Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 15, 2013

A. Farrel
J. Drake
Juniper Networks

N. Bitar
Verizon Networks

G. Swallow
Cisco Systems, Inc.

D. Ceccarelli
Ericsson

July 15, 2013

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

draft-farrel-interconnected-te-info-exchange-01.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.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute

working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

Table of Contents

1. Introduction

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

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 such as TE metrics, hop count, available bandwidth, delay, shared risk, etc.

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 on 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 allieviate this situation, but do not lead to rapid setup or guaranteed optimality.

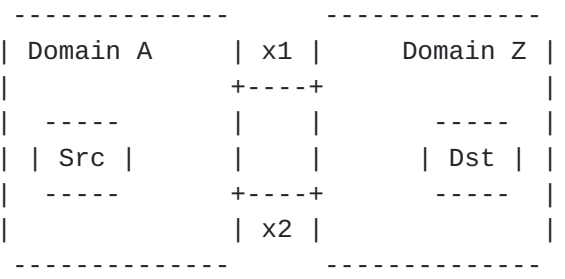


Figure 1 : Peer Networks

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.

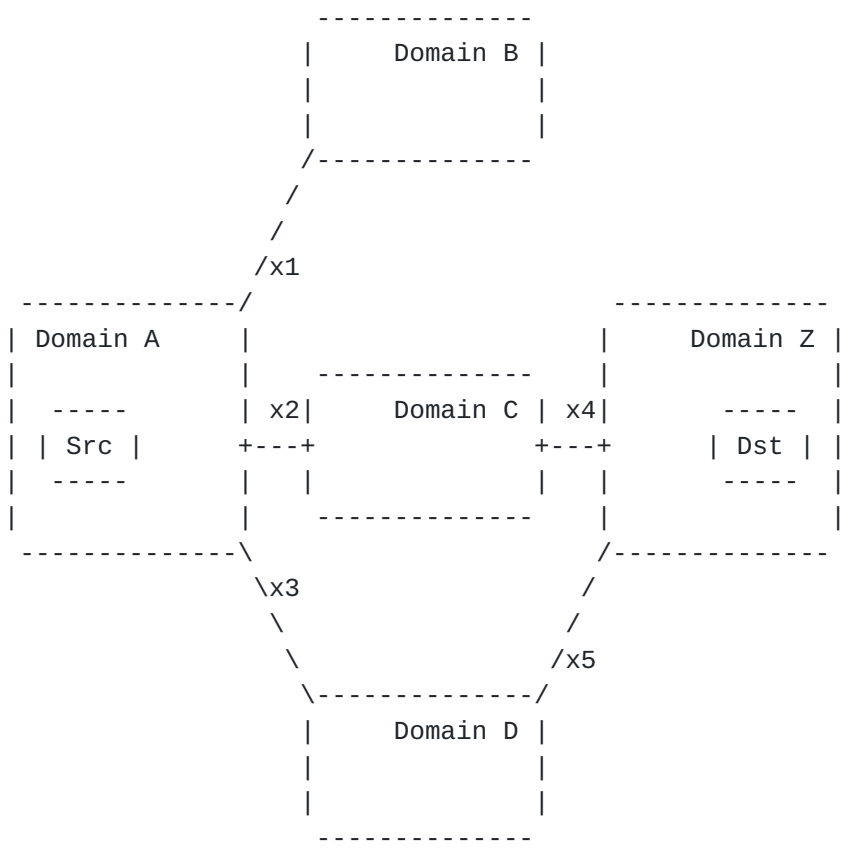


Figure 2 : Peer Networks in a Mesh

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 specific use cases relate to the client-server relationship between networks. These use cases have sometimes been referred to as overlay networks.

The first case, shown is 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.

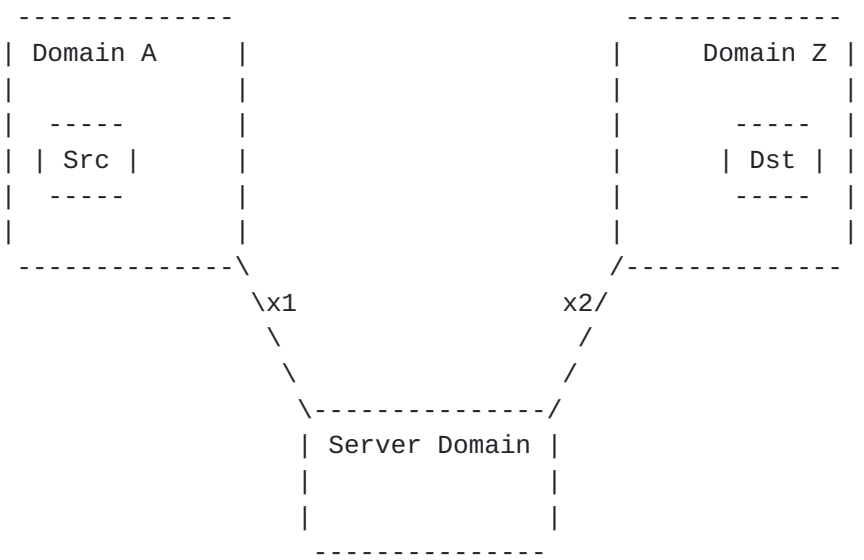


Figure 3 : Client-Server Networks

The second use case relating to client-server networking is for Virtual Private Networks (VPNs). In this case, as opposed to the former one, it is assumed that the client network has a different address space than that of the server 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.

