path Computation Element (PCE) if one is present.

Now the client network is able to make requests to the abstraction layer network to provide connectivity. In our example, it requests that C2 be connected to C3 and that C2 be connected to C0. This results in several actions:

1. The management component for the abstraction layer network asks its PCE to compute the paths necessary to make the connections. This yields C2-S1-S3-C3 and C2-S1-S9-C0.

2. The management component for the abstraction layer network instructs C2 to start the signaling process for the new LSPs in the abstraction layer.

3. C2 signals the LSPs for setup using the explicit routes C2-S1-S3-C3 and C2-S1-S9-C0.

4. When the signaling messages reach S1 (in our example, both LSPs traverse S1), the server network may support them by a number of means, including establishing server network LSPs as tunnels, depending on the mismatch of technologies between the client and server networks. For example, S1-S2-S3 and S1-S2-S5-S9 might be traversed via an LSP tunnel, using LSPs stitched together, or simply by routing the client network LSP through the server network. If server network LSPs are needed, they can be signaled at this point.

5. Once any server network LSPs that are needed have been established, S1 can continue to signal the client-edge-to-client-edge LSP across the abstraction layer, using the server network LSPs as either tunnels or stitching segments, or simply routing through the server network.

6. Finally, once the client-edge-to-client-edge LSPs have been set up, the client network can be informed and can start to advertise the new TE links C2-C3 and C2-C0. The resulting client network topology is shown in Figure 14.

```
                                      --   --
                                     |C3|-|C4|
                                     /--   --\
                                    /         \--
                          --     --/          |C5|
                         |C1|---|C2|          /--
                          --    /--\       --/    --
                               /    \     |C6|---|C7|
                              /      \    /--     --
                             /        \--/
                          --/    --   |C0|
                         |C8|---|C9|   --
                          --     --
```

Figure 14: Connected Client Network with Additional Links

7. Now the client network can compute an end-to-end path from C1 to C7.

4.2.3.1. A Server with Multiple Clients

A single server network may support multiple client networks. This is not an uncommon state of affairs -- for example, when the server network provides connectivity for multiple customers.

In this case, the abstraction provided by the server network may vary considerably according to the policies and commercial relationships with each customer. This variance would lead to a separate abstraction layer network maintained to support each client network.

On the other hand, it may be that multiple client networks are subject to the same policies and the abstraction can be identical. In this case, a single abstraction layer network can support more than one client.

The choices here are made as an operational issue by the server network.

4.2.3.2. A Client with Multiple Servers

A single client network may be supported by multiple server networks. The server networks may provide connectivity between different parts of the client network or may provide parallel (redundant) connectivity for the client network.

In this case, the abstraction layer network should contain the abstract links from all server networks so that it can make suitable computations and create the correct TE links in the client network.

That is, the relationship between the client network and the abstraction layer network should be one to one.

4.2.4. Abstraction in Peer Networks

Figure 15 shows the basic architectural concepts for connecting across peer networks. Nodes from four networks are shown: A1 and A2 come from one network; B1, B2, and B3 from another network; etc. The interfaces between the networks (sometimes known as External Network Network Interfaces - ENNIs) are A2-B1, B3-C1, and C3-D1.

The objective is to be able to support an end-to-end connection, A1-to-D2. This connection is for TE connectivity.

As shown in the figure, abstract links that span the transit networks are used to achieve the required connectivity. These links form the key building blocks of the end-to-end connectivity. An end-to-end LSP uses these links as part of its path. If the stitching capabilities of the networks are homogeneous, then the end-to-end LSP may simply traverse the path defined by the abstract links across the various peer networks or may utilize stitching of LSP segments that each traverse a network along the path of an abstract link. If the network switching technologies support or necessitate the use of LSP hierarchies, the end-to-end LSP may be tunneled across each network using hierarchical LSPs that each traverse a network along the path of an abstract link.

```
                 :                  :                  :
      Network A  :    Network B     :    Network C     :  Network D
                 :                  :                  :
       --    --     --    --    --     --    --    --     --    --
      |A1|--|A2|---|B1|--|B2|--|B3|---|C1|--|C2|--|C3|---|D1|--|D2|
       --    --    |  |   --   |  |   |  |   --   |  |    --    --
                   |  |========|  |   |  |========|  |
                    --          --     --          --

      Key
      --- Direct connection between two nodes
      === Abstract link across transit network
```

Figure 15: Architecture for Peering

Peer networks exist in many situations in the Internet. Packet networks may peer as IGP areas (levels) or as ASes. Transport networks (such as optical networks) may peer to provide concatenations of optical paths through single-vendor environments (see Section 6). Figure 16 shows a simple example of three peer networks (A, B, and C) each comprising a few nodes.

```
                 Network A    :     Network B      :   Network C
                              :                    :
           --     --      --  :  --     --     --  :  --     --
          |A1|---|A2|----|A3|---|B1|---|B2|---|B3|---|C1|---|C2|
           --     --\    /--  :  --    /--\    --  :  --     --
                     \--/     :       /    \       :
                     |A4|     :      /      \      :
                      --\     :     /        \     :
                   --    \--  :  --/          \--  :  --     --
                  |A5|---|A6|---|B4|----------|B6|---|C3|---|C4|
                   --     --  :  --            --  :  --     --
                              :                    :
                              :                    :
```

Figure 16: A Network Comprising Three Peer Networks

As discussed in Section 2, peered networks do not share visibility of their topologies or TE capabilities for scaling and confidentiality reasons. That means, in our example, that computing a path from A1 to C4 can be impossible without the aid of cooperating PCEs or some form of crankback.

But it is possible to produce abstract links for reachability across transit peer networks and to create an abstraction layer network. That network can be enhanced with specific reachability information if a destination network is partitioned, as is the case with Network C in Figure 16.

Suppose that Network B decides to offer three abstract links B1-B3, B4-B3, and B4-B6. The abstraction layer network could then be constructed to look like the network in Figure 17.

```
                        --     --      --      --
                       |A3|---|B1|====|B3|----|C1|
                        --     --    //--      --
                                    //
                                   //
                                  //
                        --     --//     --     --
                       |A6|---|B4|=====|B6|---|C3|
                        --     --       --     --
```

Figure 17: Abstraction Layer Network for the Peer Network Example

Using a process similar to that described in Section 4.2.3, Network A can request connectivity to Network C, and abstract links can be advertised that connect the edges of the two networks and that can be used to carry LSPs that traverse both networks. Furthermore, if

Network C is partitioned, reachability information can be exchanged to allow Network A to select the correct abstract link, as shown in Figure 18.

```
                       Network A       :      Network C
                                       :
                 --     --      --     :     --       --
                |A1|---|A2|----|A3|=========|C1|.....|C2|
                 --     --\    /--     :     --       --
                           \--/        :
                           |A4|        :
                            --\        :
                         --    \--     :     --       --
                        |A5|---|A6|=========|C3|.....|C4|
                         --     --     :     --       --
```

Figure 18: Tunnel Connections to Network C with TE Reachability

Peer networking cases can be made far more complex by dual-homing between network peering nodes (for example, A3 might connect to B1 and B4 in Figure 17) and by the networks themselves being arranged in a mesh (for example, A6 might connect to B4 and C1 in Figure 17).

These additional complexities can be handled gracefully by the abstraction layer network model.

Further examples of abstraction in peer networks can be found in Sections 6 and 8.

