Network Working Group
Request for Comments: 5439
Category: Informational

S. Yasukawa
NTT
A. Farrel
Old Dog Consulting
O. Komolafe
Cisco Systems

February 2009

An Analysis of Scaling Issues in MPLS-TE Core Networks

Status of This Memo

Copyright Notice

Abstract

Traffic engineered Multiprotocol Label Switching (MPLS-TE) is deployed in providers' core networks. As providers plan to grow these networks, they need to understand whether existing protocols and implementations can support the network sizes that they are planning.

This document presents an analysis of some of the scaling concerns for the number of Label Switching Paths (LSPs) in MPLS-TE core networks, and examines the value of two techniques (LSP hierarchies and multipoint-to-point LSPs) for improving scaling. The intention is to motivate the development of appropriate deployment techniques and protocol extensions to enable the application of MPLS-TE in large networks.

This document only considers the question of achieving scalability for the support of point-to-point MPLS-TE LSPs. Point-to-multipoint MPLS-TE LSPs are for future study.

Table of Contents

1. Introduction

Network operators and service providers are examining scaling issues as they look to deploy ever-larger traffic engineered Multiprotocol Label Switching (MPLS-TE) networks. Concerns have been raised about the number of Label Switched Paths (LSPs) that need to be supported at the edge and at the core of the network. The impact on control plane and management plane resources threatens to outweigh the benefits and popularity of MPLS-TE, while the physical limitations of the routers may constrain the deployment options.

Historically, it has been assumed that all MPLS-TE scaling issues can be addressed using hierarchical LSP [RFC4206]. However, analysis shows that the improvement gained by LSP hierarchies is not as significant in all topologies and at all points in the network as might have been presumed. Further, additional management issues are introduced to determine the end-points of the hierarchical LSPs and to operate them. Although this does not invalidate the benefits of LSP hierarchies, it does indicate that additional techniques may be desirable in order to fully scale MPLS-TE networks.

This document examines the scaling properties of two generic MPLS-TE network topologies and investigates the benefits of two scaling techniques.

1.1. Overview

Physical topology scaling concerns are addressed by building networks that are not fully meshed. Network topologies tend to be meshed in the core but tree-shaped at the edges, giving rise to a snowflake design. Alternatively, the core may be more of a ladder shape with tree-shaped edges.

MPLS-TE, however, establishes a logical full mesh between all edge points in the network, and this is where the scaling problems arise since the structure of the network tends to focus a large number of LSPs within the core of the network.

This document presents two generic network topologies (the snowflake and the ladder) and attempts to parameterize the networks by making some generalities. It introduces terminology for the different scaling parameters and examines how many LSPs might be required to be carried within the core of a network.

Two techniques (hierarchical LSPs and multipoint-to-point LSPs) are introduced and an examination is made of the scaling benefits that they offer as well as of some of the concerns with using these techniques.

Of necessity, this document makes many generalizations. Not least among these is a set of assumptions about the symmetry and connectivity of the physical network. It is hoped that these generalizations will not impinge on the usefulness of the overview of the scaling properties that this document attempts to give. Indeed, the symmetry of the example topologies tends to highlight the scaling issues of the different solution models, and this may be useful in exposing the worst case scenarios.

Although protection mechanisms like Fast Reroute (FRR) [RFC4090] are briefly discussed, the main body of this document considers stable network cases. It should be noted that make-before-break re-optimisation after link failure may result in a significant number of 'duplicate' LSPs. This issue is not addressed in this document.

It should also be understood that certain deployment models where separate traffic engineered LSPs are used to provide different services (such as layer 3 Virtual Private Networks (VPNs) [RFC4110] or pseudowires [RFC3985]) or different classes of service [RFC3270] may result in 'duplicate' or 'parallel' LSPs running between any pair of provider edge nodes (PEs). This scaling factor is also not considered in this document, but may be easily applied as a linear factor by the reader.

The operation of security mechanisms in MPLS-TE networks [MPLS-SEC] may have an impact on the ability of the network to scale. For example, they may increase both the size and number of control plane messages. Additionally, they may increase the processing overhead as control plane messages are subject to processing algorithms (such as encryption), and security keys need to be managed. Deployers will need to consider the trade-offs between scaling objectives and security objectives in their networks, and should resist the temptation to respond to a degradation of scaling performance by turning off security techniques that have previously been deemed as necessary. Further analysis of the effects of security measures on scalability are not considered further in this document.

This document is designed to help service providers discover whether existing protocols and implementations can support the network sizes that they are planning. To do this, it presents an analysis of some of the scaling concerns for MPLS-TE core networks and examines the

value of two techniques for improving scaling. This should motivate the development of appropriate deployment techniques and protocol extensions to enable the application of MPLS-TE in large networks.

This document only considers the question of achieving scalability for the support of point-to-point MPLS-TE LSPs. Point-to-multipoint MPLS-TE LSPs are for future study.

1.2. Glossary of Notation

This document applies consistent notation to define various parameters of the networks that are analyzed. These terms are defined as they are introduced throughout the document, but are grouped together here for quick reference. Refer to the full definitions in the text for detailed explanations.

n	A network level. n = 1 is the core of the network. See Section 3 for more details on the definition of a level.
P(n)	A node at level n in the network.
S(n)	The number of nodes at level n. That is, the number of P(n) nodes.
L(n)	The number of LSPs seen by a P(n) node.
X(n)	The number of LSP segment states held by a P(n) node.
M(n)	The number of P(n+1) nodes subtended to a P(n) node.
R	The number of rungs in a ladder network.
E	The number of edge nodes (PEs) subtended below (directly or indirectly) a spar-node in a ladder network.
K	The cost-effectiveness of the network expressed in terms of the ratio of the number of PEs to the number of network nodes.

2. Issues of Concern for Scaling

This section presents some of the issues associated with the support of LSPs at a Label Switching Router (LSR) or within the network. These issues may mean that there is a limit to the number of LSPs that can be supported.

2.1. LSP State

LSP state is the data (information) that must be stored at an LSR in order to maintain an LSP. Here, we refer to the information that is necessary to maintain forwarding plane state and the additional information required when LSPs are established through control plane protocols. While the size of the LSP state is implementation-dependent, it is clear that any implementation will require some data in order to maintain LSP state.