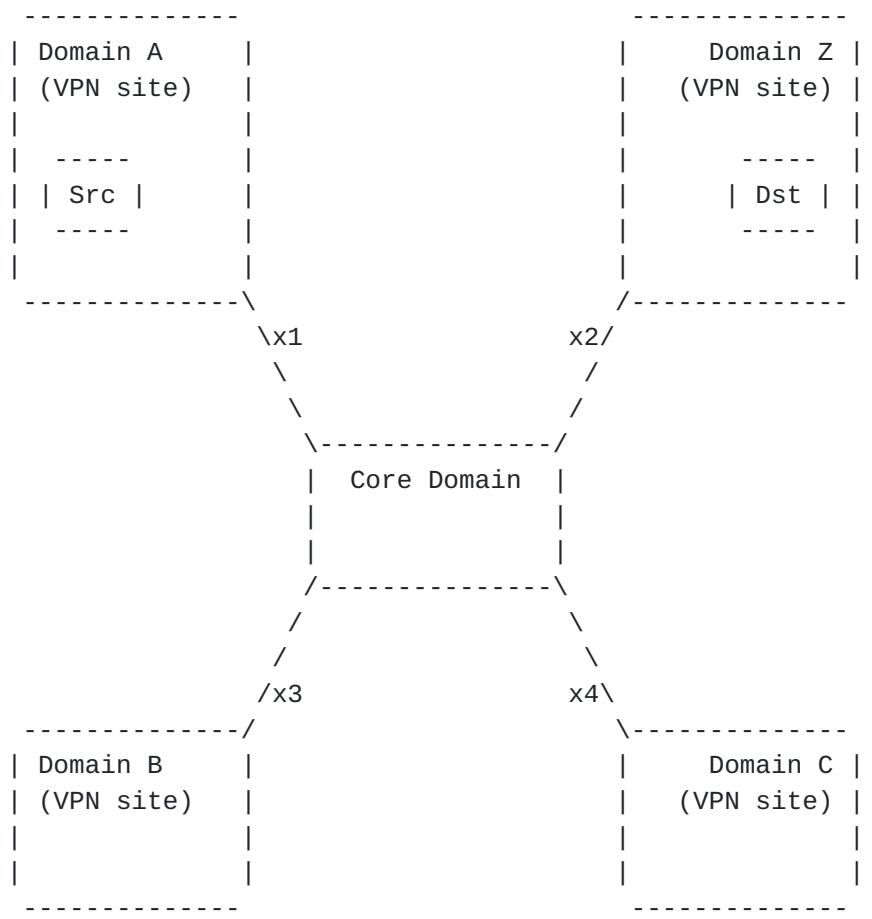


Figure 4 : A Virtual Private Network

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.