4.3. Considerations for Dynamic Abstraction

It is possible to consider a highly dynamic system where the server network adaptively suggests new abstract links into the abstraction layer, and where the abstraction layer proactively deploys new client-edge-to-client-edge LSPs to provide new links in the client network. Such fluidity is, however, to be treated with caution. In particular, in the case of client-server networks of differing technologies where hierarchical server network LSPs are used, this caution is needed for three reasons: there may be longer turn-up times for connections in some server networks; the server networks are likely to be sparsely connected; and expensive physical resources will only be deployed where there is believed to be a need for them. More significantly, the complex commercial, policy, and administrative relationships that may exist between client and server network operators mean that stability is more likely to be the desired operational practice.

Thus, proposals for fully automated multi-layer networks based on this architecture may be regarded as forward-looking topics for research both in terms of network stability and with regard to economic impact.

However, some elements of automation should not be discarded. A server network may automatically apply policy to determine the best set of abstract links to offer and the most suitable way for the server network to support them. And a client network may dynamically observe congestion, lack of connectivity, or predicted changes in traffic demand and may use this information to request additional links from the abstraction layer. And, once policies have been configured, the whole system should be able to operate independently of operator control (which is not to say that the operator will not have the option of exerting control at every step in the process).

4.4. Requirements for Advertising Links and Nodes

The abstraction layer network is "just another network layer". The links and nodes in the network need to be advertised along with their associated TE information (metrics, bandwidth, etc.) so that the topology is disseminated and so that routing decisions can be made.

This requires a routing protocol running between the nodes in the abstraction layer network. Note that this routing information exchange could be piggybacked on an existing routing protocol instance (subject to different switching capabilities applying to the links in the different networks, or to adequate address space separation) or use a new instance (or even a new protocol). Clearly, the information exchanged is only information that has been created as part of the abstraction function according to policy.

It should be noted that in many cases the abstract link represents the potential for connectivity across the server network but that no such connectivity exists. In this case, we may ponder how the routing protocol in the abstraction layer will advertise topology information for, and over, a link that has no underlying connectivity. In other words, there must be a communication channel between the abstraction layer nodes so that the routing protocol messages can flow. The answer is that control-plane connectivity already exists in the server network and on the client-server edge links, and this can be used to carry the routing protocol messages for the abstraction layer network. The same consideration applies to the advertisement, in the client network, of the potential connectivity that the abstraction layer network can provide, although it may be more normal to establish that connectivity before advertising a link in the client network.

4.5. Addressing Considerations

The network layers in this architecture should be able to operate with separate address spaces, and these may overlap without any technical issues. That is, one address may mean one thing in the client network, yet the same address may have a different meaning in the abstraction layer network or the server network. In other words, there is complete address separation between networks.

However, this will require some care, both because human operators may well become confused, and because mapping between address spaces is needed at the interfaces between the network layers. That mapping requires configuration so that, for example, when the server network announces an abstract link from A to B, the abstraction layer network must recognize that A and B are server network addresses and must map them to abstraction layer addresses (say P and Q) before including the link in its own topology. And similarly, when the abstraction layer network informs the client network that a new link is available from S to T, it must map those addresses from its own address space to that of the client network.

This form of address mapping will become particularly important in cases where one abstraction layer network is constructed from connectivity in multiple server networks, or where one abstraction layer network provides connectivity for multiple client networks.

5. Building on Existing Protocols

This section is non-normative and is not intended to prejudge a solutions framework or any applicability work. It does, however, very briefly serve to note the existence of protocols that could be examined for applicability to serve in realizing the model described in this document.

The general principle of protocol reuse is preferred over the invention of new protocols or additional protocol extensions, and it would be advantageous to make use of an existing protocol that is commonly implemented on network nodes and is currently deployed, or to use existing computational elements such as PCEs. This has many benefits in network stability, time to deployment, and operator training.

It is recognized, however, that existing protocols are unlikely to be immediately suitable to this problem space without some protocol extensions. Extending protocols must be done with care and with consideration for the stability of existing deployments. In extreme cases, a new protocol can be preferable to a messy hack of an existing protocol.

5.1. BGP-LS

BGP - Link State (BGP-LS) is a set of extensions to BGP, as described in [RFC7752]. Its purpose is to announce topology information from one network to a "northbound" consumer. Application of BGP-LS to date has focused on a mechanism to build a TED for a PCE. However, BGP’s mechanisms would also serve well to advertise abstract links from a server network into the abstraction layer network or to advertise potential connectivity from the abstraction layer network to the client network.

5.2. IGPs

Both OSPF and IS-IS have been extended through a number of RFCs to advertise TE information. Additionally, both protocols are capable of running in a multi-instance mode either as ships that pass in the night (i.e., completely separate instances using different address spaces) or as dual instances on the same address space. This means that either OSPF or IS-IS could probably be used as the routing protocol in the abstraction layer network.

5.3. RSVP-TE

RSVP-TE signaling can be used to set up all TE LSPs demanded by this model, without the need for any protocol extensions.

If necessary, LSP hierarchy [RFC4206] or LSP stitching [RFC5150] can be used to carry LSPs over the server network, again without needing any protocol extensions.

Furthermore, the procedures in [RFC6107] allow the dynamic signaling of the purpose of any LSP that is established. This means that when an LSP tunnel is set up, the two ends can coordinate into which routing protocol instance it should be advertised and can also agree on the addressing to be said to identify the link that will be created.

5.4. Notes on a Solution

This section is not intended to be prescriptive or dictate the protocol solutions that may be used to satisfy the architecture described in this document, but it does show how the existing protocols listed in the previous sections can be combined, with only minor modifications, to provide a solution.

A server network can be operated using GMPLS routing and signaling protocols. Using information gathered from the routing protocol, a TED can be constructed containing resource availability information and Shared Risk Link Group (SRLG) details. A policy-based process can then determine which nodes and abstract links it wishes to advertise to form the abstraction layer network.

The server network can now use BGP-LS to advertise a topology of links and nodes to form the abstraction layer network. This information would most likely be advertised from a single point of control that made all of the abstraction decisions, but the function could be distributed to multiple server network edge nodes. The information can be advertised by BGP-LS to multiple points within the abstraction layer (such as all client network edge nodes) or to a single controller.

Multiple server networks may advertise information that is used to construct an abstraction layer network, and one server network may advertise different information in different instances of BGP-LS to form different abstraction layer networks. Furthermore, in the case of one controller constructing multiple abstraction layer networks, BGP-LS uses the route target mechanism defined in [RFC4364] to distinguish the different applications (effectively abstraction layer network VPNs) of the exported information.

Extensions may be made to BGP-LS to allow advertisement of Macro Shared Risk Link Groups (MSRLGs) (Appendix B.1) and the identification of mutually exclusive links (Appendix B.2), and to indicate whether the abstract link has been pre-established or not. Such extensions are valid options but do not form a core component of this architecture.

The abstraction layer network may operate under central control or use a distributed control plane. Since the links and nodes may be a mix of physical and abstract links, and since the nodes may have diverse cross-connect capabilities, it is most likely that a GMPLS routing protocol will be beneficial for collecting and correlating the routing information and for distributing updates. No special additional features are needed beyond adding those extra parameters just described for BGP-LS, but it should be noted that the control plane of the abstraction layer network must run in an out-of-band control network because the data-bearing links might not yet have been established via connections in the server network.

The abstraction layer network is also able to determine potential connectivity from client network edge to client network edge. It will determine which client network links to create according to policy and subject to requests from the client network, and will take four steps:

- First, it will compute a path across the abstraction layer network.

- Then, if support of the abstract links requires the use of server network LSPs for tunneling or stitching and if those LSPs are not already established, it will ask the server layer to set them up.