Thus, LSP state becomes a scaling concern because as the number of LSPs at an LSR increases, so the amount of memory required to maintain the LSPs increases in direct proportion. Since the memory capacity of an LSR is limited, there is a related limit placed on the number LSPs that can be supported.

Note that techniques to reduce the memory requirements (such as data compression) may serve to increase the number of LSPs that can be supported, but this will only achieve a moderate multiplier and may significantly decrease the ability to process the state rapidly.

In this document, we define X(n) as "the number of LSP segment states held by a P(n) node." This definition observes that an LSR at the end of an LSP only has to maintain state in one direction (i.e., into the network), while a transit LSR must maintain state in both directions (i.e., toward both ends of the LSP). Furthermore, in multipoint-to-point (MP2P) LSPs (see Section 8), a transit LSR may need to maintain LSP state for one downstream segment (toward the destination) and multiple upstream segments (from multiple sources). That is, we define LSP segment state as the state necessary to maintain an LSP in one direction to one adjacent node.

2.2. Processing Overhead

Depending largely on implementation issues, the number of LSPs passing through an LSR may impact the processing speed for each LSP. For example, control block search times can increase with the number of control blocks to be searched, and even excellent implementations cannot completely mitigate this fact. Thus, since CPU power is constrained in any LSR, there may be a practical limit to the number of LSPs that can be supported.

Further processing overhead considerations depend on issues specific to the control plane protocols, and are discussed in the next section.

2.3. RSVP-TE Implications

Like many connection-oriented signaling protocols, RSVP-TE (Resource Reservation Protocol - Traffic Engineering) requires that state is held within the network in order to maintain LSPs. The impact of this is described in Section 2.1. Note that RSVP-TE requires that separate information is maintained for upstream and downstream relationships, but does not require any specific implementation of that state.

RSVP-TE is a soft-state protocol, which means that protocol messages (refresh messages) must be regularly exchanged between signaling neighbors in order to maintain the state for each LSP that runs between the neighbors. A common period for the transmission (and receipt) of refresh messages is 30 seconds, meaning that each LSR must send and receive one message in each direction (upstream and downstream) every 30 seconds for every LSP it supports. This has the potential to be a significant constraint on the scaling of the network, but various improvements [RFC2961] mean that this refresh processing can be significantly reduced, allowing an implementation to be optimized to remove nearly all concerns about soft-state scaling in a stable network.

Observations of existing implementations indicate that there may be a threshold of around 50,000 LSPs above which an LSR struggles to achieve sufficient processing to maintain LSP state. Although refresh reduction [RFC2961] may substantially improve this situation, it has also been observed that under these circumstances the size of the Srefresh may become very large, and the processing required may still cause significant disruption to an LSR.

Another approach is to increase the refresh time. There is a correlation between the percentage increase in refresh time and the improvement in performance for the LSR. However, it should be noted that RSVP-TE's soft-state nature depends on regular refresh messages; thus, a degree of functionality is lost by increasing the refresh time. This loss may be partially mitigated by the use of the RSVP-TE Hello message, and can also be reduced by the use of various GMPLS extensions [RFC3473], such as the use of [RFC2961] message acknowledgements on all messages.

RSVP-TE also requires that signaling adjacencies be maintained through the use of Hello message exchanges. Although [RFC3209] suggests that Hello messages should be retransmitted every 5 ms, in practice, values of around 3 seconds are more common. Nevertheless, the support of Hello messages can represent a scaling limitation on an RSVP-TE implementation since one message must be sent and received to/from each signaling adjacency every time period. This can impose limits on the number of neighbors (physical or logical) that an LSR supports, but does not impact the number of LSPs that the LSR can handle.

2.4. Management

Another practical concern for the scalability of large MPLS-TE networks is the ability to manage the network. This may be constrained by the available tools, the practicality of managing large numbers of LSPs, and the management protocols in use.

Management tools are software implementations. Although such implementations should not constrain the control plane protocols, it is realistic to appreciate that network deployments will be limited by the scalability of the available tools. In practice, most existing tools have a limit to the number of LSPs that they can support. While a Network Management System (NMS) may be able to support a large number of LSPs, the number that can be supported by an Element Management System (EMS) (or the number supported by an NMS per-LSR) is more likely to be limited.

Similarly, practical constraints may be imposed by the operation of management protocols. For example, an LSR may be swamped by management protocol requests to read information about the LSPs that it supports, and this might impact its ability to sustain those LSPs in the control plane. OAM (Operations, Administration, and Management), alarms, and notifications can further add to the burden placed on an LSR and limit the number of LSPs it can support.

All of these considerations encourage a reduction in the number of LSPs supported within the network and at any particular LSR.

3. Network Topologies

In order to provide some generic analysis of the potential scaling issues for MPLS-TE networks, this document explores two network topology models. These topologies are selected partly because of their symmetry, which makes them more tractable to a formulaic approach, and partly because they represent generalizations of real deployment models. Section 3.3 provides a discussion of the commercial drivers for deployed topologies and gives more analysis of why it is reasonable to consider these two topologies.

The first topology is the snowflake model. In this type of network, only the very core of the network is meshed. The edges of the network are formed as trees rooted in the core.

The second network topology considered is the ladder model. In this type of network, the core of the network is shaped and meshed in the form of a ladder and trees are attached rooted to the edge of the ladder.

The sections that follow examine these topologies in detail in order to parameterize them.

3.1. The Snowflake Network Topology

The snowflake topologies considered in this document are based on a hierarchy of connectivity within the core network. PE nodes have connectivity to P-nodes as shown in Figure 1. There is no direct connectivity between the PEs. Dual homing of PEs to multiple P-nodes is not considered in this document, although it may be a valuable addition to a network configuration.

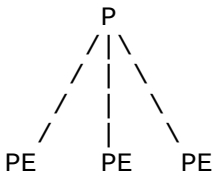

Figure 1 : PE to P-Node Connectivity

The relationship between P-nodes is also structured in a hierarchical way. Thus, as shown in Figure 2, multiple P-nodes at one level are connected to a P-node at a higher level. We number the levels such that level 1 is the top level (top in our figure, and nearest to the core of the network) and level (n) is immediately above level (n+1); we denote a P-node at level n as a P(n).

As with PEs, there is no direct connectivity between P(n+1) nodes. Again, dual homing of P(n+1) nodes to multiple P(n) nodes is not considered in this document, although it may be a valuable addition to a network configuration.

