Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 23, 2015

A. Farrel (Ed.)
J. Drake
Juniper Networks

N. Bitar
Verizon Networks

G. Swallow
Cisco Systems, Inc.

D. Ceccarelli
Ericsson

X. Zhang
Huawei

September 23, 2014

Problem Statement and Architecture for Information Exchange Between Interconnected Traffic Engineered Networks

draft-farrel-interconnected-te-info-exchange-07.txt

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 network 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. The availability of TE information is usually limited to within a network (such as an IGP area) often referred to as a domain.

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 and architecture for the exchange of TE information between interconnected TE networks in support of end-to-end TE path establishment. For reasons that are explained in the document, this work is limited to simple TE constraints and information that determine TE reachability.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Copyright Notice

Table of Contents

1. Introduction 5
1.1. Terminology 6
1.1.1. TE Paths and TE Connections 6
1.1.2. TE Metrics and TE Attributes 6
1.1.3. TE Reachability 6
1.1.4. Domain 7
1.1.5. Aggregation 7
1.1.6. Abstraction 7
1.1.7. Abstract Link 7
1.1.8. Abstraction Layer Network 8
2. Overview of Use Cases 8
2.1. Peer Networks 8
2.1.1. Where is the Destination? 9
2.2. Client-Server Networks 10
2.3. Dual-Homing 12
2.4. Requesting Connectivity 13
2.4.1. Discovering Server Network Information 15
3. Problem Statement 15
3.1. Use of Existing Protocol Mechanisms 16
3.2. Policy and Filters 16
3.3. Confidentiality 17
3.4. Information Overload 17
3.5. Issues of Information Churn 18
3.6. Issues of Aggregation 19
3.7. Virtual Network Topology 20
4. Existing Work 21
4.1. Per-Domain Path Computation 21
4.2. Crankback 22
4.3. Path Computation Element 23
4.4. GMPLS UNI and Overlay Networks 24
4.5. Layer One VPN 25
4.6. VNT Manager and Link Advertisement 25
4.7. What Else is Needed and Why? 26
5. Architectural Concepts 26
5.1. Basic Components 26
5.1.1. Peer Interconnection 27
5.1.2. Client-Server Interconnection 27
5.2. TE Reachability 28
5.3. Abstraction not Aggregation 29
5.3.1. Abstract Links 30
5.3.2. The Abstraction Layer Network 30
5.3.3. Abstraction in Client-Server Networks..... 33
5.3.4. Abstraction in Peer Networks 34
5.4. Considerations for Dynamic Abstraction 40
5.5. Requirements for Advertising Links and Nodes 40
5.6. Addressing Considerations 40

6. Building on Existing Protocols 41
6.1. BGP-LS 41
6.2. IGPs 41
6.3. RSVP-TE 41
7. Applicability to Optical Domains and Networks 42
8. Modeling the User-to-Network Interface 43
9. Abstraction in L3VPN Multi-AS Environments 47
10. Scoping Future Work 49
10.1. Not Solving the Internet 49
10.2. Working With "Related" Domains 49
10.3. Not Finding Optimal Paths in All Situations 49
10.4. Not Breaking Existing Protocols 49
10.5. Sanity and Scaling 49
11. Manageability Considerations 50
12. IANA Considerations 50
13. Security Considerations 50
14. Acknowledgements 50
15. References 50
15.1. Informative References 50
Authors' Addresses 54
Contributors 55

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 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.

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 and architecture for the exchange of TE information between interconnected TE domains in support of end-to-end TE path establishment. 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.

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. 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 formulaically from the metrics and 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. That is, 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) which is helpful especially in determining a path to a

destination that lies in an unknown domain, or may be qualified by TE attributes and TE metrics such as hop count, available bandwidth, delay, shared risk, etc.

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 Interior Gateway Protocol (IGP) areas and Autonomous Systems (ASes).

1.1.5. Aggregation

The concept of aggregation is discussed in Section 3.6. 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.

1.1.6. Abstraction

Section 5.3 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 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.

1.1.7. 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 of abstract links are provided in Section 5.3.1.

1.1.8. Abstraction Layer Network

The abstraction layer network is introduced in Section 5.3.2. It may be seen as a brokerage layer network between one or more server networks and one or more client network. The abstraction layer network is the collection of abstract links that provide potential connectivity across the server network(s) 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 resources more flexible, the abstract links might not all extend from edge to edge, but might offer connectivity between server 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), that 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 sub-optimal path, or even fail to make a path available.

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 (see Section 4.2) may be used to alleviate this situation, but do not lead to rapid setup or guaranteed optimality. Furthermore RSVP signalling creates state in the network that is immediately removed by the crankback procedure. Frequent events of such a kind impact scalability in a non-deterministic manner.

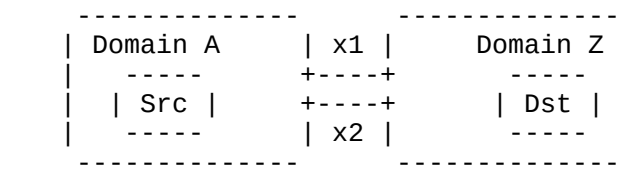

Figure 1 : Peer Networks

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 though Domain D, Domain A must be sensitive to the TE connectivity available through each of Domains C and D, as well the TE connectivity from each of interconnections x4 and x5 to Dst within Domain Z.

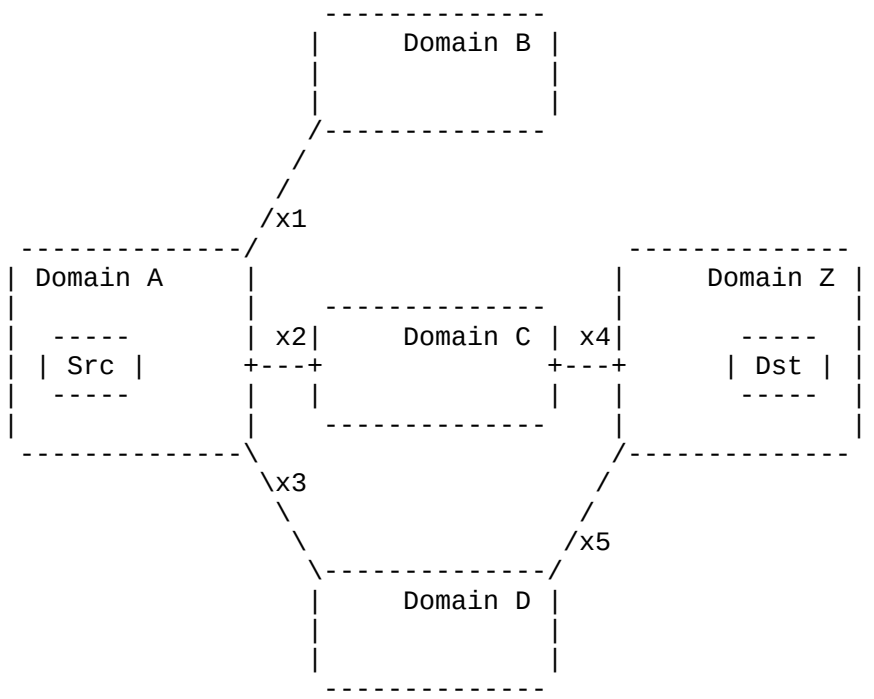

Figure 2 : Peer Networks in a Mesh

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.

2.1.1. Where is the Destination?

A variation of the problems expressed in Section 2.1 arises when the source domain (Domain A in both figures) does not know where the

destination is located. That is, when the domain in which the destination node is located is not known to the source domain.

This is most easily seen in consideration of Figure 2 where the decision about which interconnection to select needs to be based on building a path toward the destination domain. Yet this can only be achieved if it is known in which domain the destination node lies, or at least if there is some indication in which direction the destination lies. This function is obviously provided in IP networks by inter-domain routing [RFC4271].

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.