- Then, it will signal the client-edge-to-client-edge LSP.

- Finally, the abstraction layer network will inform the client network of the existence of the new client network link.

This last step can be achieved by either (1) coordination of the end points of the LSPs that span the abstraction layer (these points are client network edge nodes) using mechanisms such as those described in [RFC6107] or (2) using BGP-LS from a central controller.

Once the client network edge nodes are aware of a new link, they will automatically advertise it using their routing protocol and it will become available for use by traffic in the client network.

Sections 6, 7, and 8 discuss the applicability of this architecture to different network types and problem spaces, while Section 9 gives some advice about scoping future work. Section 10 ("Manageability Considerations") is particularly relevant in the context of this section because it contains a discussion of the policies and mechanisms for indicating connectivity and link availability between network layers in this architecture.

6. Application of the Architecture to Optical Domains and Networks

Many optical networks are arranged as a set of small domains. Each domain is a cluster of nodes, usually from the same equipment vendor and with the same properties. The domain may be constructed as a mesh or a ring, or maybe as an interconnected set of rings.

The network operator seeks to provide end-to-end connectivity across a network constructed from multiple domains, and so (of course) the domains are interconnected. In a network under management control, such as through an Operations Support System (OSS), each domain is under the operational control of a Network Management System (NMS).

In this way, an end-to-end path may be commissioned by the OSS instructing each NMS, and the NMSes setting up the path fragments across the domains.

However, in a system that uses a control plane, there is a need for integration between the domains.

Consider a simple domain, D1, as shown in Figure 19. In this case, nodes A through F are arranged in a topological ring. Suppose that there is a control plane in use in this domain and that OSPF is used as the TE routing protocol.

```
 -----------------
|              D1 |
|      B---C      |
|     /     \     |
|    /       \    |
|   A         D   |
|    \       /    |
|     \     /     |
|      F---E      |
|                 |
 -----------------
```

Figure 19: A Simple Optical Domain

Now consider that the operator’s network is built from a mesh of such domains, D1 through D7, as shown in Figure 20. It is possible that these domains share a single, common instance of OSPF, in which case there is nothing further to say because that OSPF instance will distribute sufficient information to build a single TED spanning the whole network, and an end-to-end path can be computed. A more likely scenario is that each domain is running its own OSPF instance. In this case, each is able to handle the peculiarities (or, rather, advanced functions) of each vendor’s equipment capabilities.

```
                  ------     ------     ------     ------
                 |      |   |      |   |      |   |      |
                 |  D1  |---|  D2  |---|  D3  |---|  D4  |
                 |      |   |      |   |      |   |      |
                  ------\    ------\    ------\    ------
                         \    |     \     |    \     |
                          \------    \------    \------
                          |      |   |      |   |      |
                          |  D5  |---|  D6  |---|  D7  |
                          |      |   |      |   |      |
                           ------     ------     ------
```

Figure 20: A Mesh of Simple Optical Domains

The question now is how to combine the multiple sets of information distributed by the different OSPF instances. Three possible models suggest themselves, based on pre-existing routing practices.

- In the first model (the area-based model), each domain is treated as a separate OSPF area. The end-to-end path will be specified to traverse multiple areas, and each area will be left to determine the path across the nodes in the area. The feasibility of an end-to-end path (and, thus, the selection of the sequence of areas and their interconnections) can be derived using hierarchical PCEs.

  This approach, however, fits poorly with established use of the OSPF area: in this form of optical network, the interconnection points between domains are likely to be links, and the mesh of domains is far more interconnected and unstructured than we are used to seeing in the normal area-based routing paradigm.

  Furthermore, while hierarchical PCEs may be able to resolve this type of network, the effort involved may be considerable for more than a small collection of domains.

- Another approach (the AS-based model) treats each domain as a separate Autonomous System (AS). The end-to-end path will be specified to traverse multiple ASes, and each AS will be left to determine the path across the nodes in that AS.

  This model sits more comfortably with the established routing paradigm but causes a massive escalation of ASes in the global Internet. It would, in practice, require that the operator use private AS numbers [RFC6996], of which there are plenty.

Then, as suggested in the area-based model, hierarchical PCEs could be used to determine the feasibility of an end-to-end path and to derive the sequence of domains and the points of interconnection to use. But just as in the area-based model, the scalability of this model using a hierarchical PCE must be questioned, given the sheer number of ASes and their interconnectivity.

Furthermore, determining the mesh of domains (i.e., the inter-AS connections) conventionally requires the use of BGP as an inter-domain routing protocol. However, not only is BGP not normally available on optical equipment, but this approach indicates that the TE properties of the inter-domain links would need to be distributed and updated using BGP -- something for which it is not well suited.

- The third approach (the Automatically Switched Optical Network (ASON) model) follows the architectural model set out by the ITU-T [G.8080] and uses the routing protocol extensions described in [RFC6827]. In this model, the concept of "levels" is introduced to OSPF. Referring back to Figure 20, each OSPF instance running in a domain would be construed as a "lower-level" OSPF instance and would leak routes into a "higher-level" instance of the protocol that runs across the whole network.

  This approach handles the awkwardness of representing the domains as areas or ASes by simply considering them as domains running distinct instances of OSPF. Routing advertisements flow "upward" from the domains to the high-level OSPF instance, giving it a full view of the whole network and allowing end-to-end paths to be computed. Routing advertisements may also flow "downward" from the network-wide OSPF instance to any one domain so that it can see the connectivity of the whole network.

  Although architecturally satisfying, this model suffers from having to handle the different characteristics of different equipment vendors. The advertisements coming from each low-level domain would be meaningless when distributed into the other domains, and the high-level domain would need to be kept up to date with the semantics of each new release of each vendor’s equipment. Additionally, the scaling issues associated with a well-meshed network of domains, each with many entry and exit points and each with network resources that are continually being updated, reduces to the same problem, as noted in the virtual link model. Furthermore, in the event that the domains are under the control of different administrations, the domains would not want to distribute the details of their topologies and TE resources.

Practically, this third model turns out to be very close to the methodology described in this document. As noted in Section 6.1 of [RFC6827], there are policy rules that can be applied to define exactly what information is exported from or imported to a low-level OSPF instance. [RFC6827] even notes that some forms of aggregation may be appropriate. Thus, we can apply the following simplifications to the mechanisms defined in [RFC6827]:

- Zero information is imported to low-level domains.

- Low-level domains export only abstracted links as defined in this document and according to local abstraction policy, and with appropriate removal of vendor-specific information.

- There is no need to formally define routing levels within OSPF.

- Export of abstracted links from the domains to the network-wide routing instance (the abstraction routing layer) can take place through any mechanism, including BGP-LS or direct interaction between OSPF implementations.