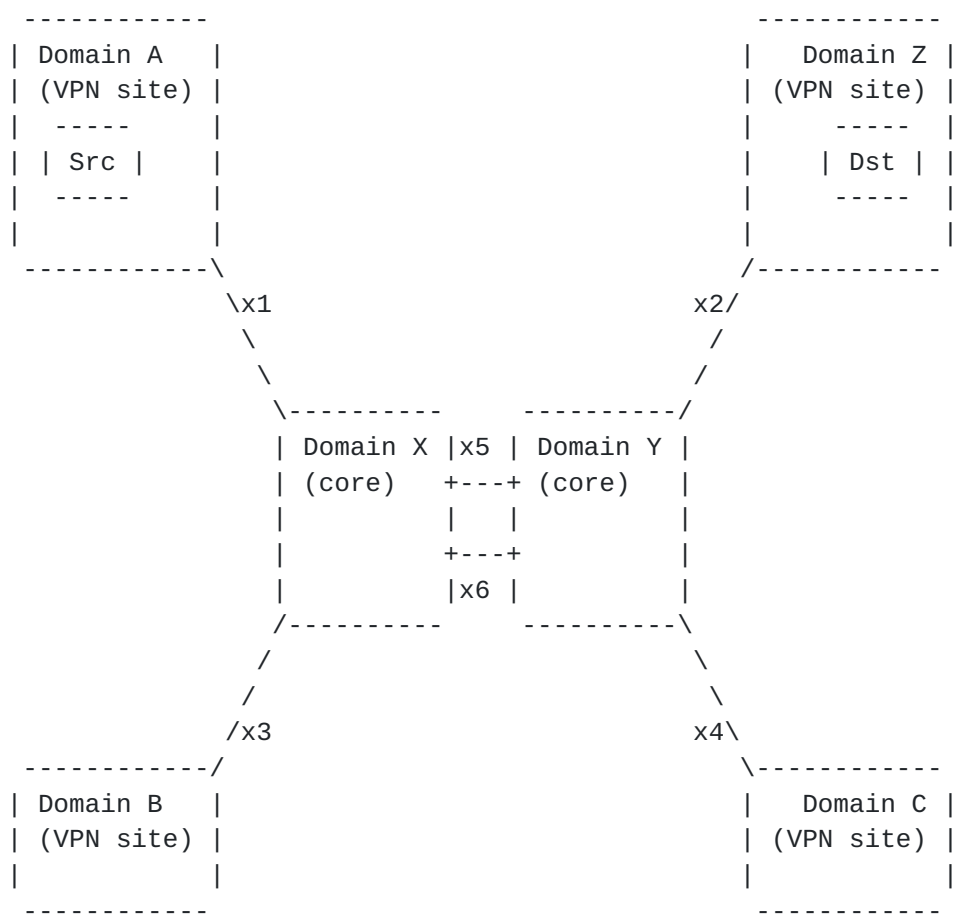


Figure 5 : A VPN Supported Over Multiple Server Domains

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.

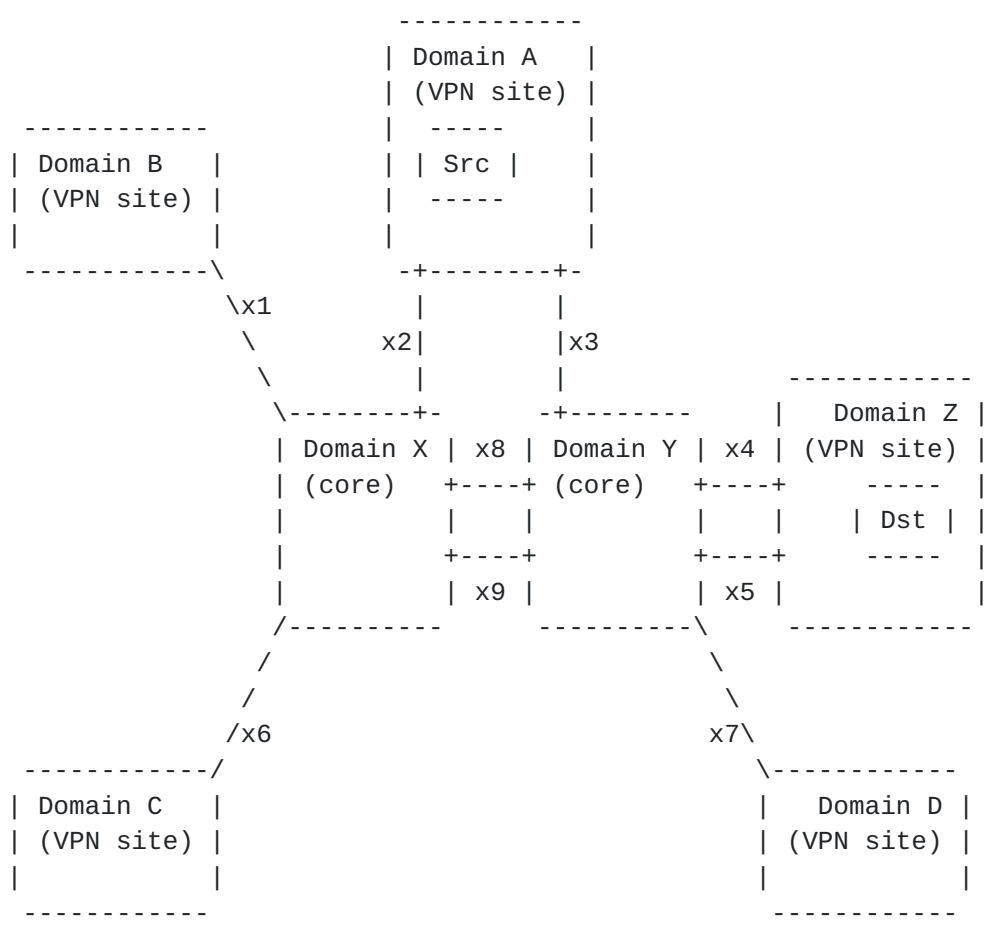


Figure 6 : Dual-Homing in a Virtual Private Network

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 path computation in one domain to make informed choices about the exit point 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 routers 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.

- 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

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 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 point of 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 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 cranback 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 emd-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 PCE. 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.

[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 [RFC5553], [I-D.ietf-ccamp-xro-lsp-subobject], 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. 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 VNT Manager 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.

<TBD>

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.

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 and may be advertised to the source network to enable it to determine the right way to route a TE connection to the destination.

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. Two different abstraction models may be applied (as described further in Section 5.3): the connection B1-B3

can be expressed as an abstract link; or the network {C1, C2, C3} can be represented as an abstract node.