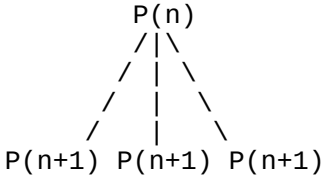

Figure 2 : Relationship between P-Nodes

At the top level, P(1) nodes are connected in a full mesh. In reality, the level 1 part of the network may be slightly less well-connected than this, but assuming a full mesh provides for generality. Thus, the snowflake topology comprises a clique with topologically equivalent trees subtended from each node in the clique.

The key multipliers for scalability are the number of P(1) nodes and the multiplier relationship between P(n) and P(n+1) at each level, down to and including PEs.

We define the multiplier M(n) as the number of P(n+1) nodes at level (n+1) attached to any one P(n). Assume that M(n) is constant for all nodes at level n. Since nodes at the same level are not interconnected (except at the top level), and since each P(n+1) node is connected to precisely one P(n) node, M(n) is one less than the degree of the node at level n (that is, the P(n) node is attached to M(n) nodes at level (n+1) and to 1 node at level (n-1)).

We define S(n) as the number of nodes at level (n).

Thus:

$$S(n) = S(1)*M(1)*M(2)*...*M(n-1)$$

So the number of PEs can be expressed as:

$$S(PE) = S(1)*M(1)*M(2)*...*M(n)$$

where the network has (n) layers of P-nodes.

Thus, we may depict an example snowflake network as shown in Figure 3. In this case:

$$S(1) = 3$$
$$M(1) = 3$$
$$S(2) = S(1)*M(1) = 9$$
$$M(2) = 2$$
$$S(PE) = S(1)*M(1)*M(2) = 18$$

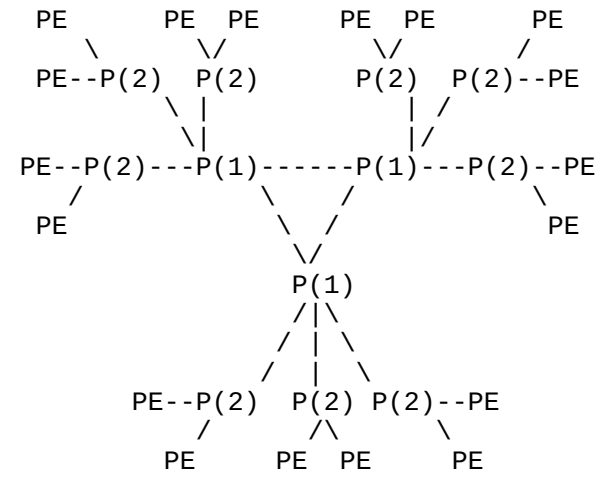

Figure 3 : An Example Snowflake Network

3.2. The Ladder Network Topology

The ladder networks considered in this section are based on an arrangement of routers in the core network that resembles a ladder.

Ladder networks typically have long and thin cores that are arranged as conventional ladders. That is, they have one or more spars connected by rungs. Each node on a spar may have:

- connection to one or more other spars,
- connection to a tree of other core nodes,
- connection to customer nodes.

Figure 4 shows a simplified example of a ladder network. A core of twelve nodes makes up two spars connected by six rungs.

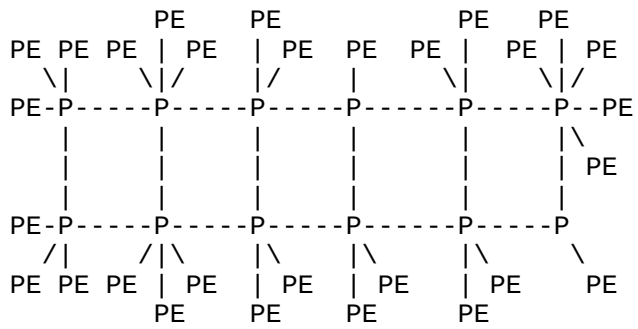

Figure 4 : A Simplified Ladder Network

In practice, not all nodes on a spar (call them spar-nodes) need to have subtended PEs. That is, they can exist simply to give connectivity along the spar to other spar-nodes, or across a rung to another spar. Similarly, the connectivity between spars can be more complex with multiple connections from one spar-node to another spar. Lastly, the network may be complicated by the inclusion of more than two spars (or simplified by reduction to a single spar).

These variables make the ladder network non-trivial to model. For the sake of simplicity, we will make the following restrictions:

- There are precisely two spars in the core network.

- Every spar-node connects to precisely one spar-node on the other spar. That is, each spar-node is attached to precisely one rung.

- Each spar-node connects to either one (end-spar) or two (core-spar) other spar-nodes on the same spar.

- Every spar-node has the same number of PEs subtended. This does not mean that there are no P-nodes subtended to the spar-nodes, but does mean that the edge tree subtended to each spar-node is identical.

From these restrictions, we are able to quantify a ladder network as follows:

R - The number of rungs. That is, the number of spar-nodes on each spar.
S(1) - The number of spar-nodes in the network. S(1)=2*R.
E - The number of subtended edge nodes (PEs) to each spar-node.

The number of rungs may vary considerably. A number less than 3 is unlikely (since that would not be a significantly connected network), and a number greater than 100 seems improbable (because that would represent a very long, thin network).

E can be treated as for the snowflake network. That is, we can consider a number of levels of attachment from P(1) nodes, which are the spar-nodes, through P(i) down to P(n), which are the PEs. Practically, we need to only consider n=2 (PEs attached direct to the spar-nodes) and n=3 (one level of P-nodes between the PEs and the spar-nodes).