With these simplifications, it can be seen that the framework defined in this document can be constructed from the architecture discussed in [RFC6827], but without needing any of the protocol extensions defined in that document. Thus, using the terminology and concepts already established, the problem may be solved as shown in Figure 21. The abstraction layer network is constructed from the inter-domain links, the domain border nodes, and the abstracted (cross-domain) links.

```
                                                       Abstraction Layer
      --             --    --             --    --             --
     |  |===========|  |--|  |===========|  |--|  |===========|  |
     |  |           |  |  |  |           |  |  |  |           |  |
   ..|  |...........|  |..|  |...........|  |..|  |...........|  |......
     |  |           |  |  |  |           |  |  |  |           |  |
     |  |  --   --  |  |  |  |  --   --  |  |  |  |  --   --  |  |
     |  |_|  |_|  |_|  |  |  |_|  |_|  |_|  |  |  |_|  |_|  |_|  |
     |  | |  | |  | |  |  |  | |  | |  | |  |  |  | |  | |  | |  |
      --   --   --   --    --   --   --   --    --   --   --   --
          Domain 1             Domain 2             Domain 3
     Key                                                   Optical Layer
       ...  Layer separation
       ---  Physical link
       ===  Abstract link
```

Figure 21: The Optical Network Implemented through the Abstraction Layer Network

7. Application of the Architecture to the User-Network Interface

The User-Network Interface (UNI) is an important architectural concept in many implementations and deployments of client-server networks, especially those where the client and server network have different technologies. The UNI is described in [G.8080], and the GMPLS approach to the UNI is documented in [RFC4208]. Other GMPLS-related documents describe the application of GMPLS to specific UNI scenarios: for example, [RFC6005] describes how GMPLS can support a UNI that provides access to Ethernet services.

Figure 1 of [RFC6005] is reproduced here as Figure 22. It shows the Ethernet UNI reference model, and that figure can serve as an example for all similar UNIs. In this case, the UNI is an interface between client network edge nodes and the server network. It should be noted that neither the client network nor the server network need be an Ethernet switching network.

There are three network layers in this model: the client network, the "Ethernet service network", and the server network. The so-called Ethernet service network consists of links comprising the UNI links and the tunnels across the server network, and nodes comprising the client network edge nodes and various server network nodes. That is, the Ethernet service network is equivalent to the abstraction layer network, with the UNI links being the physical links between the client and server networks, the client edge nodes taking the role of UNI Client-side (UNI-C) nodes, and the server edge nodes acting as the UNI Network-side (UNI-N) nodes.

```
     Client                                          Client
     Network       +----------+    +-----------+       Network
-------------+     |          |    |           |     +-------------
      +----+ |     |  +-----+ |    |  +-----+  |     | +----+
------+    | |     |  |     | |    |  |     |  |     | |    +------
------+ EN +-+-----+--+ CN  +-+----+--+  CN +--+-----+-+ EN +------
      |    | |  +--+--|     +-+-+  |  |     +--+-----+-+    |
      +----+ |  |  |  +--+--+ | |  |  +--+--+  |     | +----+
             |  |  |     |    | |  |     |     |     |
-------------+  |  |     |    | |  |     |     |     +-------------
                |  |     |    | |  |     |     |
-------------+  |  |     |    | |  |     |     |     +-------------
             |  |  |  +--+--+ | |  |  +--+--+  |     |
      +----+ |  |  |  |     | | +--+--+     |  |     | +----+
------+    +-+--+  |  | CN  +-+----+--+ CN  |  |     | |    +------
------+ EN +-+-----+--+     | |    |  |     +--+-----+-+ EN +------
      |    | |     |  +-----+ |    |  +-----+  |     | |    |
      +----+ |     |          |    |           |     | +----+
             |     +----------+    |-----------+     |
-------------+           Server Networks             +-------------
     Client    UNI                               UNI   Client
     Network <----->                           <-----> Network
                       Scope of This Document

                     Legend:   EN  -  Client Network Edge Node
                               CN  -  Server Network (Core) Node
```

Figure 22: Ethernet UNI Reference Model

An issue that is often raised relates to how a dual-homed client network edge node (such as that shown at the bottom left-hand corner of Figure 22) can make determinations about how they connect across the UNI. This can be particularly important when reachability across the server network is limited or when two diverse paths are desired (for example, to provide protection). However, in the model described in this network, the edge node (the UNI-C node) is part of the abstraction layer network and can see sufficient topology information to make these decisions. If the approach introduced in this document is used to model the UNI as described in this section, there is no need to enhance the signaling protocols at the GMPLS UNI nor to add routing exchanges at the UNI.

8. Application of the Architecture to L3VPN Multi-AS Environments

Serving Layer 3 VPNs (L3VPNs) across a multi-AS or multi-operator environment currently provides a significant planning challenge. Figure 6 shows the general case of the problem that needs to be solved. This section shows how the abstraction layer network can address this problem.

In the VPN architecture, the CE nodes are the client network edge nodes, and the PE nodes are the server network edge nodes. The abstraction layer network is made up of the CE nodes, the CE-PE links, the PE nodes, and PE-PE tunnels that are the abstract links.

In the multi-AS or multi-operator case, the abstraction layer network also includes the PEs (maybe Autonomous System Border Routers (ASBRs)) at the edges of the multiple server networks, and the PE-PE (maybe inter-AS) links. This gives rise to the architecture shown in Figure 23.

The policy for adding abstract links to the abstraction layer network will be driven substantially by the needs of the VPN. Thus, when a new VPN site is added and the existing abstraction layer network cannot support the required connectivity, a new abstract link will be created out of the underlying network.

```
       ...........                                     .............
        VPN Site :                                     : VPN Site
        --   --  :                                     :  --   --
       |C1|-|CE| :                                     : |CE|-|C2|
        --  |  | :                                     : |  |  --
            |  | :                                     : |  |
            |  | :                                     : |  |
            |  | :                                     : |  |
            |  | :   --            --     --      --   : |  |
            |  |----|PE|=========|PE|---|PE|=====|PE|----|  |
             --  :  |  |         |  |   |  |     |  |  :  --
       ...........  |  |         |  |   |  |     |  |  ............
                    |  |         |  |   |  |     |  |
                    |  |         |  |   |  |     |  |
                    |  |         |  |   |  |     |  |
                    |  |  -   -  |  |   |  |  -  |  |
                    |  |-|P|-|P|-|  |   |  |-|P|-|  |
                     --   -   -   --     --   -   --
```

Figure 23: The Abstraction Layer Network for a Multi-AS VPN

It is important to note that each VPN instance can have a separate abstraction layer network. This means that the server network resources can be partitioned and that traffic can be kept separate.

This can be achieved even when VPN sites from different VPNs connect at the same PE. Alternatively, multiple VPNs can share the same abstraction layer network if that is operationally preferable.

Lastly, just as for the UNI discussed in Section 7, the issue of dual-homing of VPN sites is a function of the abstraction layer network and so is just a normal routing problem in that network.

9. Scoping Future Work

This section is provided to help guide the work on this problem. The overarching view is that it is important to limit and focus the work on those things that are core and necessary to achieve the main function, and to not attempt to add unnecessary features or to over-complicate the architecture or the solution by attempting to address marginal use cases or corner cases. This guidance is non-normative for this architecture description.

9.1. Limiting Scope to Only Part of the Internet

The scope of the use cases and problem statement in this document is limited to "some small set of interconnected domains." In particular, it is not the objective of this work to turn the whole Internet into one large, interconnected TE network.

9.2. Working with "Related" Domains

Starting with this subsection, the intention of this work is to solve the TE interconnectivity for only "related" domains. Such domains may be under common administrative operation (such as IGP areas within a single AS, or ASes belonging to a single operator) or may have a direct commercial arrangement for the sharing of TE information to provide specific services. Thus, in both cases, there is a strong opportunity for the application of policy.

9.3. Not Finding Optimal Paths in All Situations

As has been well described in this document, abstraction necessarily involves compromises and removal of information. That means that it is not possible to guarantee that an end-to-end path over interconnected TE domains follows the absolute optimal (by any measure of optimality) path. This is taken as understood, and future work should not attempt to achieve such paths, which can only be found by a full examination of all network information across all connected networks.