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

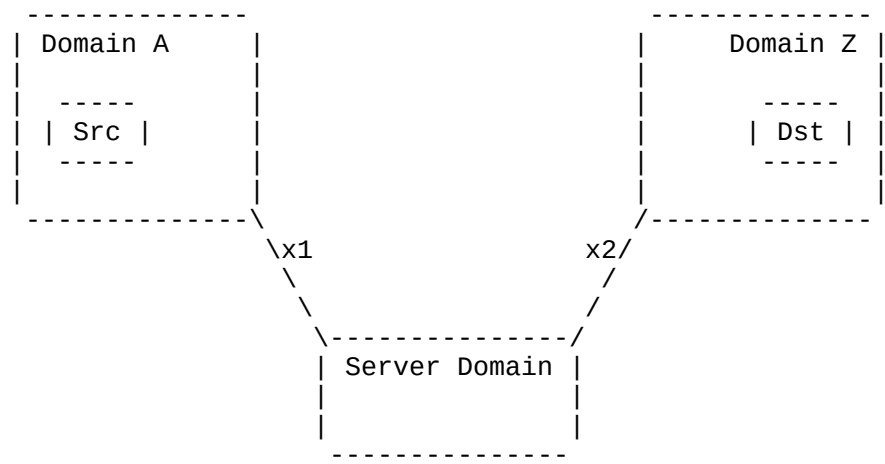

Figure 3 : Client-Server Networks

The second class of use case of 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 layer 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, the provider network.

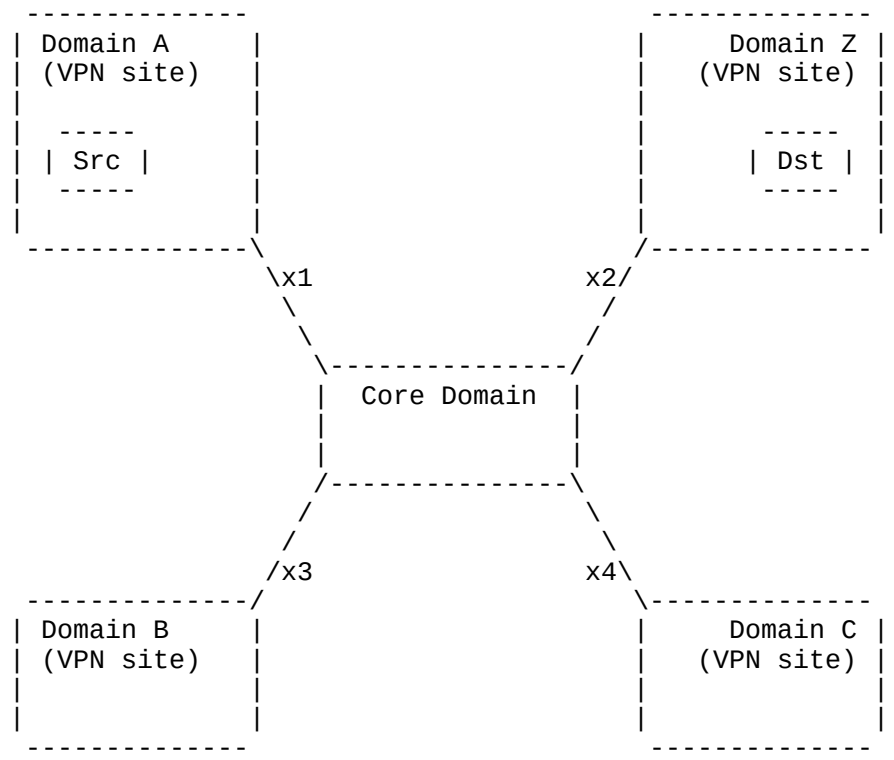

Figure 4 : A Virtual Private Network

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

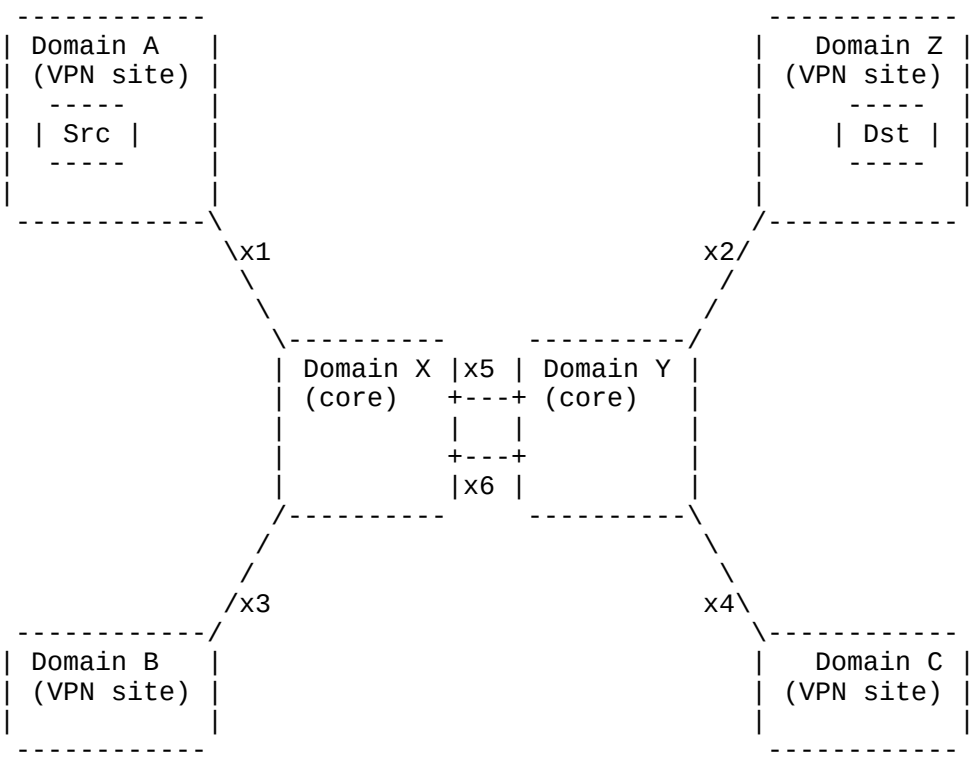

Figure 5 : A VPN Supported Over Multiple Server Domains

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 domain is attached to more than one server domain, or has two points of attachment to a server domain. Figure 6 shows an example of this for a VPN.