Let M(i) be the ratio of P(i) nodes to P(i-1) nodes, i.e., the connectivity between levels of P-node as defined for the snowflake topology. Hence, the number of nodes at any level (n) is:

$$S(n) = S(1)*M(1)*M(2)*...*M(n-1)$$

So the number of PEs subtended to a spar-node is:

$$E = M(1)*M(2)*...*M(n)$$

And the number of PEs can be expressed as:

$$S(PE) = S(1)*M(1)*M(2)*...*M(n)$$
$$= S(1)*E$$

Thus, we may depict an example ladder network as shown in Figure 5. In this case:

R = 5
S(1) = 10
M(1) = 2
S(2) = S(1)*M(1) = 20
M(2) = 2
E = M(1)*M(2) = 4
S(PE) = S(1)*E = 40

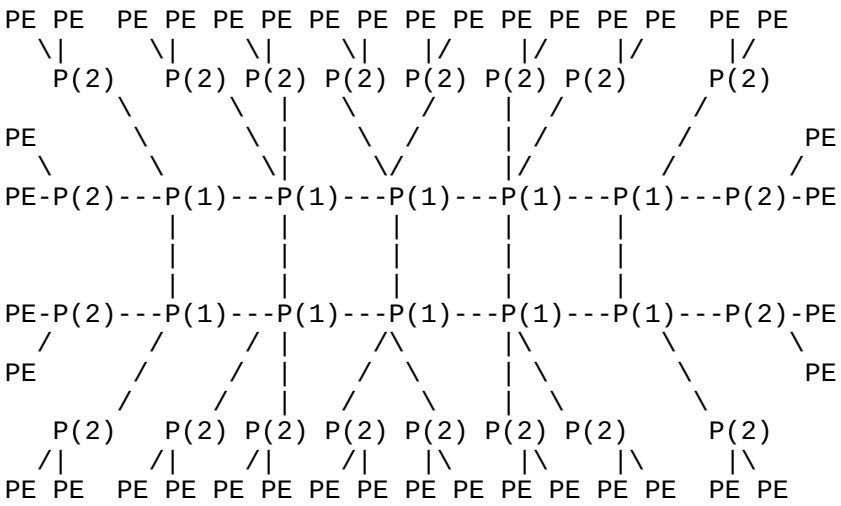

Figure 5 : An Example Ladder Network

3.3. Commercial Drivers for Selected Configurations

It is reasonable to ask why these two particular network topologies have been chosen.

The most important consideration is physical scalability. Each node (Label Switching Router - LSR) is only able to support a limited number of physical interfaces. This necessarily reduces the ability to fully mesh a network and leads to the tree-like structure of the network toward the PEs.

A realistic commercial consideration for an operator is the fact that the only revenue-generating nodes in the network are the PEs. Other nodes are needed only to support connectivity and scalability. Therefore, there is a desire to maximize S(PE) while minimizing the sum of S(n) for all values of (n). This could be achieved by minimizing the number of levels and maximizing the connectivity at each layer, M(n). Ultimately, however, this would produce a network of just interconnected PEs, which is clearly in conflict with the physical scaling situation.

Therefore, the solution calls for a "few" levels with "relatively large" connectivity at each level. We might say that the cost-effectiveness of the network can be stated as:

K = S(PE)/(S(1)+S(2) + ... + S(n)) where n is the level above the PEs

We should observe, however, that this equation may be naive in that the cost of a network is not actually a function of the number of routers (since a router chassis is often free or low cost), but is really a function of the cost of the line cards, which is, itself, a product of the capacity of the line cards. Thus, the relatively high connectivity decreases the cost-effectiveness, while a topology that tends to channel data through a network core tends to demand higher capacity (and so, more expensive) line cards.

A further consideration is the availability of connectivity (usually fibers) between LSR sites. Although it is always possible to lay new fiber, this may not be cost-effective or timely. The physical shape and topography of the country in which the network is laid is likely to be as much of a problem. If the country is 'long and thin', then a ladder network is likely to be used.

This document examines the implications for control plane and data plane scalability of this type of network when MPLS-TE LSPs are used to provide full connectivity between all PEs.

3.4. Other Network Topologies

As explained in Section 1, this document is using two symmetrical and generalized network topologies for simplicity of modelling. In practice, there are two other topological considerations.

a. Multihoming
   It is relatively common for a node at level (n) to be attached to more than one node at level (n-1). This is particularly common at PEs that may be connected to more than one P(n).

b. Meshing within a level
   A level in the network will often include links between P-nodes at the same level, including the possibility of links between PEs. This may result in a network that looks like a series of concentric circles with spokes.

Both of these features are likely to have some impact on the scaling of the networks. However, for the purposes of establishing the ground rules for scaling, this document restricts itself to the consideration of the symmetrical networks described in Sections 2.1 and 2.2. Discussion of other network formats is for future study.

4. Required Network Sizes

An important question for this evaluation and analysis is the size of the network that operators require. How many PEs are required? What ratio of P to PE is acceptable? How many ports do devices have for physical connectivity? What type of MPLS-TE connectivity between PEs is required?

Although presentation of figures for desired network sizes must be treated with caution because history shows that networks grow beyond all projections, it is useful to set some acceptable lower bounds. That is, we can state that we are interested in networks of at least a certain size.

The most important features are:

- The network should have at least 1000 PEs.
- Each pair of PEs should be connected by at least one LSP in each direction.

4.1. Practical Numbers

In practice, reasonable target numbers are as follows.

S(PE) >= 1000
Number of levels is 3. That is: 1, 2, and PE.
M(2) <= 20
M(1) <= 20
S(1) <= 100

5. Scaling in Flat Networks

Before proceeding to examine potential scaling improvements, we need to examine how well the flat networks described in the previous sections scale.

Consider the requirement for a full mesh of LSPs linking all PEs. That is, each PE has an LSP to and from every other LSP. Thus, if there are S(PE) PEs in the network, there are S(PE)*(S(PE) - 1) LSPs.

Define L(n) as the number of LSPs handled by a level (n) LSR.

L(PE) = 2*(S(PE) - 1)

5.1. Snowflake Networks

There are a total of S(PE) PEs in the network and, since each PE establishes an LSP with every other PE, it would be expected that there are S(PE) - 1 LSPs incoming to each PE and the same number of LSPs outgoing from the same PE, giving a total of 2(S(PE) - 1) on the incident link. Hence, in a snowflake topology (see Figure 3), since there are M(2) PEs attached to each P(2) node, it may tempting to think that L(2) (the number of LSPs traversing each P(2) node) is simply 2*(S(PE) - 1)*M(2). However, it should be noted that of the S(PE) - 1 LSPs incoming to each PE, M(2) - 1 originated from nodes attached to the same P(2) node, and so this value would count the LSPs between the M(2) PEs attached to each P(2) node twice: once when outgoing from the M(2) - 1 other nodes and once when incoming into a particular PE.