9.4. Sanity and Scaling

All of the above points play into a final observation. This work is intended to "bite off" a small problem for some relatively simple use cases as described in Section 2. It is not intended that this work will be immediately (or even soon) extended to cover many large interconnected domains. Obviously, the solution should, as far as possible, be designed to be extensible and scalable; however, it is also reasonable to make trade-offs in favor of utility and simplicity.

10. Manageability Considerations

Manageability should not be a significant additional burden. Each layer in the network model can, and should, be managed independently.

That is, each client network will run its own management systems and tools to manage the nodes and links in the client network: each client network link that uses an abstract link will still be available for management in the client network as any other link.

Similarly, each server network will run its own management systems and tools to manage the nodes and links in that network just as normal.

Three issues remain for consideration:

- How is the abstraction layer network managed?

- How is the interface between the client network and the abstraction layer network managed?

- How is the interface between the abstraction layer network and the server network managed?

10.1. Managing the Abstraction Layer Network

Management of the abstraction layer network differs from the client and server networks because not all of the links that are visible in the TED are real links. That is, it is not possible to run Operations, Administration, and Maintenance (OAM) on the links that constitute the potential of a link.

Other than that, however, the management of the abstraction layer network should be essentially the same. Routing and signaling protocols can be run in the abstraction layer (using out-of-band channels for links that have not yet been established), and a centralized TED can be constructed and used to examine the availability and status of the links and nodes in the network.

Note that different deployment models will place the "ownership" of the abstraction layer network differently. In some cases, the abstraction layer network will be constructed by the operator of the server network and run by that operator as a service for one or more client networks. In other cases, one or more server networks will present the potential of links to an abstraction layer network run by the operator of the client network. And it is feasible that a business model could be built where a third-party operator manages the abstraction layer network, constructing it from the connectivity available in multiple server networks and facilitating connectivity for multiple client networks.

10.2. Managing Interactions of Abstraction Layer and Client Networks

The interaction between the client network and the abstraction layer network is a management task. It might be automated (software driven), or it might require manual intervention.

This is a two-way interaction:

- The client network can express the need for additional connectivity. For example, the client network may try, and fail, to find a path across the client network and may request additional, specific connectivity (this is similar to the situation with the Virtual Network Topology Manager (VNTM) [RFC5623]). Alternatively, a more proactive client network management system may monitor traffic demands (current and predicted), network usage, and network "hot spots" and may request changes in connectivity by both releasing unused links and requesting new links.

- The abstraction layer network can make links available to the client network or can withdraw them. These actions can be in response to requests from the client network or can be driven by processes within the abstraction layer (perhaps reorganizing the use of server network resources). In any case, the presentation of new links to the client network is heavily subject to policy, since this is both operationally key to the success of this architecture and the central plank of the commercial model described in this document. Such policies belong to the operator of the abstraction layer network and are expected to be fully configurable.

  Once the abstraction layer network has decided to make a link available to the client network, it will install it at the link end points (which are nodes in the client network) such that it appears and can be advertised as a link in the client network.

In all cases, it is important that the operators of both networks are able to track the requests and responses, and the operator of the client network should be able to see which links in that network are "real" physical links and which links are presented by the abstraction layer network.

10.3. Managing Interactions of Abstraction Layer and Server Networks

The interactions between the abstraction layer network and the server network are similar to those described in Section 10.2, but there is a difference in that the server network is more likely to offer up connectivity and the abstraction layer network is less likely to ask for it.

That is, the server network will, according to policy that may include commercial relationships, offer the abstraction layer network a "set" of potential connectivity that the abstraction layer network can treat as links. This server network policy will include:

- how much connectivity to offer

- what level of server network redundancy to include

- how to support the use of the abstract links

This process of offering links from the server network may include a mechanism to indicate which links have been pre-established in the server network and can include other properties, such as:

- link-level protection [RFC4202]

- SRLGs and MSRLGs (see Appendix B.1)

- mutual exclusivity (see Appendix B.2)

The abstraction layer network needs a mechanism to tell the server network which links it is using. This mechanism could also include the ability to request additional connectivity from the server network, although it seems most likely that the server network will already have presented as much connectivity as it is physically capable of, subject to the constraints of policy.

Finally, the server network will need to confirm the establishment of connectivity, withdraw links if they are no longer feasible, and report failures.

Again, it is important that the operators of both networks are able to track the requests and responses, and the operator of the server network should be able to see which links are in use.

11. Security Considerations

Security of signaling and routing protocols is usually administered and achieved within the boundaries of a domain. Thus, and for example, a domain with a GMPLS control plane [RFC3945] would apply the security mechanisms and considerations that are appropriate to GMPLS [RFC5920]. Furthermore, domain-based security relies strongly on ensuring that control-plane messages are not allowed to enter the domain from outside.

In this context, additional security considerations arising from this document relate to the exchange of control-plane information between domains. Messages are passed between domains using control-plane protocols operating between peers that have predictable relationships (for example, UNI-C to UNI-N, between BGP-LS speakers, or between peer domains). Thus, the security that needs to be given additional attention for inter-domain TE concentrates on authentication of peers; assertion that messages have not been tampered with; and, to a lesser extent, protecting the content of the messages from inspection, since that might give away sensitive information about the networks. The protocols described in Appendix A, which are likely to provide the foundation for solutions to this architecture, already include such protection and also can be run over protected transports such as IPsec [RFC6071], Transport Layer Security (TLS) [RFC5246], and the TCP Authentication Option (TCP-AO) [RFC5925].

It is worth noting that the control plane of the abstraction layer network is likely to be out of band. That is, control-plane messages will be exchanged over network links that are not the links to which they apply. This models the facilities of GMPLS (but not of MPLS-TE), and the security mechanisms can be applied to the protocols operating in the out-of-band network.

12. Informative References

[G.8080] International Telecommunication Union, "Architecture for the automatically switched optical network", ITU-T Recommendation G.8080/Y.1304, February 2012, <https://www.itu.int/rec/T-REC-G.8080-201202-I/en>.

[GMPLS-ENNI] Bryskin, I., Ed., Doonan, W., Beeram, V., Ed., Drake, J., Ed., Grammel, G., Paul, M., Kunze, R., Armbruster, F., Margaria, C., Gonzalez de Dios, O., and D. Ceccarelli, "Generalized Multiprotocol Label Switching (GMPLS) External Network Network Interface (E-NNI): Virtual Link Enhancements for the Overlay Model", Work in Progress, draft-beeram-ccamp-gmpls-enni-03, September 2013.

[RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, DOI 10.17487/RFC2702, September 1999, <http://www.rfc-editor.org/info/rfc2702>.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <http://www.rfc-editor.org/info/rfc3209>.

[RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <http://www.rfc-editor.org/info/rfc3473>.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <http://www.rfc-editor.org/info/rfc3630>.

[RFC3945] Mannie, E., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, DOI 10.17487/RFC3945, October 2004, <http://www.rfc-editor.org/info/rfc3945>.

[RFC4105] Le Roux, J.-L., Ed., Vasseur, J.-P., Ed., and J. Boyle, Ed., "Requirements for Inter-Area MPLS Traffic Engineering", RFC 4105, DOI 10.17487/RFC4105, June 2005, <http://www.rfc-editor.org/info/rfc4105>.

[RFC4202] Kompella, K., Ed., and Y. Rekhter, Ed., "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4202, DOI 10.17487/RFC4202, October 2005, <http://www.rfc-editor.org/info/rfc4202>.

[RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, DOI 10.17487/RFC4206, October 2005, <http://www.rfc-editor.org/info/rfc4206>.