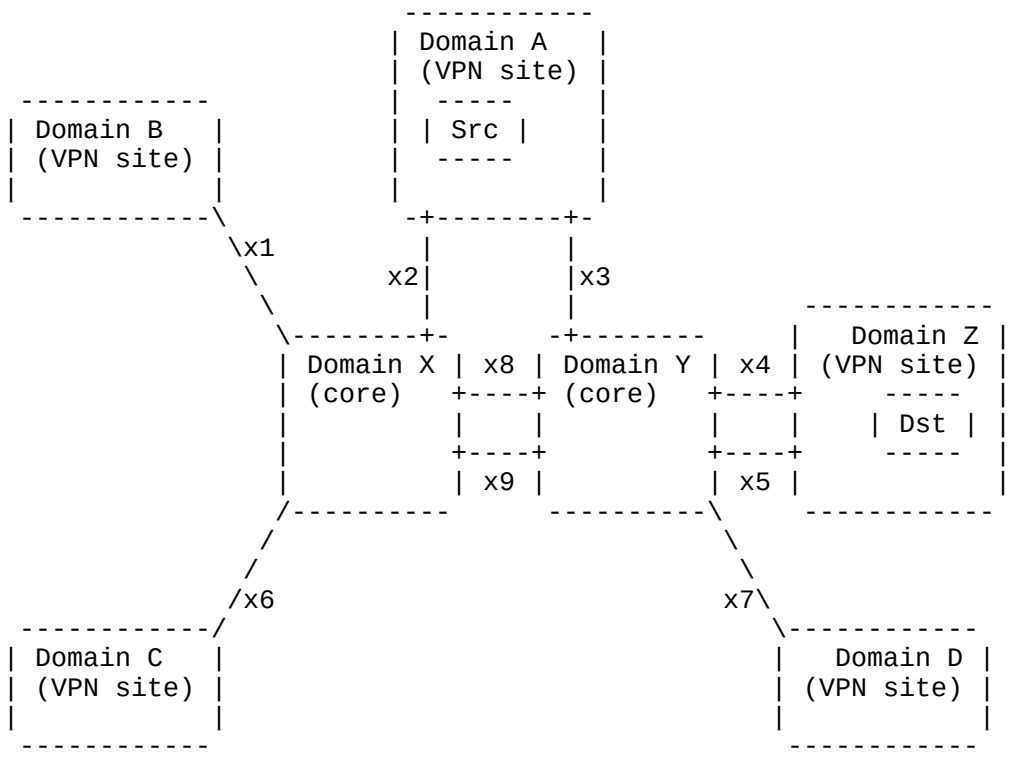

Figure 6 : Dual-Homing in a Virtual Private Network

2.4. Requesting Connectivity

This 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 requested to establish a set of LSPs to provide client layer connectivity. In the dynamic case, the client 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 (such as 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 requested 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 some

intermediary broker such as the Virtual Network Topology Manager discussed in Section 4.6.

The trigger that causes the request to the server layer is also flexible. It could be that the client layer discovers a pressing need for server layer resources (such as the desire to provision an end-to-end connection in the client layer, 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 resources are under the administrative control of the operator or the server layer network and are only used to support a client layer network in ways that the server layer 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 layer. In the case of some server networks, even the bandwidth component of a basic provisioning request is superfluous because the server layer has no facility to vary bandwidth, but 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 layer 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 my 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 for the server layer 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

layer to construct protection services over the server layer network, for example by establishing virtual 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 to follow same path in the server network. This may be requested, for example, to create a bundled or aggregated link in the client layer where each component of the client layer composite link is required to have the same server layer properties (metrics, delay, etc.) and the same failure characteristics.

- Concurrent Provisioning enables the inter-related connections requests described in the previous two bullets to be enacted through a single, compound service request.

- Service Resilience requests the server layer to provide connectivity for which the server layer 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 was 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.

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. Use of Existing Protocol Mechanisms

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]. TE information may be exported from a domain (for example, northbound) using link state extensions to BGP [I-D.ietf-idr-ls-distribution].

It is desirable that a solution to the problem described in this document does not require the implementation of a new, network-wide protocol. Instead, 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 Path Computation Elements (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.

3.2. 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.

3.3. Confidentiality

A feature of the policy described in Section 3.3 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 domains described in Section 2.4) 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 information share to just that which the computing domain needs to know with the understanding that 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.

3.4. 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 3.2 because such filters might not always be enabled.

3.5. 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 which 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 considerably longer. Thus the information churn may be sustainable.

3.6. Issues of Aggregation

One possible solution to the issues raised in other sub-sections 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 [I-D.ietf-ccamp-general-constraint-encode]).

- 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 other and with what TE attributes.

It is of the nature of aggregation 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 3.5 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 in

reduced, the aggregation algorithm (operating on the routers responsible for sharing information) may be heavily exercised.

3.7. Virtual Network Topology

The terms "virtual topology" and "virtual network topology" have become overloaded in a relatively short time. We draw on [RFC5212] and [RFC5623] for inspiration to provide a definition for use in this document. Our definition is based on the fact that a topology at the and [RFC5623] for inspiration to provide a definition for use in this document. Our definition is based on the fact that a topology at the client network layer is constructed of nodes and links. Typically, the nodes are routers in the client layer, and the links are data links. However, a layered network provides connectivity through the lower layer as LSPs, and these LSPs can provide links in the client layer. Furthermore, those LSPs may have been established in advance, or might be LSPs that could be set up if required. This leads to the definition:

> A Virtual Network Topology (VNT) is made up of links in a network layer. Those links may be realized as direct data links or as multi-hop connections (LSPs) in a lower network layer. Those underlying LSPs may be established in advance or created on demand.

The creation and management of a VNT requires interaction with management and policy. Activity is needed in both the client and server layer:

- In the server layer, LSPs need to be set up either in advance in response to management instructions or in answer to dynamic requests subject to policy considerations.

- In the server layer, evaluation of available TE resources can lead to the announcement of potential connectivity (i.e., LSPs that could be set up on demand).

- In the client layer, connectivity (lower layer LSPs or potential LSPs) needs to be announced in the IGP as a normal TE link. Such links may or may not be made available to IP routing: but, they are never made available to IP routing until fully instantiated.

- In the client layer, requests to establish lower layer LSPs need to be made either when links supported by potential LSPs are about to be used (i.e., when a higher layer LSP is signalled to cross the link, the setup of the lower layer LSP is triggered), or when the client layer determines it needs more connectivity or capacity.

It is a fundamental of the use of a VNT that there is a policy point

at the lower-layer node responsible for the instantiation of a lower-layer LSP. At the moment that the setup of a lower-layer LSP is triggered, whether from a client-layer management tool or from signaling in the client layer, the server layer must be able to apply policy to determine whether to actually set up the LSP. Thus, fears that a micro-flow in the client layer might cause the activation of 100G optical resources in the server layer can be completely controlled by the policy of the server layer network's operator (and could even be subject to commercial terms).

These activities require an architecture and protocol elements as well as management components and policy elements.

4. Existing Work

This section briefly summarizes relevant existing work that is used to route TE paths across multiple domains.

4.1. Per-Domain Path Computation

The per-domain mechanism of 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 of 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 the 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 set-up to date.

4.2. 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 RSVP-TE signaling ties up networks resources for partially established LSPs, since network conditions may be in flux, and most particularly since 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 to find the optimal path. (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.)

4.3. 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 (Section 4.1) the source node and each domain entry point is a PCE. 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 has responsibility 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 off-load complex, CPU-intensive, or memory-intensive computation functions from routers in the network. But the use of PCE in this way does not solve any of the problems articulated in Sections 4.1 and 4.2.

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 domain entry point (or source node) to 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 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.

- Hierarchical PCE (H-PCE) [RFC6805] 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 2.1.1. Furthermore, it raises the question of who operates the parent PCE especially in networks where the domains are under different administrative and commercial control.

Further issues and considerations of the use of PCE can be found in [I-D.farrkingel-pce-questions].

4.4. GMPLS UNI and Overlay Networks

[RFC4208] defines the GMPLS User-to-Network Interface (UNI) to present a routing boundary between an overlay network and the core network, i.e. the client-server interface. In the client network, the nodes connected directly to the core 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 cross the core network, and leaves the core network to select the paths and set up the core LSPs. This solution is supplemented by a number of signaling extensions such as [RFC4874], [RFC5553], [I-D.ietf-ccamp-xro-lsp-subobject], [I-D.ietf-ccamp-rsvp-te-srlg-collect], and [I-D.ietf-ccamp-te-metric-recording] to give the edge node more control over the LSP that the core network will set up by exchanging information about core LSPs that have been established and by allowing the edge nodes to supply additional constraints on the core LSPs that are to be set up.

Nevertheless, in this UNI/overlay model, the edge node has limited

information of precisely what LSPs could be set up across the core, and what TE services (such as diverse routes for end-to-end protection, end-to-end bandwidth, etc.) can be supported.

4.5. Layer One VPN

A Layer One VPN (L1VPN) is a service offered by a core layer 1 network to provide layer 1 connectivity (TDM, 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 the connectivity. In the L1VPN context three different service models have been defined classified by the semantics of information exchanged over the customer interface: Management Based, Signaling Based (a.k.a. basic), and Signaling and Routing service model (a.k.a. enhanced).

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 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 core 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 enhanced 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: Overlay Extension, Virtual Node, Virtual Link and Per-VPN service models. All of these represent particular cases of the TE information aggregation and representation.