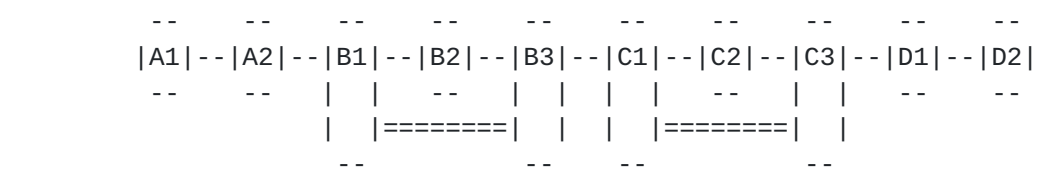


Figure 7 : Architecture for Peering

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 UNIs 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. The three-hop LSP is often called the UNI-LSP, and its middle hop is comprised of the core LSP. It is the UNI-LSP that is presented as a link in the client network.

The practicalities of how the UNI-LSP is carried across the core LSP may depend on the switching and signaling options available in the core network. The UNI-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].

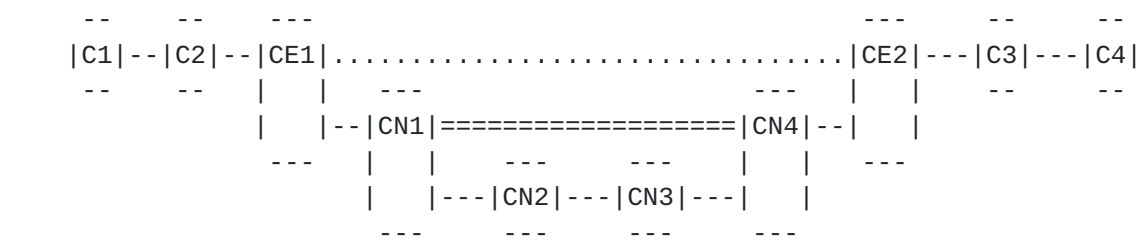


Figure 8 : Architecture for Client-Server Network

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 reaching a specific destination node from an edge 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 is exchanged between networks so that nodes in one network can determine whether they can establish TE paths across or into another network.

5.3. Abstraction not Aggregation

Aggregation is the process of synthesizing from available information. Thus, the virtual node and virtual link models 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 this work 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.

However, when dealing with requested TE parameters that are only a small percentage of the available resources, abstraction is likely to prove more than adequate. For example, when setting up an end-to-end LSP that needs 64 MB bandwidth, an abstraction that offers 100 GB connectivity is unlikely to result in a setup failure.

While abstraction uses available TE information, it will be subject to policy and management choice. 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 client-server network such as that in Figure 8, the link from CE1 to CE2 may be an abstract link. If the LSP has already been set up, it is easy to advertise it into the client layer IGP with known TE attributes. However, if the LSP is not yet established, the potential for an LSP must be abstracted from the TE information in the core network. Since the client nodes (CE1 and CE2) 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

from CN1 to CN4, and CE1 will build on this to generate the abstraction for the UNI connectivity.

5.3.2. Abstract Nodes

<TBD>

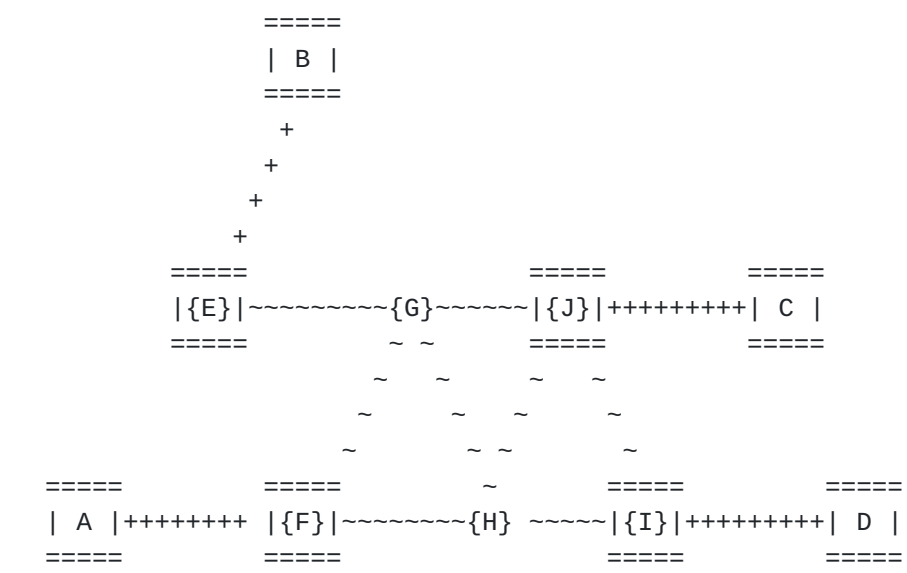
5.3.3. Abstraction in Peer Networks

<TBD>

5.3.4. Abstraction in Client-Server Networks

In client-server networks, path computation and qualification operations occur within a given layer, and hence information about topology and resource availability are required for the specific layer where path computation is being performed, hence the server layer TE-links are of no importance for the client layer network and vice versa; rather, it can be desirable to block their advertisements into the client TE domain by the border nodes.