[RFC4208] Swallow, G., Drake, J., Ishimatsu, H., and Y. Rekhter, "Generalized Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model", RFC 4208, DOI 10.17487/RFC4208, October 2005, <http://www.rfc-editor.org/info/rfc4208>.

[RFC4216] Zhang, R., Ed., and J.-P. Vasseur, Ed., "MPLS Inter-Autonomous System (AS) Traffic Engineering (TE) Requirements", RFC 4216, DOI 10.17487/RFC4216, November 2005, <http://www.rfc-editor.org/info/rfc4216>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <http://www.rfc-editor.org/info/rfc4271>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <http://www.rfc-editor.org/info/rfc4364>.

[RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <http://www.rfc-editor.org/info/rfc4655>.

[RFC4726] Farrel, A., Vasseur, J.-P., and A. Ayyangar, "A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering", RFC 4726, DOI 10.17487/RFC4726, November 2006, <http://www.rfc-editor.org/info/rfc4726>.

[RFC4847] Takeda, T., Ed., "Framework and Requirements for Layer 1 Virtual Private Networks", RFC 4847, DOI 10.17487/RFC4847, April 2007, <http://www.rfc-editor.org/info/rfc4847>.

[RFC4874] Lee, CY., Farrel, A., and S. De Cnodder, "Exclude Routes - Extension to Resource ReserVation Protocol-Traffic Engineering (RSVP-TE)", RFC 4874, DOI 10.17487/RFC4874, April 2007, <http://www.rfc-editor.org/info/rfc4874>.

[RFC4920] Farrel, A., Ed., Satyanarayana, A., Iwata, A., Fujita, N., and G. Ash, "Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE", RFC 4920, DOI 10.17487/RFC4920, July 2007, <http://www.rfc-editor.org/info/rfc4920>.

[RFC5150] Ayyangar, A., Kompella, K., Vasseur, JP., and A. Farrel, "Label Switched Path Stitching with Generalized Multiprotocol Label Switching Traffic Engineering (GMPLS TE)", RFC 5150, DOI 10.17487/RFC5150, February 2008, <http://www.rfc-editor.org/info/rfc5150>.

[RFC5152] Vasseur, JP., Ed., Ayyangar, A., Ed., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, DOI 10.17487/RFC5152, February 2008, <http://www.rfc-editor.org/info/rfc5152>.

[RFC5195] Ould-Brahim, H., Fedyk, D., and Y. Rekhter, "BGP-Based Auto-Discovery for Layer-1 VPNs", RFC 5195, DOI 10.17487/RFC5195, June 2008, <http://www.rfc-editor.org/info/rfc5195>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[RFC5251] Fedyk, D., Ed., Rekhter, Y., Ed., Papadimitriou, D., Rabbat, R., and L. Berger, "Layer 1 VPN Basic Mode", RFC 5251, DOI 10.17487/RFC5251, July 2008, <http://www.rfc-editor.org/info/rfc5251>.

[RFC5252] Bryskin, I. and L. Berger, "OSPF-Based Layer 1 VPN Auto-Discovery", RFC 5252, DOI 10.17487/RFC5252, July 2008, <http://www.rfc-editor.org/info/rfc5252>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <http://www.rfc-editor.org/info/rfc5305>.

[RFC5440] Vasseur, JP., Ed., and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <http://www.rfc-editor.org/info/rfc5440>.

[RFC5441] Vasseur, JP., Ed., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, DOI 10.17487/RFC5441, April 2009, <http://www.rfc-editor.org/info/rfc5441>.

[RFC5523] Berger, L., "OSPFv3-Based Layer 1 VPN Auto-Discovery", RFC 5523, DOI 10.17487/RFC5523, April 2009, <http://www.rfc-editor.org/info/rfc5523>.

[RFC5553] Farrel, A., Ed., Bradford, R., and JP. Vasseur, "Resource Reservation Protocol (RSVP) Extensions for Path Key Support", RFC 5553, DOI 10.17487/RFC5553, May 2009, <http://www.rfc-editor.org/info/rfc5553>.

[RFC5623] Oki, E., Takeda, T., Le Roux, JL., and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, DOI 10.17487/RFC5623, September 2009, <http://www.rfc-editor.org/info/rfc5623>.

[RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <http://www.rfc-editor.org/info/rfc5920>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <http://www.rfc-editor.org/info/rfc5925>.

[RFC6005] Berger, L. and D. Fedyk, "Generalized MPLS (GMPLS) Support for Metro Ethernet Forum and G.8011 User Network Interface (UNI)", RFC 6005, DOI 10.17487/RFC6005, October 2010, <http://www.rfc-editor.org/info/rfc6005>.

[RFC6071] Frankel, S. and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap", RFC 6071, DOI 10.17487/RFC6071, February 2011, <http://www.rfc-editor.org/info/rfc6071>.

[RFC6107] Shiomoto, K., Ed., and A. Farrel, Ed., "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, DOI 10.17487/RFC6107, February 2011, <http://www.rfc-editor.org/info/rfc6107>.

[RFC6805] King, D., Ed., and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <http://www.rfc-editor.org/info/rfc6805>.

[RFC6827] Malis, A., Ed., Lindem, A., Ed., and D. Papadimitriou, Ed., "Automatically Switched Optical Network (ASON) Routing for OSPFv2 Protocols", RFC 6827, DOI 10.17487/RFC6827, January 2013, <http://www.rfc-editor.org/info/rfc6827>.

[RFC6996] Mitchell, J., "Autonomous System (AS) Reservation for Private Use", BCP 6, RFC 6996, DOI 10.17487/RFC6996, July 2013, <http://www.rfc-editor.org/info/rfc6996>.

[RFC7399] Farrel, A. and D. King, "Unanswered Questions in the Path Computation Element Architecture", RFC 7399, DOI 10.17487/RFC7399, October 2014, <http://www.rfc-editor.org/info/rfc7399>.

[RFC7579] Bernstein, G., Ed., Lee, Y., Ed., Li, D., Imajuku, W., and J. Han, "General Network Element Constraint Encoding for GMPLS-Controlled Networks", RFC 7579, DOI 10.17487/RFC7579, June 2015, <http://www.rfc-editor.org/info/rfc7579>.

[RFC7580] Zhang, F., Lee, Y., Han, J., Bernstein, G., and Y. Xu, "OSPF-TE Extensions for General Network Element Constraints", RFC 7580, DOI 10.17487/RFC7580, June 2015, <http://www.rfc-editor.org/info/rfc7580>.

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <http://www.rfc-editor.org/info/rfc7752>.

[RSVP-TE-EXCL]
Ali, Z., Ed., Swallow, G., Ed., Zhang, F., Ed., and D. Beller, Ed., "Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Path Diversity using Exclude Route", Work in Progress, draft-ietf-teas-lsp-diversity-05, June 2016.

[RSVP-TE-EXT]
Zhang, F., Ed., Gonzalez de Dios, O., Ed., Hartley, M., Ali, Z., and C. Margaria, "RSVP-TE Extensions for Collecting SRLG Information", Work in Progress, draft-ietf-teas-rsvp-te-srlg-collect-06, May 2016.

[RSVP-TE-METRIC]
Ali, Z., Swallow, G., Filsfils, C., Hartley, M., Kumaki, K., and R. Kunze, "Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) extension for recording TE Metric of a Label Switched Path", Work in Progress, draft-ietf-teas-te-metric-recording-04, March 2016.

Appendix A. Existing Work

This appendix briefly summarizes relevant existing work that is used to route TE paths across multiple domains. It is non-normative.