4.6. VNT Manager and Link Advertisement

As discussed in Section 3.7, operation of a VNT requires policy and management input. In order to handle this, [RFC5623] introduces the concept of the Virtual Network Topology Manager (VNTM). This is a functional component that applies policy to requests from client networks (or agents of the client network, such as a PCE) for the establishment of LSPs in the server network to provide connectivity in the client network.

The VNTM would, in fact, form part of the provisioning path for all

server network LSPs whether they are set up ahead of client network demand or triggered by end-to-end client network LSP signaling.

An important companion to this function is determining how the LSP set up across the server network is made available as a TE link in the client network. Obviously, if the LSP is established using management intervention, the subsequent client network TE link can also be configured manually. However, if the LSP is signaled dynamically 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 a server network that supports more than one client, it will be necessary to indicate which client or clients can use the link. This capability it provided in [RFC6107].

Note that a potential server network LSP that is advertised as a TE link in the client network might to be determined dynamically by the edge nodes. In this case there will need to be some effort to ensure that both ends of the link have the same view of the available TE resources, or else the advertised link will be asymmetrical.

4.7. What Else is Needed and Why?

As can be seen from Sections 4.1 through 4.6, a lot of effort has focused on client-server networks as described in Figure 3. Far less consideration has been given to network peering or the combination of the two use cases.

Various work has been suggested to extend the definition of the UNI such that routing information can be passed across the interface. However, this approach seems to break the architectural concept of network separation that the UNI facilitates.

Other approaches are working toward a flattening of the network with complete visibility into the server networks being made available in the client network. These approaches, while functional, ignore the main reasons for introducing network separation in the first place.

The remainder of this document introduces a new approach based on network abstraction that allows a server network to use its own knowledge of its resources and topology combined with its own policies to determine what edge-to-edge connectivity capabilities it will inform the client networks about.

5. Architectural Concepts

5.1. Basic Components

This section revisits the use cases from Section 2 to present the

basic architectural components that provide connectivity in the peer and client-server cases. These component models can then be used in later sections to enable discussion of a solution architecture.

5.1.1. Peer Interconnection

Figure 7 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-to-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, LSP tunnels that span the transit networks are used to achieve the required connectivity. These transit LSPs form the key building blocks of the end-to-end connectivity.

The transit tunnels can be used as hierarchical LSPs [RFC4206] to carry the end-to-end LSP, or can become stitching segments [RFC5150] of the end-to-end LSP. The transit tunnels B1-B3 and C-C3 can be as an abstract link as discussed in Section 5.3.

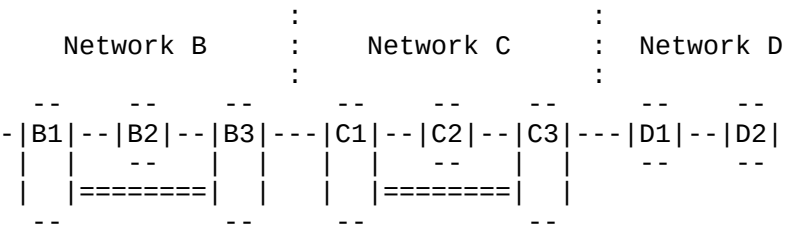

Figure 7 : Architecture for Peering

5.1.2. Client-Server Interconnection

Figure 8 shows the basic architectural concepts for a client-server network. The client network nodes are C1, C2, CE1, CE2, C3, and C4. The core network nodes are CN1, CN2, CN3, and CN4. The interfaces CE1-CN1 and CE2-CN2 are the interfaces between the client and core networks.

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 layer that is supported by a core network LSP.

As shown in the figure, two LSPs are used to achieve the required connectivity. One LSP is set up across the core from CN1 to CN2. This core LSP then supports a three-hop LSP from CE1 to CE2 with its middle hop being the core LSP. It is this LSP that is presented as a link in the client network.

The practicalities of how the CE1-CE2 LSP is carried across the core LSP may depend on the switching and signaling options available in the core network. The LSP may be tunneled down the core LSP using the mechanisms of a hierarchical LSP [RFC4206], or the LSP segments CE1-CN1 and CN2-CE2 may be stitched to the core LSP as described in [RFC5150].

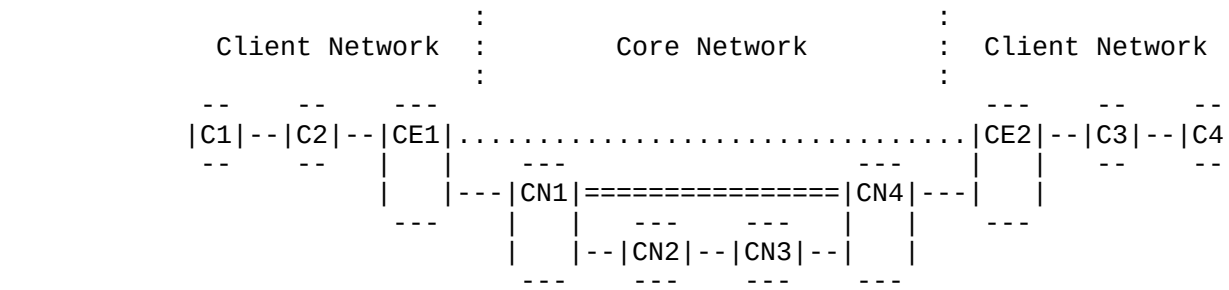

Figure 8 : Architecture for Client-Server Network

5.2. 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) that 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 unqualified (there is a TE path), or may be qualified by TE attributes such as TE metrics, hop count, available bandwidth, delay, shared risk, etc.

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).

5.3. Abstraction not Aggregation

Aggregation is the process of synthesizing from available information. Thus, the virtual node and virtual link models described in Section 3.6 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 3.6), 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.

While abstraction uses available TE information, it is subject to policy and management choices. Thus, not all potential connectivity

will be advertised to each client. 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.

5.3.1. Abstract Links

An abstract link is a measure of the potential to connect a pair of points with certain TE parameters. An abstract link may be realized by an existing LSP, or may represent the possibility of setting up an LSP.

When looking at a network such as that in Figure 8, the link from CN1 to CN4 may be an abstract link. If the LSP has already been set up, it is easy to advertise it as a link with known TE attributes: policy will have been applied in the server network to decide what LSP to set up. If the LSP has not yet been established, the potential for an LSP can be abstracted from the TE information in the core network subject to policy, and the resultant potential LSP can be advertised.

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

5.3.2. The Abstraction Layer Network

Figure 9 introduces the Abstraction Layer Network. This construct separates the client layer resources (nodes C1, C2, C3, and C4, and the corresponding links), and the server layer resources (nodes CN1, CN2, CN3, and CN4 and the corresponding links). Additionally, the architecture introduces an intermediary layer called the Abstraction Layer. The Abstraction Layer contains the client layer edge nodes (C2 and C3), the server layer edge nodes (CN1 and CN4), the client-server links (C2-CN1 and CN4-C3) and the abstract link CN1-CN4.