There are a total of M(2)*(M(2) - 1) LSPs between these M(2) PEs and, since this value is erroneously included twice in 2*(S(PE) - 1)*M(2), the correct value is:

```
L(2) = 2*M(2)*(S(PE) - 1) - M(2)*(M(2) - 1)
     = M(2)*(2*S(PE) - M(2) - 1)
```

An alternative way of looking at this, that proves extensible for the calculation of L(1), is to observe that each PE subtended to a P(2) node has an LSP in each direction to all S(PE) - M(2) PEs in the rest of the system, and there are M(2) such locally subtended PEs; thus, 2*M(2)*(S(PE) - M(2)). Additionally, there are M(2)*(M(2) - 1) LSPs between the locally subtended PEs. So:

```
L(2) = 2*M(2)*(S(PE) - M(2)) + M(2)*(M(2) - 1)
     = M(2)*(2*S(PE) - M(2) - 1)
```

L(1) can be computed in the same way as this second evaluation of L(2). Each PE subtended below a P(1) node has an LSP in each direction to all PEs not below the P(1) node. There are M(1)*M(2) PEs below the P(1) node, so this accounts for 2*M(1)*M(2)*(S(PE) - M(1)*M(2)) LSPs. To this, we need to add the number of LSPs that pass through the P(1) node and that run between the PEs subtended below the P(1). Consider each P(2): it has M(2) PEs, each of which has an LSP going to all of the PEs subtended to the other P(2) nodes subtended to the P(1). There are M(1) - 1 such other P(2) nodes, and so M(2)*(M(1) - 1) other such PEs. So the number of LSPs from the PEs below a P(2) node is M(2)*M(2)*(M(1) - 1). And there are M(1) P(2) nodes below the P(1), giving rise to a total of M(2)*M(2)*M(1)*(M(1) - 1) LSPs. Thus:

```
L(1) = 2*M(1)*M(2)*(S(PE) - M(1)*M(2)) + M(2)*M(2)*M(1)*(M(1) - 1)
     = M(1)*M(2)*(2*S(PE) - M(2)*(M(1) + 1))
```

So, for example, with S(1) = 5, M(1) = 10, and M(2) = 20, we see:

```
S(PE) = 1000
L(PE) = 1998
L(2)  = 39580
L(1)  = 356000
```

Alternatively, with S(1) = 10, M(1) = 10, and M(2) = 20, we see:

```
S(PE) = 2000
L(PE) = 3998
L(2)  = 79580
L(1)  = 756000
```

In both examples, the number of LSPs at the core (P(1)) nodes is probably unacceptably large, even though there are only a relatively modest number of PEs. In fact, L(2) may even be too large in the second example.

5.2. Ladder Networks

In ladder networks, L(PE) remains the same at 2*(S(PE) - 1).

L(2) can be computed using the same mechanism as for the snowflake topology because the subtended tree is the same format. Hence,

```
L(2) = 2*M(2)*(S(PE) - 1) - M(2)*(M(2) - 1)
```

But L(1) requires a different computation because each P(1) not only sees LSPs for the subtended PEs, but is also a transit node for some of the LSPs that cross the core (the core is not fully meshed).

Each P(1) sees:

- all of the LSPs between locally attached PEs,
- less those LSPs between locally attached PEs that can be served exclusively by the attached P(2) nodes,
- all LSPs between locally attached PEs and remote PEs, and
- LSPs in transit that pass through the P(1).

The first three numbers are easily determined and match what we have seen from the snowflake network. They are:

- E*(E-1)
- M(1)*M(2)*(M(2)-1) = E*(M(2) - 1)
- 2*E*E*(S(1) - 1)

The number of LSPs in transit is more complicated to compute. It is simplified by not considering the ends of the ladders but by examining an arbitrary segment of the middle of the ladder, such as shown in Figure 6. We look to compute and generalize the number of LSPs traversing each core link (labeled a and b in Figure 6) and so determine the number of transit LSPs seen by each P(1).

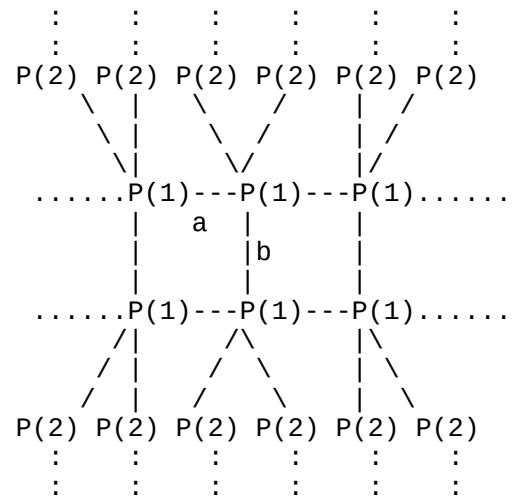

Figure 6 : An Arbitrary Section of a Ladder Network

Of course, the number of LSPs carried on links a and b in Figure 6 depends on how LSPs are routed through the core network. But if we assume a symmetrical routing policy and an even distribution of LSPs across all shortest paths, the result is the same.

Now we can see that each P(1) sees half of 2a+b LSPs (since each LSP would otherwise be counted twice as it passed through the P(1)), except that some of the LSPs are locally terminated and so are only included once in the sum 2a+b.

So L(1) = a + b/2 -
(locally terminated transit LSPs)/2 +
(locally contained LSPs)

Thus:

```
L(1) = a + b/2 -
       2*E*E*(S(1) - 1)/2 +
       E*(E-1) - E*(M(2) - 1)
     = a + b/2 +
       E*E*(2 - S(1)) - E*M(2)
```

So all we have to do is work out a and b.

Recall that the ladder length R = S(1)/2, and define X = E*E.

Consider the contribution made by all of the LSPs that make n hops on the ladder to the totals of each of a and b. If the ladder was unbounded, then we could say that in the case of a, there are n*2X LSPs along the spar only, and n(n-1)*2X/n = 2X(n-1) LSPs use a rung and the spar. Thus, the LSPs that make n hops on the ladder contribute (4n-2)X LSPs to a. Note that the edge cases are special because LSPs that make only one hop on the ladder cannot transit a P(1) but only start or end there.