A.1. Per-Domain Path Computation

The mechanism for per-domain path establishment is described in [RFC5152], and its applicability is discussed in [RFC4726]. In summary, this mechanism assumes that each domain entry point is responsible for computing the path across the domain but that details regarding the path in the next domain are left to the next domain entry point. The computation may be performed directly by the entry point or may be delegated to a computation server.

This basic mode of operation can run into many of the issues described alongside the use cases in Section 2. However, in practice it can be used effectively, with a little operational guidance.

For example, RSVP-TE [RFC3209] includes the concept of a "loose hop" in the explicit path that is signaled. This allows the original request for an LSP to list the domains or even domain entry points to include on the path. Thus, in the example in Figure 1, the source can be told to use interconnection x2. Then, the source computes the path from itself to x2 and initiates the signaling. When the signaling message reaches Domain Z, the entry point to the domain computes the remaining path to the destination and continues the signaling.

Another alternative suggested in [RFC5152] is to make TE routing attempt to follow inter-domain IP routing. Thus, in the example shown in Figure 2, the source would examine the BGP routing information to determine the correct interconnection point for forwarding IP packets and would use that to compute and then signal a path for Domain A. Each domain in turn would apply the same approach so that the path is progressively computed and signaled domain by domain.

Although the per-domain approach has many issues and drawbacks in terms of achieving optimal (or, indeed, any) paths, it has been the mainstay of inter-domain LSP setup to date.

A.2. Crankback

Crankback addresses one of the main issues with per-domain path computation: What happens when an initial path is selected that cannot be completed toward the destination? For example, what happens if, in Figure 2, the source attempts to route the path through interconnection x2 but Domain C does not have the right TE resources or connectivity to route the path further?

Crankback for MPLS-TE and GMPLS networks is described in [RFC4920] and is based on a concept similar to the Acceptable Label Set mechanism described for GMPLS signaling in [RFC3473]. When a node (i.e., a domain entry point) is unable to compute a path further across the domain, it returns an error message in the signaling protocol that states where the blockage occurred (link identifier, node identifier, domain identifier, etc.) and gives some clues about what caused the blockage (bad choice of label, insufficient bandwidth available, etc.). This information allows a previous computation point to select an alternative path, or to aggregate crankback information and return it upstream to a previous computation point.

Crankback is a very powerful mechanism and can be used to find an end-to-end path in a multi-domain network if one exists.

On the other hand, crankback can be quite resource-intensive, as signaling messages and path setup attempts may "wander around" in the network, attempting to find the correct path for a long time. Since (1) RSVP-TE signaling ties up network resources for partially established LSPs, (2) network conditions may be in flux, and (3) most particularly, LSP setup within well-known time limits is highly desirable, crankback is not a popular mechanism.

Furthermore, even if crankback can always find an end-to-end path, it does not guarantee that the optimal path will be found. (Note that there have been some academic proposals to use signaling-like techniques to explore the whole network in order to find optimal paths, but these tend to place even greater burdens on network processing.)

A.3. Path Computation Element

The Path Computation Element (PCE) is introduced in [RFC4655]. It is an abstract functional entity that computes paths. Thus, in the example of per-domain path computation (see Appendix A.1), both the source node and each domain entry point are PCEs. On the other hand, the PCE can also be realized as a separate network element (a server) to which computation requests can be sent using the Path Computation Element Communication Protocol (PCEP) [RFC5440].

Each PCE is responsible for computations within a domain and has visibility of the attributes within that domain. This immediately enables per-domain path computation with the opportunity to offload complex, CPU-intensive, or memory-intensive computation functions from routers in the network. But the use of PCEs in this way does not solve any of the problems articulated in Appendices A.1 and A.2.

Two significant mechanisms for cooperation between PCEs have been described. These mechanisms are intended to specifically address the problems of computing optimal end-to-end paths in multi-domain environments.

- The Backward-Recursive PCE-Based Computation (BRPC) mechanism [RFC5441] involves cooperation between the set of PCEs along the inter-domain path. Each one computes the possible paths from the domain entry point (or source node) to the domain exit point (or destination node) and shares the information with its upstream neighbor PCE, which is able to build a tree of possible paths rooted at the destination. The PCE in the source domain can select the optimal path.

  BRPC is sometimes described as "crankback at computation time". It is capable of determining the optimal path in a multi-domain network but depends on knowing the domain that contains the destination node. Furthermore, the mechanism can become quite complicated and can involve a lot of data in a mesh of interconnected domains. Thus, BRPC is most often proposed for a simple mesh of domains and specifically for a path that will cross a known sequence of domains, but where there may be a choice of domain interconnections. In this way, BRPC would only be applied to Figure 2 if a decision had been made (externally) to traverse Domain C rather than Domain D (notwithstanding that it could functionally be used to make that choice itself), but BRPC could be used very effectively to select between interconnections x1 and x2 in Figure 1.

- The Hierarchical PCE (H-PCE) [RFC6805] mechanism offers a parent PCE that is responsible for navigating a path across the domain mesh and for coordinating intra-domain computations by the child PCEs responsible for each domain. This approach makes computing an end-to-end path across a mesh of domains far more tractable. However, it still leaves unanswered the issue of determining the location of the destination (i.e., discovering the destination domain) as described in Section 2.1. Furthermore, it raises the question of who operates the parent PCE, especially in networks where the domains are under different administrative and commercial control.

It should also be noted that [RFC5623] discusses how PCEs are used in a multi-layer network with coordination between PCEs operating at each network layer. Further issues and considerations regarding the use of PCEs can be found in [RFC7399].

A.4. GMPLS UNI and Overlay Networks

[RFC4208] defines the GMPLS User-Network Interface (UNI) to present a routing boundary between an overlay (client) network and the server network, i.e., the client-server interface. In the client network, the nodes connected directly to the server network are known as edge nodes, while the nodes in the server network are called core nodes.

In the overlay model defined by [RFC4208], the core nodes act as a closed system and the edge nodes do not participate in the routing protocol instance that runs among the core nodes. Thus, the UNI allows access to, and limited control of, the core nodes by edge nodes that are unaware of the topology of the core nodes. This respects the operational and layer boundaries while scaling the network.

[RFC4208] does not define any routing protocol extension for the interaction between core and edge nodes but allows for the exchange of reachability information between them. In terms of a VPN, the client network can be considered as the customer network comprised of a number of disjoint sites, and the edge nodes match the VPN CE nodes. Similarly, the provider network in the VPN model is equivalent to the server network.

[RFC4208] is, therefore, a signaling-only solution that allows edge nodes to request connectivity across the server network and leaves the server network to select the paths for the LSPs as they traverse the core nodes (setting up hierarchical LSPs if necessitated by the technology). This solution is supplemented by a number of signaling extensions, such as [RFC4874], [RFC5553], [RSVP-TE-EXCL], [RSVP-TE-EXT], and [RSVP-TE-METRIC], to give the edge node more control over the path within the server network and by allowing the edge nodes to supply additional constraints on the path used in the server network. Nevertheless, in this UNI/overlay model, the edge node has limited information regarding precisely what LSPs could be set up across the server network and what TE services (diverse routes for end-to-end protection, end-to-end bandwidth, etc.) can be supported.

A.5. Layer 1 VPN

A Layer 1 VPN (L1VPN) is a service offered by a Layer 1 server network to provide Layer 1 connectivity (Time-Division Multiplexing (TDM), Lambda Switch Capable (LSC)) between two or more customer networks in an overlay service model [RFC4847].

As in the UNI case, the customer edge has some control over the establishment and type of connectivity. In the L1VPN context, three different service models have been defined, classified by the semantics of information exchanged over the customer interface: the management-based model, the signaling-based (a.k.a. basic) service model, and the signaling and routing (a.k.a. enhanced) service model.