For example, in the sample hybrid network shown Figure 9, there are many possible ways of connecting client nodes A and C through server network links, however the server network elements have a different switching capability with respect to the client network elements (e.g., LSP vs PSC) and hence they are not part of the topology required by the client layer path computation process.



Key

+++++ client-layer TE link

~~~~~ server-layer TE link

| N | client-layer-only TE node

{N} server-layer-only TE node

|{N}| client-server TE node

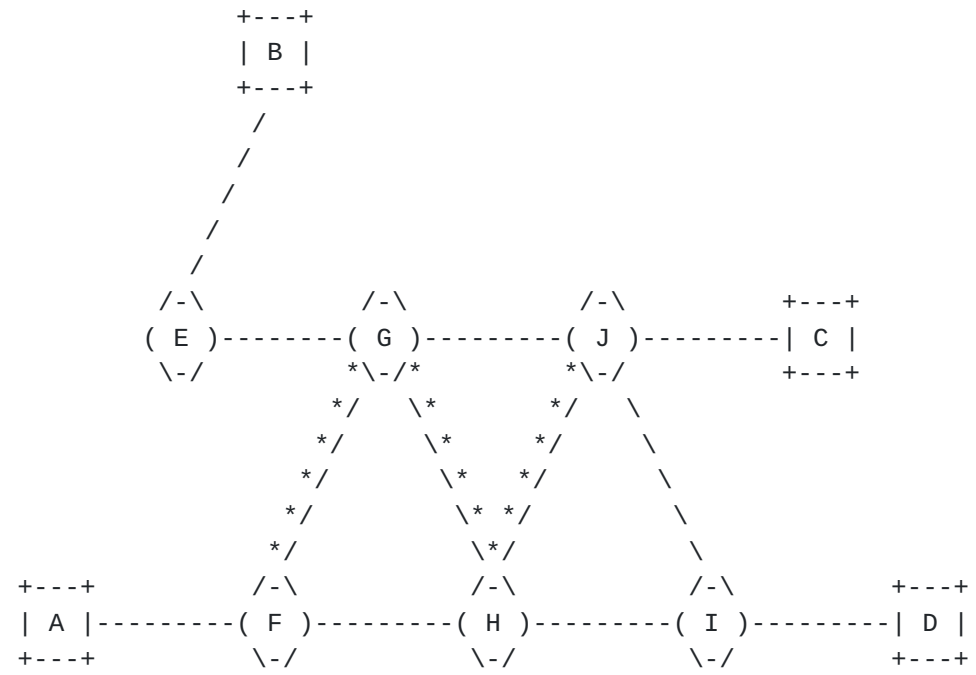
Figure 9 : Client TE Database

In this example, the TE topology associated with the client layer network is indicated by the client layer links (marked with '+') and the client layer nodes (marked without brackets), whereas the TE topology associated with the server layer network is indicated by the server layer TE links and nodes. The client-server nodes are visible in both topologies. The client topology is capable of switching traffic within the client layer, whereas the server topology is capable of switching traffic within the server layer.

In order to be able to compute an end-to-end path between two client layer endpoints, the client topology must be sufficiently augmented to indicate where there are paths through the server topology which can provide connectivity between nodes in the client topology. In
~~~~~

other words, in order for a client to compute path(s) across the server layer network to other clients, the feasible paths across the server layer network should be made available (in terms of TE links and nodes that exist in the client layer network) to all the clients.

With reference to the same sample network in Figure 9, suppose a server-layer service is created through nodes F, G, H, and J as shown in Figure 10.



Key

----- physical link

***** server-layer service

Figure 10 : Physical Topology

The server-layer service is made available in the client layer network as a TE-link with the switching capability of the client layer with the result that the client-layer TE link is as shown in Figure 11.