So with a ladder of length R = S(1)/2, we could say:

```
      R
a = SUM[(4i-2)*X] + 2RX
    i=2

  = 2*X*R*(R+1)
```

And similarly, considering b in an unbounded ladder, the LSPs that only travel one hop on the LSP are a special case, contributing 2X LSPs, and every other LSP that traverses n hops on the ladder contributes 2n*2X/n = 4X LSPs. So:

```
         R+1
b = 2X + SUM[4X]
         i=2

  = 2*X + 4*X*R
```

In fact, the ladders are bounded, and so the number of LSPs is reduced because of the effect of the ends of the ladders. The links that see the most LSPs are in the middle of the ladder. Consider a ladder of length R; a node in the middle of the ladder is R/2 hops away from the end of the ladder. So we see that the formula for the contribution to the count of spar-only LSPs for a is only valid up to n=R/2, and for spar-and-rung LSPs, up to n=1+R/2. Above these limits, the contribution made by spar-only LSPs decays as (n-R/2)*2X.

However, for a first-order approximation, we will use the values of a and b as computed above. This gives us an upper bound of the number of LSPs without using a more complex formula for the reduction made by the effect of the ends of the ladder.

From this:

```
L(1) = a + b/2 +
       E*E*(2 - S(1)) - E*M(2)
     = 2*X*R*(R+1) +
       X + 2*X*R +
       E*E*(2 - S(1)) - E*M(2)
     = E*E*S(1)*(1 + S(1)/2) +
       E*E + E*E*S(1) +
       2*E+E - E*E*S(1) - E*M(2)
     = E*E*S(1)*(1 + S(1)/2) + 3*E+E - E*M(2)
     = E*E*S(1)*S(1)/2 + E*E*S(1) + 3*E*E - E*M(2)
```

So, for example, with S(1) = 6, M(1) = 10, and M(2) = 17, we see:

```
E     = 170
S(PE) = 1020
L(PE) = 2038
L(2)  = 34374
L(1)  = 777410
```

Alternatively, with S(1) = 10, M(1) = 10, and M(2) = 20, we see:

```
E     = 200
S(PE) = 2000
L(PE) = 3998
L(2)  = 79580
L(1)  = 2516000
```

In both examples, the number of LSPs at the core (P(1)) nodes is probably unacceptably large, even though there are only a relatively modest number of PEs. In fact, L(2) may even be too large in the second example.

Compare the L(1) values with the total number of LSPs in the system S(PE)*(S(PE) - 1), which is 1039380 and 3998000, respectively.

6. Scaling Snowflake Networks with Forwarding Adjacencies

One of the purposes of LSP hierarchies [RFC4206] is to improve the scaling properties of MPLS-TE networks. LSP tunnels (sometimes known as Forwarding Adjacencies (FAs)) may be established to provide connectivity over the core of the network, and multiple edge-to-edge LSPs may be tunneled down a single FA LSP.

In our network we consider a mesh of FA LSPs between all core nodes at the same level. We consider two possibilities here. In the first, all P(2) nodes are connected to all other P(2) nodes by LSP tunnels, and the PE-to-PE LSPs are tunneled across the core of the network. In the second, an extra layer of LSP hierarchy is introduced by connecting all P(1) nodes in an LSP mesh and tunneling the P(2)-to-P(2) tunnels through these.

6.1. Two-Layer Hierarchy

In this hierarchy model, the P(2) nodes are connected by a mesh of tunnels. This means that the P(1) nodes do not see the PE-to-PE LSPs.

It remains the case that:

$$L(PE) = 2*(S(PE) - 1)$$

L(2) is slightly increased. It can be computed as the sum of all LSPs for all attached PEs, including the LSPs between the attached PE (this figure is unchanged from Section 5.1, i.e., $M(2)*(2*S(PE) - M(2) - 1)$), plus the number of FA LSPs providing a mesh to the other P(2) nodes. Since the number of P(2) nodes is S(2), each P(2) node sees $2*(S(2) - 1)$ FA LSPs. Thus:

$$L(2) = M(2)*(2*S(PE) - M(2) - 1) + 2*(S(2) - 1)$$

L(1), however, is significantly reduced and can be computed as the sum of the number of FA LSPs to and from each attached P(2) to each other P(2) in the network, including (but counting only once) the FA LSPs between attached P(2) nodes. In fact, the problem is identical to the L(2) computation in Section 5.1. So:

$$L(1) = M(1)*(2*S(2) - M(1) - 1)$$

So, for example, with S(1) = 5, M(1) = 10, and M(2) = 20, we see:

```
S(PE) = 1000
S(2)  = 50
L(PE) = 1998
L(2)  = 39678
L(1)  = 890
```

Alternatively, with S(1) = 10, M(1) = 10, and M(2) = 20, we see:

```
S(PE) = 2000
S(2)  = 100
L(PE) = 3998
L(2)  = 79778
L(1)  = 1890
```

So, in both examples, potential problems at the core (P(1)) nodes caused by an excessive number of LSPs can be avoided, but any problem with L(2) is made slightly worse, as can be seen from the table below.

Example	Count	Unmodified (Section 5.1)	2-Layer Hierarchy
A	L(2)	39580	39678
	L(1)	356000	890
B	L(2)	79580	79778
	L(1)	756000	1890

6.1.1. Tuning the Network Topology to Suit the Two-Layer Hierarchy

Clearly, we can reduce L(2) by selecting appropriate values of S(1), M(1), and M(2). We can do this without negative consequences, since no change will affect L(PE) and since a large percentage increase in L(1) is sustainable now that L(1) is so small.

Observe that:

```
L(2) = M(2)*(2*S(PE) - M(2) - 1) + 2*(S(2) - 1)
```

where S(PE) = S(1)*M(1)*M(2) and S(2) = S(1)*M(1). So L(2) scales with M(2)^2 and we can have the most impact by reducing M(2) while keeping S(PE) constant.

For example, with S(1) = 10, M(1) = 10, and M(2) = 10, we see:

```
S(PE) = 1000
S(2)  = 100
L(PE) = 1998
L(2)  = 20088
L(1)  = 1890
```

And similarly, with S(1) = 20, M(1) = 20, and M(2) = 5, we see:

```
S(PE) = 2000
S(2)  = 400
L(PE) = 3998
L(2)  = 20768
L(1)  = 15580
```

These considerable scaling benefits must be offset against the cost-effectiveness of the network. Recall from Section 3.3 that:

```
K = S(PE)/(S(1)+S(2) ... + S(n))
```

where n is the level above the PEs, so that for our network:

```
K = S(PE) / (S(1) + S(2))
```

Thus, in the first example the cost-effectiveness has been halved from 1000/55 to 1000/110. In the second example, it has been reduced to roughly one quarter, changing from 2000/110 to 2000/420.