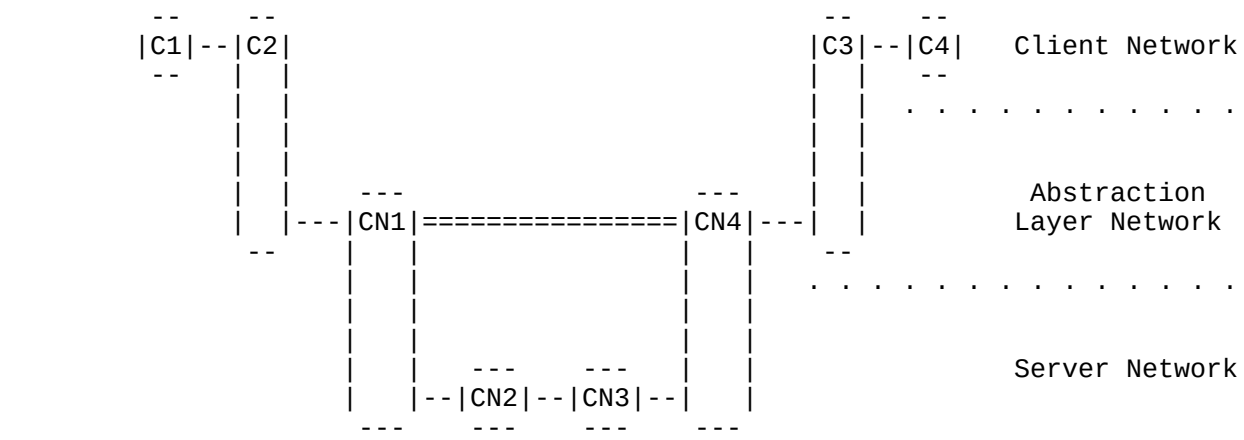

Figure 9 : Architecture for Abstraction Layer Network

The client layer network is able to operate as normal. Connectivity across the network can either be found or not found based on links that appear in the client layer 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 layer.

The server network layer also operates as normal. LSPs across the server layer 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 which 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.

When the client layer 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 asks the server layer to realise the abstract link CN1-CN4. The server

layer provisions the LSP CN1-CN2-CN3-CN4 and makes the LSP available as a hierarchical LSP to turn the abstract link into a link that can be used in the client network. The Abstraction Layer can then set up an LSP C2-CN1-CN4-C3 using stitching or tunneling, and make the LSP available as a virtual link in the client network.

Sections 5.3.3 and 5.3.4 show how this model is used to satisfy the requirements for connectivity in client-server networks and in peer networks.

5.3.2.1. Nodes in the Abstraction Layer Network

Figure 9 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 10, 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 10, 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.

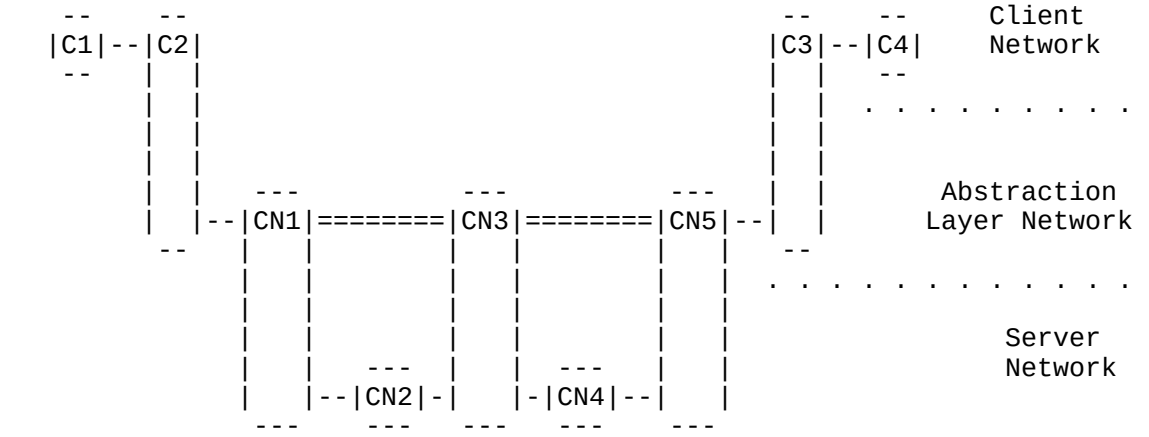

Figure 10 : Abstraction Layer Network with Additional Node

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 3.6. While 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. It may be noted that recent work on limited cross-connect capabilities such as exist in asymmetrical switches could be used to represent the limitations in an Abstract Node [I-D.ietf-ccamp-general-constraint-encode], [I-D.ietf-ccamp-gmpls-general-constraints-ospf-te].

5.3.3. Abstraction in Client-Server Networks

Section 5.3.2 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 11 shows a multi-layer network comprising client nodes (labeled as Cn for n= 0 to 9) and server nodes (labeled as Sn for n = 1 to 9).

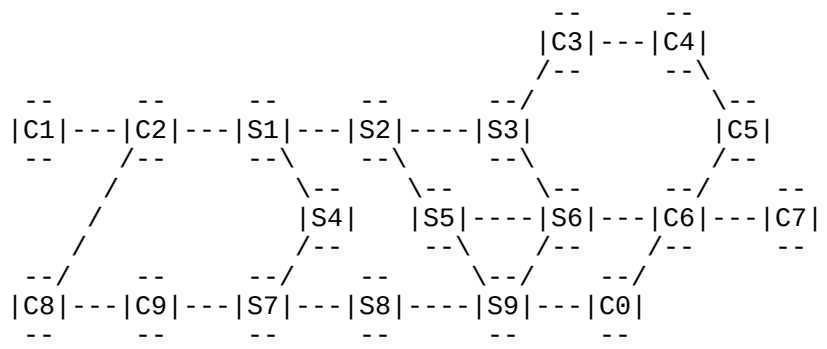

Figure 11 : An example Multi-Layer Network

If the network in Figure 11 is operated as separate client and server networks then the client layer topology will appear as shown in Figure 12. 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).

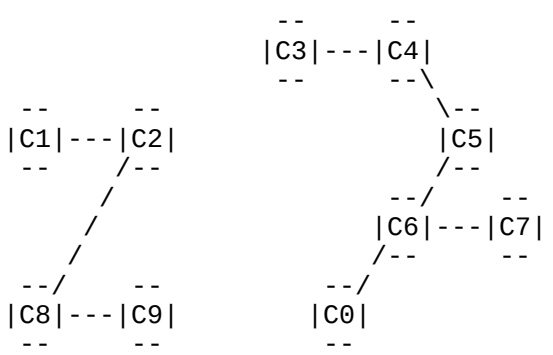

Figure 12 : Client Layer Topology Showing Partitioned Network

For reference, Figure 13 shows the corresponding server layer topology.

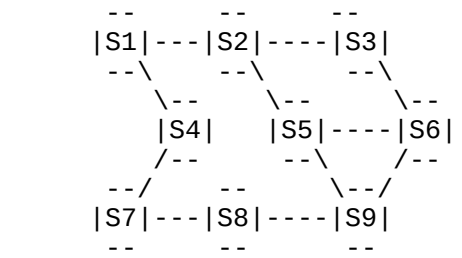

Figure 13 : Server Layer Topology

Operating on the TED for the server layer, a management entity or a software component may apply policy and consider what abstract links it might offer for use by the client layer. 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 layer, 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 14. As can be seen, two of the links must share part of a path (S1-S9 must share with either S1-S3 or with 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.

That would mean that when both S1-S3 and S7-S9 are realized as links carrying Abstraction Layer LSPs, the link S1-S9 can no longer be used.

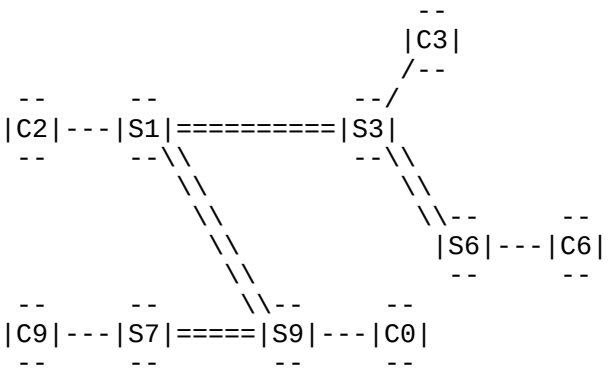

Figure 14 : Abstraction Layer Network with Abstract Links

The separate IGP instance running in the Abstraction Layer Network mean that this topology is visible at the edge nodes (C2, C3, C6, C9, and C0) as well as at a PCE if one is present.