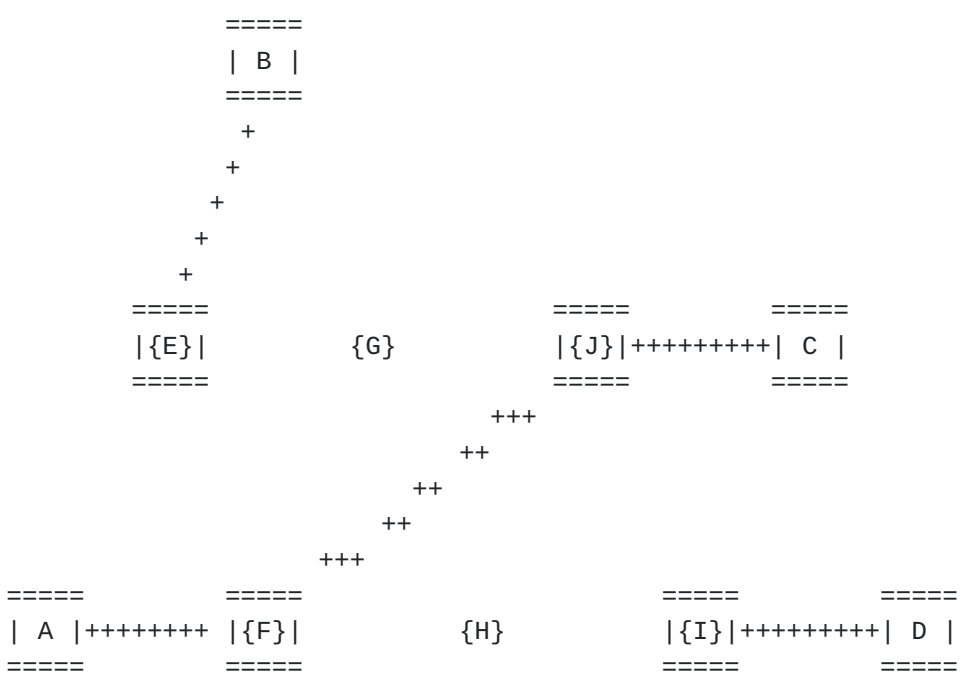


Figure 11 : Client TE Database

As a consequence, A is now able to compute an end-to-end path from A to C. In this example, in order for the TE link to be made available in the client layer network topology, the network resources supporting the underlying server layer LSP are fully committed beforehand. However there are scenarios in which the server layer not only has the capability of providing fixed services (e.g. FOADMs in WDM networks), but is also able to provide connectivity between any pair of network elements (e.g. ROADMs in WDM networks) and hence the network operator might want to commit network resources in the server layer only after the signaling of the service in the client layer.

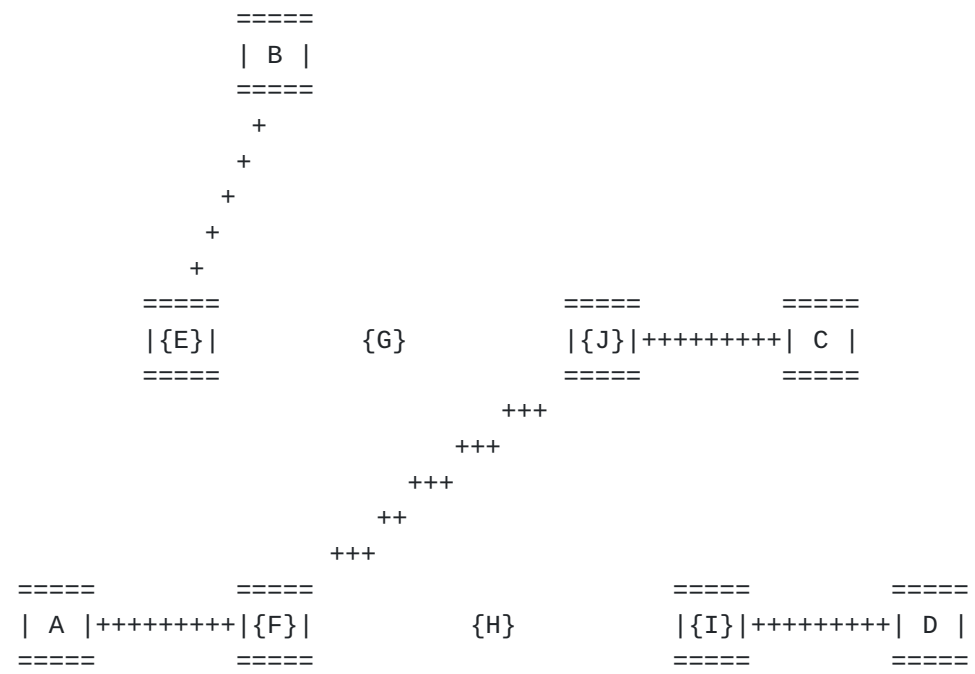
5.3.4.1 Virtual TE links

A Virtual TE Link [RFC4847] is a TE link of the server domain that is advertised into the client domain like a real TE link of the client domain and hence contributes to the buildup of the client layer network topology.

The advertisement includes information about available, but not necessarily reserved/committed, resources in the server layer network necessary to support that TE link. In other words, Virtual TE Links represent specific transport capabilities available in the server layer network, which can support the establishment of LSPs in the client layer network.

Since resources in the server domain are allocated for the exclusive use of each virtual link, it is not possible to share server domain resources among different virtual links.

For example, Figure 12 shows the client-layer view as reflected in the client-layer TE database, while Figure 13 shows the physical topology with a potential server-layer service. The availability of a lambda channel along the path F-G-H-J results in the advertisement by nodes F and J of a Virtual TE Link between F and J into the client layer network topology. With the advertisement of this Virtual TE Link, the path computation function at node A is able to compute an end-to-end path from A to C.