So, although the tuning changes may be necessary to reach the desired network size, they come at a considerable cost to the operator.

6.2. Alternative Two-Layer Hierarchy

An alternative to the two-layer hierarchy presented in Section 6.1 is to provide a full mesh of FA LSPs between P(1) nodes. This technique is only of benefit to any nodes in the core of the level 1 network. It makes no difference to the PE and P(2) nodes since they continue to see only the PE-to-PE LSPs. Furthermore, this approach increases the burden at the P(1) nodes since they have to support all of the PE-to-PE LSPs as in the flat model plus the additional 2*(S(1) - 1) P(1)-to-P(1) FA LSPs. Thus, this approach should only be considered where there is a mesh of P-nodes within the ring of P(1) nodes, and is not considered further in this document.

6.3. Three-Layer Hierarchy

As demonstrated by Section 6.2, introducing a mesh of FA LSPs at the top level (P(1)) has no benefit, but if we introduce an additional level in the network (P(3) between P(2) and PE) to make a four-level snowflake, we can introduce a new layer of FA LSPs so that we have a full mesh of FA LSPs between all P(3) nodes to carry the PE-to-PE LSPs, and a full mesh of FA LSPs between all P(2) nodes to carry the P(3)-to-P(3) LSPs.

The number of PEs is S(PE) = S(1)*M(1)*M(2)*M(3), and the number of PE-to-PE LSPs at a PE remains L(PE) = 2*(S(PE) - 1).

The number of LSPs at a P(3) can be deduced from Section 6.1. It is the sum of all LSPs for all attached PEs, including the LSPs between the attached PE, plus the number of FA LSPs providing a mesh to the other P(3) nodes.

L(3) = M(3)*(2*S(PE) - M(3) - 1) + 2*(S(3) - 1)

The number of LSPs at P(2) can also be deduced from Section 6.1 since it is the sum of all LSPs for all attached P(3) nodes, including the LSPs between the attached PE plus the number of FA LSPs providing a mesh to the other P(2) nodes.

L(2) = M(2)*(2*S(3) - M(2) - 1) + 2*(S(2) - 1)

Finally, L(1) can be copied straight from 6.1.

L(1) = M(1)*(2*S(2) - M(1) - 1)

For example, with S(1) = 5, M(1) = 5, M(2) = 5, and M(3) = 8, we see:

```
S(PE) = 1000
S(3)  = 125
S(2)  = 25
L(PE) = 1998
L(3)  = 16176
L(2)  = 1268
L(1)  = 220
```

Similarly, with S(1) = 5, M(1) = 5, M(2) = 8, and M(3) = 10, we see:

```
S(PE) = 2000
S(3)  = 200
S(2)  = 25
L(PE) = 3998
L(3)  = 40038
L(2)  = 3184
L(1)  = 220
```

Clearly, there are considerable scaling improvements with this three-layer hierarchy, and all of the numbers (even L(3) in the second example) are manageable.

Of course, the extra level in the network tends to reduce the cost-effectiveness of the networks with values of K = 1000/155 and K = 2000/230 (from 1000/55 and 2000/110) for the examples above. That is a reduction by a factor of 3 in the first case and 2 in the second case. Such a change in cost-effectiveness has to be weighed against the desire to deploy such a large network. If LSP hierarchies are the only scaling tool available, and networks this size are required, the cost-effectiveness may need to be sacrificed.

6.4. Issues with Hierarchical LSPs

A basic observation for hierarchical scaling techniques is that it is hard to have any impact on the number of LSPs that must be supported by the level of P(n) nodes adjacent to the PEs (for example, it is hard to reduce L(3) in Section 6.3). In fact, the only way we can change the number of LSPs supported by these nodes is to change the scaling ratio M(n) in the network -- in other words, to change the number of PEs subtended to any P(n). But such a change has a direct effect on the number of PEs in the network and so the cost-effectiveness is impacted.

Another concern with the hierarchical approach is that it must be configured and managed. This may not seem like a large burden, but it must be recalled that the P(n) nodes are not at the edge of the network -- they are a set of nodes that must be identified so that the FA LSPs can be configured and provisioned. Effectively, the operator must plan and construct a layered network with a ring of P(n) nodes giving access to the level (n) network. This design activity is open to considerable risk as failing to close the ring (i.e., allowing a node to be at both level (n+1) and at level (n)) may cause operational confusion.

Protocol techniques (such as IGP automesh [RFC4972]) have been developed to reduce the configuration necessary to build this type of multi-level network. In the case of automesh, the routing protocol is used to advertise the membership of a 'mesh group', and all members of the mesh group can discover each other and connect with LSP tunnels. Thus, the P(n) nodes giving access to level (n) can advertise their existence to each other, and it is not necessary to configure each with information about all of the others. Although this process can help to reduce the configuration overhead, it does not eliminate it, as each member of the mesh group must still be planned and configured for membership.

An important consideration for the use of hierarchical LSPs is how they can be protected using MPLS Fast Reroute (FRR) [RFC4090]. FRR may provide link protection either by protecting the tunnels as they traverse a broken link or by treating each level (n) tunnel LSP as a link in level (n+1) and providing protection for the level (n+1) LSPs (although in this model, fault detection and propagation time may be an issue). Node protection may be performed in a similar way, but protection of the first and last nodes of a hierarchical LSP is particularly difficult. Additionally, the whole notion of scaling with regard to FRR gives rise to separate concerns that are outside the scope of this document as currently formulated.

Finally, observe that we have been explaining these techniques using conveniently symmetrical networks. Consider how we would arrange the hierarchical LSPs in a network where some PEs are connected closer to the center of the network than others.

7. Scaling Ladder Networks with Forwarding Adjacencies

7.1. Two-Layer Hierarchy

In Section 6.2, we observed that there is no value to placing FA LSPs between the P(1) nodes of our example snowflake topologies. This is because those LSPs would be just one hop long and would, in fact, only serve to increase the burden at the P(1) nodes. However, in the ladder model, there is value to this approach. The P(1) nodes are the spar-nodes of the ladder, and they are not all mutually adjacent. That is, the P(1)-to-P(1) hierarchical LSPs can create a full mesh of P(1) nodes where one does not exist in the physical topology.