Now the client layer is able to make requests to the Abstraction Layer Network to provide connectivity. In our example, it requests that C2 is connected to C3 and that C2 is 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 Abstraction Layer Network may find that the necessary underlying LSPs (S1-S2-S3 and S1-S2-S5-S9) have not been established since it is not a requirement that an abstract link be backed up by a real LSP. In this case, S1 computes the paths of the underlying LSPs and signals them.

5. Once the serve layer LSPs have been established, S1 can continue

to signal the Abstraction Layer LSPs either using the server layer LSPs as tunnels or as stitching segments.

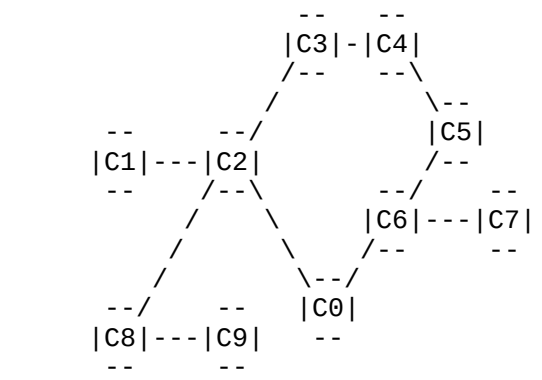

Figure 15 : Connected Client Layer Network with Additional Links

6. Finally, once the Abstraction Layer LSPs have been set up, the client layer can be informed and can start to advertise the new TE links C2-C3 and C2-C0. The resulting client layer topology is shown in Figure 15.

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

5.3.3.1 Macro Shared Risk Link Groups

Network links often share fate with one or more other links. That is, a scenario that may cause a links 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 layer LSPs that are used to create the links in the client network.

For example, looking at the example used in Section 5.3.3 and considering the two abstract links S1-S3 and S1-S9 there is no way for the client layer to know whether the 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

layer 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 mutli-layer network that equates to advertising in the client layer the server layer 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, only the links that actually links in the server layer need to be advertised rather than every link that potentially shares resources. This saving is possible because the abstract links are formulated on behalf of the server layer 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 layer (S1-S4-S7-S8-S9). It is would be possible for the client layer request for connectivity C2-C0 to request that the path be maximally disjoint from the path C2-C3. While nothing can be done about the shared link C2-S1, the Abstraction Layer could request that the server layer instantiate the link S1-S9 to be diverse from the link S1-S3, and this request could be honored if the server layer policy allows.

5.3.3.2 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 layer 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 clients 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 layer network.

5.3.3.3 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 client network and Abstraction Layer Network should be one-to-one.

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

5.3.4. Abstraction in Peer Networks

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 7). Figure 16 shows a simple example of three peer networks (A, B, and C) each comprising a few nodes.

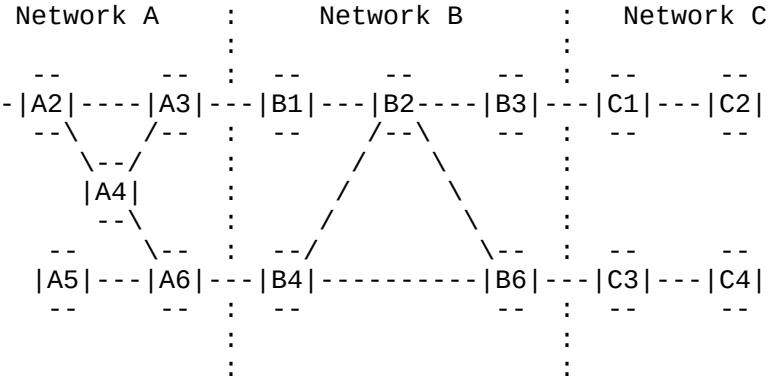

Figure 16 : A Network Comprising Three Peer Networks

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 the reachability across transit peer networks and instantiate 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 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.

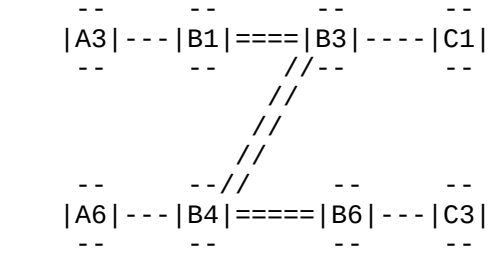

Figure 17 : Abstraction Layer Network for the Peer Network Example

Using a process similar to that described in Section 5.3.3, Network A can request connectivity to Network C and the abstract links can be instantiated as tunnels across the transit network, and edge-to-edge LSPs can be set up to join the two networks. Furthermore, if Network C is partitioned, reachability information can be exchanged to allow Network A to select the correct edge-to-edge LSP as shown in Figure 18.

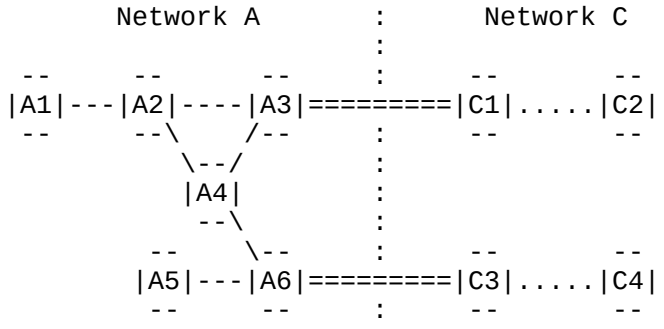

Figure 18 : Tunnel Connections to Network C with TE Reachability

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 7 and 9.

5.4. Considerations for Dynamic Abstraction

<TBD>

5.5. 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 piggy-backed on an existing routing protocol instance, or use a new instance (or even a new protocol). Clearly, the information exchanged is only that which has been created as part of the abstraction function according to policy.

It should be noted that in some cases Abstract Link enablement is on-demand and all that is advertised in the topology for the Abstraction Layer Network is the potential for an Abstract Link to be set up. In this case we may ponder how the routing protocol will advertise topology information over a link that is not yet established. In other words, there must be a communication channel between the participating nodes so that the routing protocol messages can flow. The answer is that control plane connectivity 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.

5.6. Addressing Considerations

[Editor Note: Need to work up some text on addressing to cover the case of each domain having a different (potentially overlapping) address space and the need for inter-domain addressing. In fact, this should be quite simple but needs discussion.
Also needed is a discussion of the case where two client networks share an abstraction network (section 5.3.3.2). How does addressing work here? Are there security issues?]

6. Building on Existing Protocols

This section 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 re-use is preferred over the invention of new protocols or additional protocol extensions as mentioned in Section 3.1.

6.1. BGP-LS

BGP-LS is a set of extensions to BGP described in [I-D.ietf-idr-ls-distribution]. It's purpose is to announce topology information from one network to a "north-bound" 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.

6.2. 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) or as dual instances on the same address space. This means that either IGP could probably be used as the routing protocol in the Abstraction Layer Network.

6.3. RSVP-TE

RSVP-TE signaling can be used to set up traffic engineered LSPs to serve as hierarchical LSPs in the core network providing Abstract Links for the Abstraction Layer Network as described in [RFC4206]. Similarly, the CE-to-CE LSP tunnel across the Abstraction Layer Network can be established using RSVP-TE without 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.

7. Applicability to Optical Domains and Networks

Many optical networks are arranged 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, the 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.

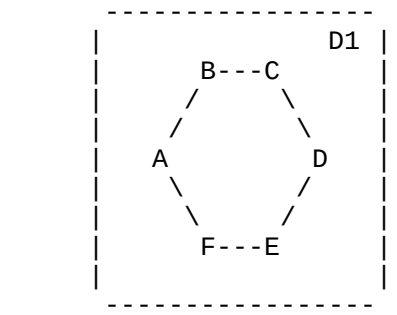

Figure 19 : A Simple Optical Domain

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.