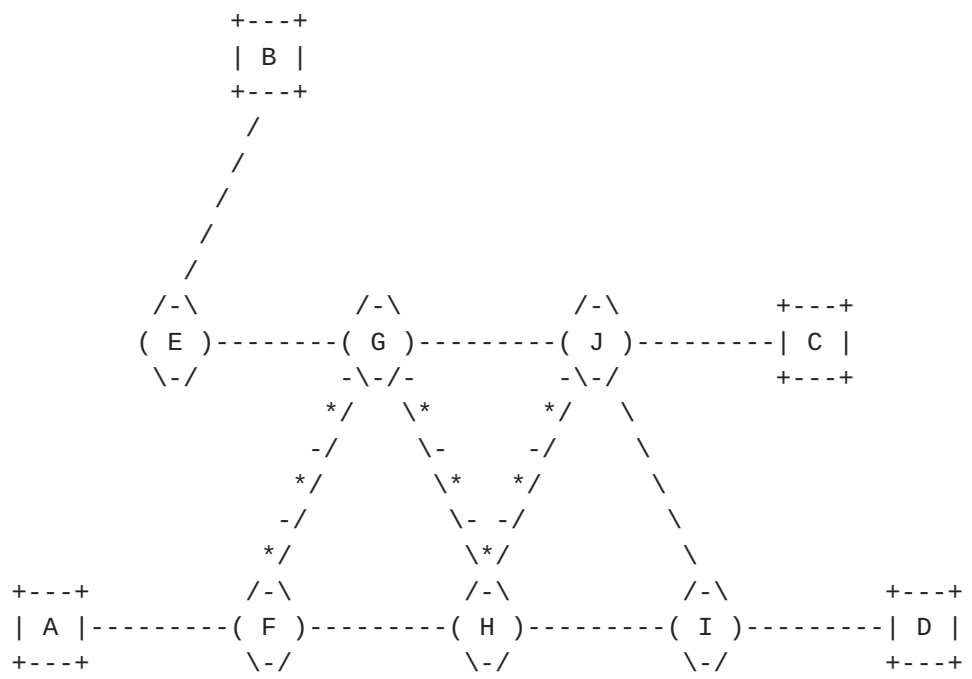


Key

+++++ client-layer TE link

| N | client-layer TE node

Figure 12 : Client-Layer TE Database



Key

--* potential server-layer WDM service

Figure 13 : Physical Topology

Whenever a Virtual TE Link gets selected and signaled in the ERO of a client layer LSP, it ceases temporarily to be "virtual" and transforms into a regular TE link. When this transformation takes place, the clients will notice the change in the advertised available bandwidth of this TE link. Also, all other Virtual TE Links that share in a mutual exclusive way some of server layer resources with the TE link in question should start advertising "zero" available bandwidth. Likewise, the TE network image reverts back to the original form as soon as the last client layer LSP, going through the TE link in question, is released, i.e. Virtual TE Link becomes "virtual" again.

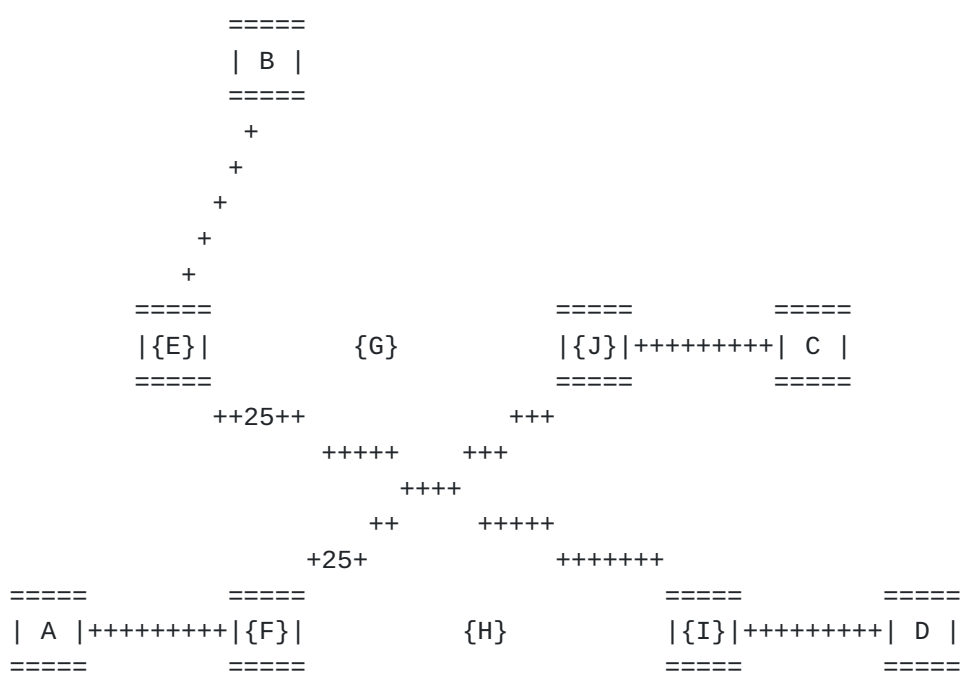
The overlay topology, advertised into the client domain as a set of Virtual TE Links, along with access TE links (the TE links interconnecting client network elements with the network domain) makes up the topology that in the overlay model allows for the client domain path computation function to compute end-to-end paths interconnecting client network elements across the network domain.