In the management-based model, all edge-to-edge connections are set up using configuration and management tools. This is not a dynamic control-plane solution and need not concern us here.

In the signaling-based (basic) service model [RFC5251], the CE-PE interface allows only for signaling message exchange, and the provider network does not export any routing information about the server network. VPN membership is known a priori (presumably through configuration) or is discovered using a routing protocol [RFC5195] [RFC5252] [RFC5523], as is the relationship between CE nodes and ports on the PE. This service model is much in line with GMPLS UNI as defined in [RFC4208].

In the signaling and routing (enhanced) service model, there is an additional limited exchange of routing information over the CE-PE interface between the provider network and the customer network. The enhanced model considers four different types of service models, namely the overlay extension, virtual node, virtual link, and per-VPN service models. All of these represent particular cases of the TE information aggregation and representation.

A.6. Policy and Link Advertisement

Inter-domain networking relies on policy and management input to coordinate the allocation of resources under different administrative control. [RFC5623] introduces a functional component called the VNTM for this purpose.

An important companion to this function is determining how connectivity across the abstraction layer network is made available as a TE link in the client network. Obviously, if the connectivity is established using management intervention, the consequent client network TE link can also be configured manually. However, if connectivity from client edge to client edge is achieved using dynamic signaling, then there is need for the end points to exchange the link properties that they should advertise within the client network, and in the case of support for more than one client network, it will be necessary to indicate which client network or networks can use the link. This capability it provided in [RFC6107].

Appendix B. Additional Features

This appendix describes additional features that may be desirable and that can be achieved within this architecture. It is non-normative.

B.1. Macro Shared Risk Link Groups

Network links often share fate with one or more other links. That is, a scenario that may cause a link to fail could cause one or more other links to fail. This may occur, for example, if the links are supported by the same fiber bundle, or if some links are routed down the same duct or in a common piece of infrastructure such as a bridge. A common way to identify the links that may share fate is to label them as belonging to a Shared Risk Link Group (SRLG) [RFC4202].

TE links created from LSPs in lower layers may also share fate, and it can be hard for a client network to know about this problem because it does not know the topology of the server network or the path of the server network LSPs that are used to create the links in the client network.

For example, looking at the example used in Section 4.2.3 and considering the two abstract links S1-S3 and S1-S9, there is no way for the client network to know whether links C2-C0 and C2-C3 share fate. Clearly, if the client layer uses these links to provide a link-diverse end-to-end protection scheme, it needs to know that the links actually share a piece of network infrastructure (the server network link S1-S2).

Per [RFC4202], an SRLG represents a shared physical network resource upon which the normal functioning of a link depends. Multiple SRLGs can be identified and advertised for every TE link in a network. However, this can produce a scalability problem in a multi-layer network that equates to advertising in the client network the server network route of each TE link.

Macro SRLGs (MSRLGs) address this scaling problem and are a form of abstraction performed at the same time that the abstract links are derived. In this way, links that actually share resources in the server network are advertised as having the same MSRLG, rather than advertising each SRLG for each resource on each path in the server network. This saving is possible because the abstract links are formulated on behalf of the server network by a central management agency that is aware of all of the link abstractions being offered.

It may be noted that a less optimal alternative path for the abstract link S1-S9 exists in the server network (S1-S4-S7-S8-S9). It would be possible for the client network request for C2-C0 connectivity to also ask that the path be maximally disjoint from path C2-C3. Although nothing can be done about the shared link C2-S1, the abstraction layer could make a request to use link S1-S9 in a way that is diverse from the use of link S1-S3, and this request could be honored if the server network policy allows it.

Note that SRLGs and MSRLGs may be very hard to describe in the case of multiple server networks because the abstraction points will not know whether the resources in the various server layers share physical locations.

B.2. Mutual Exclusivity

As noted in the discussion of Figure 13, it is possible that some abstraction layer links cannot be used at the same time. This arises when the potentiality of the links is indicated by the server network, but the use of the links would actually compete for server network resources. Referring to Figure 13, this situation would arise when both link S1-S3 and link S7-S9 are used to carry LSPs: in that case, link S1-S9 could no longer be used.

Such a situation need not be an issue when client-edge-to-client-edge LSPs are set up one by one, because the use of one abstraction layer link and the corresponding use of server network resources will cause the server network to withdraw the availability of the other abstraction layer links, and these will become unavailable for further abstraction layer path computations.

Furthermore, in deployments where abstraction layer links are only presented as available after server network LSPs have been established to support them, the problem is unlikely to exist.

However, when the server network is constrained but chooses to advertise the potential of multiple abstraction layer links even though they compete for resources, and when multiple client-edge-to-client-edge LSPs are computed simultaneously (perhaps to provide protection services), there may be contention for server network resources. In the case where protected abstraction layer LSPs are being established, this situation would be avoided through the use of SRLGs and/or MSRLGs, since the two abstraction layer links that compete for server network resources must also fate-share across those resources. But in the case where the multiple client-edge-to-client-edge LSPs do not care about fate sharing, it may be necessary to flag the mutually exclusive links in the abstraction layer TED so that path computation can avoid accidentally attempting to utilize two of a set of such links at the same time.

Acknowledgements

Thanks to Igor Bryskin for useful discussions in the early stages of this work and to Gert Grammel for discussions on the extent of aggregation in abstract nodes and links.

Thanks to Deborah Brungard, Dieter Beller, Dhruv Dhody, Vallinayakam Somasundaram, Hannes Gredler, Stewart Bryant, Brian Carpenter, and Hilarie Orman for review and input.

Particular thanks to Vishnu Pavan Beeram for detailed discussions and white-board scribbling that made many of the ideas in this document come to life.

Text in Section 4.2.3 is freely adapted from the work of Igor Bryskin, Wes Doonan, Vishnu Pavan Beeram, John Drake, Gert Grammel, Manuel Paul, Ruediger Kunze, Friedrich Armbruster, Cyril Margaria, Oscar Gonzalez de Dios, and Daniele Ceccarelli in [GMPLS-ENNI], for which the authors of this document express their thanks.

Contributors

Gert Grammel
Juniper Networks
Email: ggrammel@juniper.net

Vishnu Pavan Beeram
Juniper Networks
Email: vbeeram@juniper.net

Oscar Gonzalez de Dios
Email: ogondio@tid.es

Fatai Zhang
Email: zhangfatai@huawei.com

Zafar Ali
Email: zali@cisco.com

Rajan Rao
Email: rrao@infinera.com

Sergio Belotti
Email: sergio.belotti@alcatel-lucent.com

Diego Caviglia
Email: diego.caviglia@ericsson.com

Jeff Tantsura
Email: jeff.tantsura@ericsson.com

Khuzema Pithewan
Email: kpithewan@infinera.com

Cyril Margaria
Email: cyril.margaria@googlemail.com

Victor Lopez
Email: vlopez@tid.es

Authors' Addresses

Adrian Farrel (editor)
Juniper Networks

Email: adrian@olddog.co.uk

John Drake
Juniper Networks

Email: jdrake@juniper.net

Nabil Bitar
Nokia

Email: nbitar40@gmail.com

George Swallow
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough, MA 01719

Email: swallow@cisco.com

Daniele Ceccarelli
Ericsson
Via A. Negrone 1/A
Genova - Sestri Ponente
Italy

Email: daniele.ceccarelli@ericsson.com

Xian Zhang
Huawei Technologies

Email: zhang.xian@huawei.com