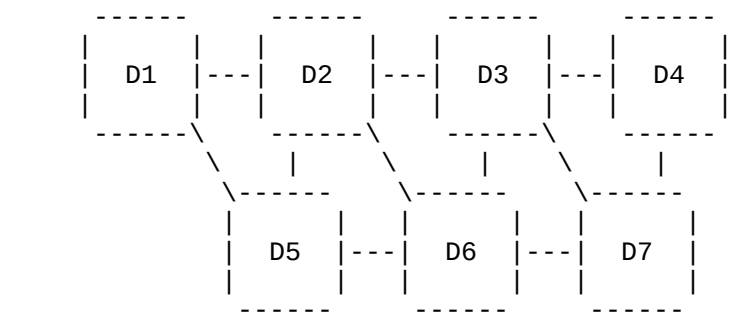

Figure 20 : A Simple Optical Domain

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.

o 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 PCE.

  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 PCE may be able to solve this type of network, the effort involved may be considerable for more than a small collection of domains.

o 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 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 used private AS numbers [RFC6996] of which there are plenty.

  Then, as suggested in the Area-Based model, hierarchical PCE 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 that other model, the scalability of the hierarchical PCE approach must be questioned.

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 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 has visibility of the connectivity of the whole network.

  While 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 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 7.1 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. The document even notes that some forms of aggregation may be appropriate. Thus, we can apply the following

simplifications to the mechanisms defined in RFC 6827:

- 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 RFC 6827, but without needing any of the protocol extensions that that document defines. Thus, using the terminology and concepts already established, the problem may 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.

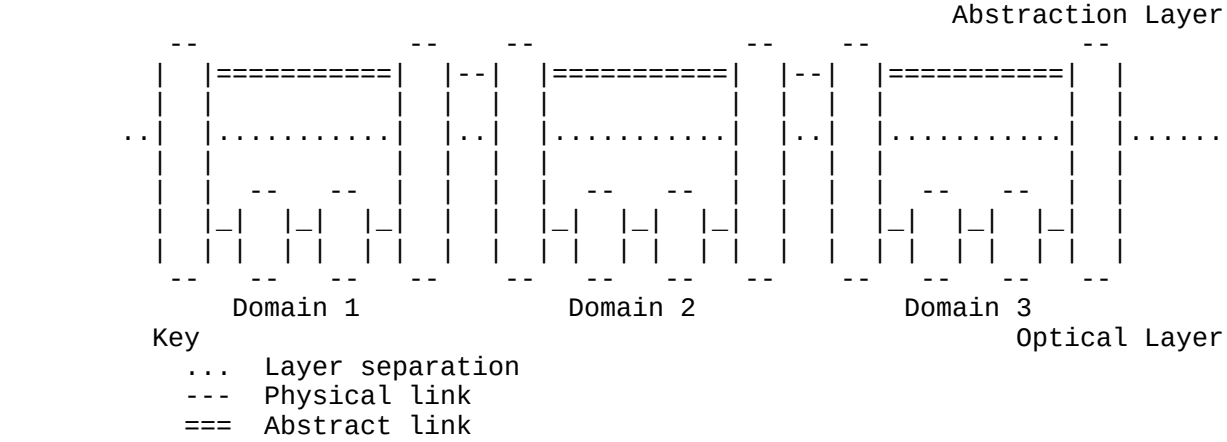

Figure 21 : The Optical Network Implemented Through the Abstraction Layer Network

8. Modeling the User-to-Network Interface

The User-to-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 can be seen 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 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, and the client edge nodes taking the role of UNI Client-side (UNI-C) and the server edge nodes acting as the UNI Network-side (UNI-N) nodes.

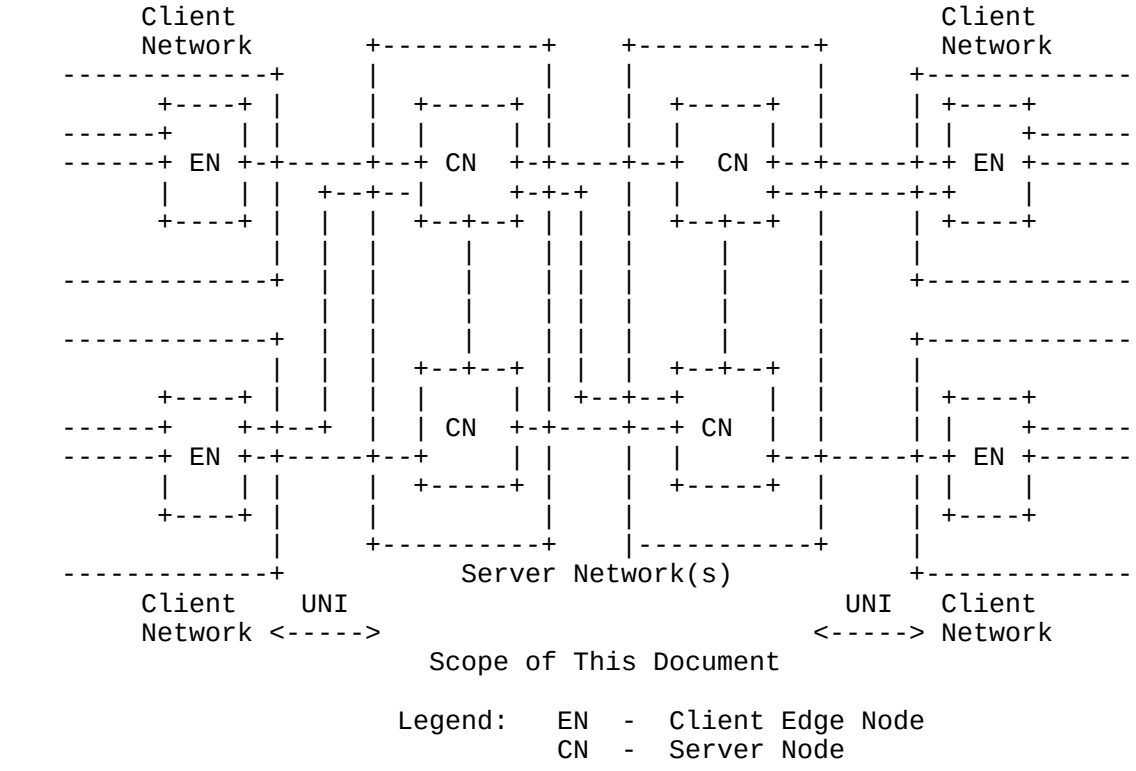

Figure 22 : Ethernet UNI Reference Model

An issue that is often raised concerns how a dual-homed client 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) is part of the Abstraction Layer Network and can see sufficient topology information to make these decisions. There is, therefore, no need to enhance the signaling protocols at the GMPLS UNI nor to add routing exchanges at the UNI.

9. Abstraction in L3VPN Multi-AS Environments

Serving layer-3 VPNs (L3PVNs) 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 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.

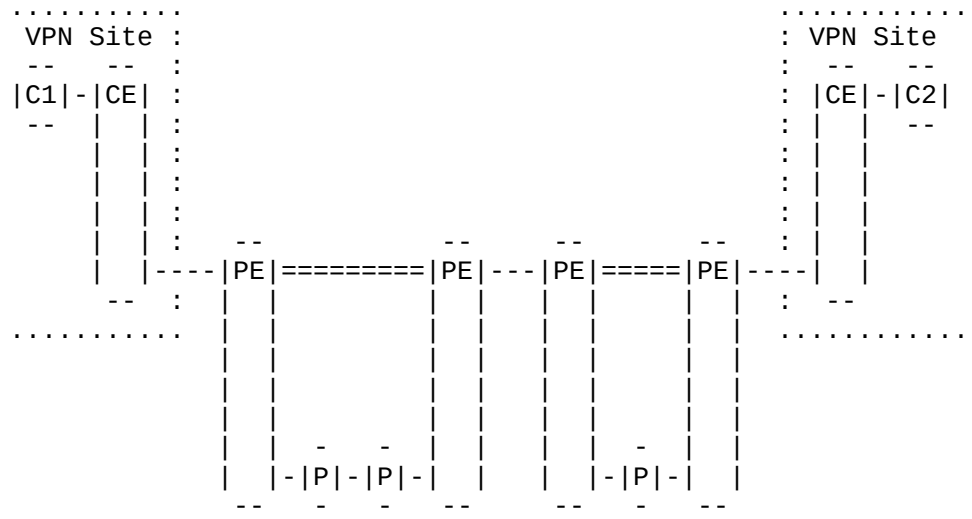

Figure 23 : The Abstraction Layer Network for a Multi-AS VPN

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.