5.3.4.2 Macro SRLG

Virtual TE Links often share fate with one or more other Virtual TE links. This is owing to the fact that the LSPs in the server domain (established or potential) can traverse the same physical resources (i.e., link and/or node) and a failure of any of such resources would cause the simultaneous failure of all those LSP and consequently of all the virtual links they support. If diverse end-to-end paths for client domain LSPs are needed (e.g., in a protection scheme), the fate-sharing information of the Virtual TE Links needs to be taken into account during the path computation phase. The standard way of addressing this problem is via the concept of Shared Risk Link Group (SRLG) [RFC4202].

A "traditional" SRLG (per [RFC4202]) represents a shared physical network resource, upon which normal function of a link depends. Such SRLGs can also be referred to as physical SRLGs. Zero, one or more physical SRLGs could be identified and advertised for every TE link in a given layer network. There is a scalability issue with physical SRLGs in multi-layer environments. For example, if a server layer LSP serves a client layer link, every server layer link and node traversed by the LSP must be considered as a separate SRLG. The number of server layer SRLGs to be advertised to client layer per TE link is directly proportional to the number of hops traversed by the underlying server layer LSP.

The notion of Macro SRLGs addresses this scaling problem. Macro SRLGs have the same protocol format as their physical counterparts and can be assigned automatically for each TE link that is advertised into the client layer network supported by an underlying server layer LSP (instantiated or otherwise). A Macro SRLG represents a shared path segment that is traversed by two or more of the underlying server layer LSPs. Each shared path segment can be viewed as a set of shared server layer resources. The actual procedure for deriving the Macro SRLGs is beyond the scope of this document.



Key

+25+ client-layer TE link with SRLG ID "25"

Figure 14 : Client-Layer TE Database with SRLG

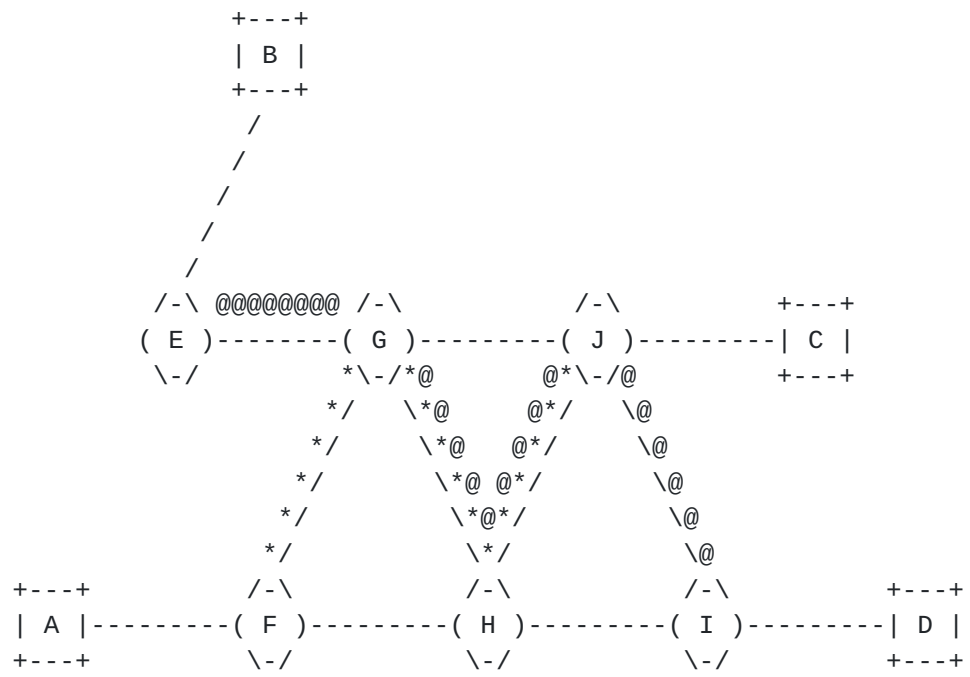


Figure 15 : Macro SRLGs

5.4. Considerations for Dynamic Abstraction

<TBD>

5.5. Requirements for Advertising Abstracted Links and Nodes

<TBD>

6. Building on Existing Protocols

6.1. BGP

<TBD>

6.1.1. Current Uses of BGP

<TBD>

6.1.1.1. IP Reachability

<TBD>

6.1.1.2. VPNs

<TBD>

6.1.1.3. Link State Distribution

<TBD>

6.1.2. Potential Extensions to BGP for TE Reachability

<TBD>

6.2. IGPs

<TBD>

6.3. RSVP-TE

<TBD>

7. Scoping Future Work

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

7.1. 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.

7.2. Working With "Related" Domains

Subsequent to Section 7.1, 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.

7.3. 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.

7.4. Sanity and Scaling

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

8. Manageability Considerations

<TBD>

9. IANA Considerations

This document makes no requests for IANA action.

10. Security Considerations

<TBD>

11. Acknowledgements

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

Thanks to Vishnu Pavan Beeram and Igor Bryskin for useful discussions.

Text in Section 5.3.4 is freely adapted from the work of Igo 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.

12. References

12.1. Normative References

12.2. Informative References

[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-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-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-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.

[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.

[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.

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