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 8, 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.

10. Scoping Future Work

The section is provided to help guide the work on this problem and to ensure that oceans are not knowingly boiled.

10.1. Not Solving 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.

10.2. Working With "Related" Domains

Subsequent to Section 10.1, 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.

10.3. 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.

10.4. Not Breaking Existing Protocols

It is a clear objective of this work to not break existing protocols. The Internet relies on the stability of a few key routing protocols, and so it is critical that any new work must not make these protocols brittle or unstable.

10.5. 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.

11. Manageability Considerations

<TBD>

12. IANA Considerations

This document makes no requests for IANA action. The RFC Editor may safely remove this section.

13. Security Considerations

<TBD>

14. Acknowledgements

Thanks to Igor Bryskin for useful discussions in the early stages of this work.

Thanks to Gert Grammel for discussions on the extent of aggregation in abstract nodes and links.

Thanks to Deborah Brungard, Dieter Beller, and Vallinayakam Somasundaram 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 5.3.3 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 [I-D.beeram-ccamp-gmpls-enni] for which the authors of this document express their thanks.

15. References

15.1. Informative References

[G.8080] ITU-T, "Architecture for the automatically switched optical network (ASON)", Recommendation G.8080.

[I-D.beeram-ccamp-gmpls-enni]
Bryskin, I., Beeram, V. P., Drake, J. et al., "Generalized Multiprotocol Label Switching (GMPLS) External Network Network Interface (E-NNI): Virtual Link Enhancements for the Overlay Model", draft-beeram-ccamp-gmpls-enni, work in progress.

[I-D.farrkingel-pce-questions]
Farrel, A., and D. King, "Unanswered Questions in the Path Computation Element Architecture", draft-farrkingel-pce-questions, work in progress.

[I-D.ietf-ccamp-general-constraint-encode]
Bernstein, G., Lee, Y., Li, D., and Imajuku, W., "General Network Element Constraint Encoding for GMPLS Controlled Networks", draft-ietf-ccamp-general-constraint-encode, work in progress.

[I-D.ietf-ccamp-gmpls-general-constraints-ospf-te]
Zhang, F., Lee, Y,. Han, J, Bernstein, G., and Xu, Y., "OSPF-TE Extensions for General Network Element Constraints", draft-ietf-ccamp-gmpls-general-constraints-ospf-te, work in progress.

[I-D.ietf-ccamp-rsvp-te-srlg-collect]
Zhang, F. (Ed.) and O. Gonzalez de Dios (Ed.), "RSVP-TE Extensions for Collecting SRLG Information", draft-ietf-ccamp-rsvp-te-srlg-collect, work in progress.

[I-D.ietf-ccamp-te-metric-recording]
Z. Ali, et al., "Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) extension for recording TE Metric of a Label Switched Path," draft-ali-ccamp-te-metric-recording, work in progress.

[I-D.ietf-ccamp-xro-lsp-subobject]
Z. Ali, et al., "Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) LSP Route Diversity using Exclude Routes," draft-ali-ccamp-xro-lsp-subobject, work in progress.

[I-D.ietf-idr-ls-distribution]
Gredler, H., Medved, J., Previdi, S., Farrel, A., and Ray, S., "North-Bound Distribution of Link-State and TE Information using BGP", draft-ietf-idr-ls-distribution, work in progress.

[RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and McManus, J., "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC3473] L. Berger, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RC 3473, January 2003.

[RFC3630] Katz, D., Kompella, and K., Yeung, D., "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.

[RFC3945] Mannie, E., (Ed.), "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, October 2004.

[RFC4105] Le Roux, J.-L., Vasseur, J.-P., and Boyle, J., "Requirements for Inter-Area MPLS Traffic Engineering", RFC 4105, June 2005.

[RFC4202] Kompella, K. and Y. Rekhter, "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4202, October 2005.

[RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, October 2005.

[RFC4208] Swallow, G., Drake, J., Ishimatsu, H., and Y. Rekhter, "User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model", RFC 4208, October 2005.

[RFC4216] Zhang, R., and Vasseur, J.-P., "MPLS Inter-Autonomous System (AS) Traffic Engineering (TE) Requirements", RFC 4216, November 2005.

[RFC4271] Rekhter, Y., Li, T., and Hares, S., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[RFC4726] Farrel, A., Vasseur, J.-P., and Ayyangar, A., "A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering", RFC 4726, November 2006.

[RFC4847] T. Takeda (Ed.), "Framework and Requirements for Layer 1 Virtual Private Networks," RFC 4847, April 2007.

[RFC4874] Lee, CY., Farrel, A., and S. De Cnodder, "Exclude Routes - Extension to Resource ReserVation Protocol-Traffic Engineering (RSVP-TE)", RFC 4874, April 2007.

[RFC4920] Farrel, A., Satyanarayana, A., Iwata, A., Fujita, N., and Ash, G., "Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE", RFC 4920, July 2007.

[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, February 2008.

[RFC5152] Vasseur, JP., Ayyangar, A., and Zhang, R., "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, February 2008.

[RFC5195] Ould-Brahim, H., Fedyk, D., and Y. Rekhter, "BGP-Based Auto-Discovery for Layer-1 VPNs", RFC 5195, June 2008.

[RFC5212] Shiomoto, K., Papadimitriou, D., Le Roux, JL., Vigoureux, M., and D. Brungard, "Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)", RFC 5212, July 2008.

[RFC5251] Fedyk, D., Rekhter, Y., Papadimitriou, D., Rabbat, R., and L. Berger, "Layer 1 VPN Basic Mode", RFC 5251, July 2008.

[RFC5252] Bryskin, I. and L. Berger, "OSPF-Based Layer 1 VPN Auto-Discovery", RFC 5252, July 2008.

[RFC5305] Li, T., and Smit, H., "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

[RFC5440] Vasseur, JP. and Le Roux, JL., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[RFC5441] Vasseur, JP., Zhang, R., Bitar, N, and Le Roux, JL., "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, April 2009.

[RFC5523] L. Berger, "OSPFv3-Based Layer 1 VPN Auto-Discovery", RFC 5523, April 2009.

[RFC5553] Farrel, A., Bradford, R., and JP. Vasseur, "Resource Reservation Protocol (RSVP) Extensions for Path Key Support", RFC 5553, May 2009.

[RFC5623] Oki, E., Takeda, T., Le Roux, JL., and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, September 2009.

[RFC6005] Nerger, L., and D. Fedyk, "Generalized MPLS (GMPLS) Support for Metro Ethernet Forum and G.8011 User Network Interface (UNI)", RFC 6005, October 2010.

[RFC6107] Shiomoto, K., and A. Farrel, "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, February 2011.

[RFC6805] King, D., and A. Farrel, "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, November 2012.

[RFC6827] Malis, A., Lindem, A., and D. Papadimitriou, "Automatically Switched Optical Network (ASON) Routing for OSPFv2 Protocols", RFC 6827, January 2013.

[RFC6996] J. Mitchell, "Autonomous System (AS) Reservation for Private Use", BCP 6, RFC 6996, July 2013.

Authors' Addresses

Adrian Farrel
Juniper Networks
EMail: adrian@olddog.co.uk

John Drake
Juniper Networks
EMail: jdrake@juniper.net

Nabil Bitar
Verizon
40 Sylvan Road
Waltham, MA 02145
EMail: nabil.bitar@verizon.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

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