The number of LSPs seen by a P(1) node is then:

- all of the tunnels terminating at the P(1) node,
- any transit tunnels, and
- all of the LSPs due to subtended PEs.

This is a substantial reduction; all of the transit LSPs are reduced to just one per remote P(1) that causes any transit LSP. So:

```
L(1) = 2*(S(1) - 1) +
       O(S(1)*S(1)/2) +
       2*E*E*(S(1) - 1) + E*(E-1) - E*(M(2) - 1)
```

where O(S(1)*S(1)/2) gives an upper bound order of magnitude. So:

```
L(1) = S(1)*S(1)/2 + 2*S(1) + 2*E*E*(S(1) - 1) - E*M(2) - 2
```

So, in our two examples:

With S(1) = 6, M(1) = 10, and M(2) = 17, we see:

```
E     = 170
S(PE) = 1020
L(PE) = 2038
L(2)  = 34374
L(1)  = 286138
```

Alternatively, with S(1) = 10, M(1) = 10, and M(2) = 20, we see:

```
E     = 200
S(PE) = 2000
L(PE) = 3998
L(2)  = 79580
L(1)  = 716060
```

Both of these show significant improvements over the previous L(1) figures of 777410 and 2516000. But the numbers are still too large to manage, and there is no improvement in the L(2) figures.

7.2. Three-Layer Hierarchy

We can also apply the three-layer hierarchy to the ladder model. In this case, the number of LSPs between P(1) nodes is not reduced, but tunnels are also set up between all P(2) nodes. Thus, the number of LSPs seen by a P(1) node is:

- all of the tunnels terminating at the P(1) node,
- any transit tunnels between P(1) nodes, and
- all of the LSPs due to subtended P(2) nodes.

No PE-to-PE LSPs are seen at the P(1) nodes.

```
L(1) = 2*(S(1) - 1) +
       O(S(1)*S(1)/2) +
       2*(S(1) - 1)*M(1)*M(1) + M(1)*(M(1) - 1)
```

where O(S(1)*S(1)/2) gives an upper bound order of magnitude. So:

```
L(1) = S(1)*S(1)/2 + 2*S(1) + 2*M(1)*M(1)*S(1) - M(1)(M(1) + 1) - 2
```

Unfortunately, there is a small increase in the number of LSPs seen by the P(2) nodes. Each P(2) now sees all of the PE-to-PE LSPs that it saw before and is also an end-point for a set of P(2)-to-P(2) tunnels. Thus, L(2) increases to:

```
L(2) = 2*M(2)*(S(PE) - 1) - M(2)*(M(2) - 1) + 2*(S(1)*M(1) - 1)
```

So, in our two examples:

With S(1) = 6, M(1) = 10, and M(2) = 17, we see:

```
E     = 170
S(PE) = 1020
L(PE) = 2038
L(2)  = 34492
L(1)  = 1118
```

Alternatively, with S(1) = 10, M(1) = 10, and M(2) = 20, we see:

```
E     = 200
S(PE) = 2000
L(PE) = 3998
L(2)  = 79778
L(1)  = 1958
```

This represents a very dramatic decrease in LSPs across the core.

7.3. Issues with Hierarchical LSPs

The same issues exist for hierarchical LSPs as described in Section 6.4. Although dramatic improvements can be made to the scaling numbers for the number of LSPs at core nodes, this can only be done at the cost of configuring P(2) to P(2) tunnels. The mesh of P(1) tunnels is not enough.

But the sheer number of P(2) to P(2) tunnels that must be configured is a significant management burden that can only be eased by using a technique like automesh [RFC4972].

It is significant, however, that the scaling problem at the P(2) nodes cannot be improved by using tunnels and that the only solution to ease this in the hierarchical approach would be to institute another layer of hierarchy (that is, P(3) nodes) between the P(2) nodes and the PEs. This is, of course, a significant expense.

8. Scaling Improvements through Multipoint-to-Point LSPs

An alternative (or complementary) scaling technique has been proposed using multipoint-to-point (MP2P) LSPs. The fundamental improvement in this case is achieved by reducing the number of LSPs toward the destination as LSPs toward the same destination are merged.

This section presents an overview of MP2P LSPs and describes their applicability and scaling benefits.

8.1. Overview of MP2P LSPs

Note that the MP2P LSPs discussed here are for MPLS-TE and are not the same concept familiar in the Label Distribution Protocol (LDP) described in [RFC5036].

Traffic flows generally converge toward their destination and this can be utilized by MPLS in constructing an MP2P LSP. With such an LSP, the Label Forwarding Information Base (LFIB) mappings at each LSR are many-to-one so that multiple pairs {incoming interface, incoming label} are mapped to a single pair {outgoing interface, outgoing label}. Obviously, if per-platform labels are used, this mapping may be optimized within an implementation.

It is important to note that with MP2P MPLS-TE LSPs, the traffic flows are merged. That is, some additional form of identifier is required if de-merging is required. For example, if the payload is IP traffic belonging to the same client network, no additional de-merging information is required since the IP packet contains sufficient data. On the other hand, if the data comes, for example, from a variety of VPN client networks, then the flows will need to be labeled in their own right as point-to-point (P2P) flows, so that traffic can be disambiguated at the egress of the MP2P LSPs.

Techniques for establishing MP2P MPLS-TE LSPs and for assigning the correct bandwidth downstream of LSP merge points are out of the scope of this document.

8.2. LSP State: A Better Measure of Scalability

Consider the network topology shown in Figure 3. Suppose that we establish MP2P LSP tunnels such that there is one tunnel terminating at each PE, and that that tunnel has every other PE as an ingress. Thus, a PE-to-PE MP2P LSP tunnel would have S(PE)-1 ingresses and one egress, and there would be S(PE) such tunnels.

Note that there still remain 2*(S(PE) - 1) PE-to-PE P2P LSPs that are carried through these tunnels.

Let’s consider the number of LSPs handled at each node in the network.

The PEs continue to handle the same number of PE-to-PE P2P LSPs, and must also handle the MP2P LSPs. So:

L(PE) = 2*(S(PE) - 1) + S(PE)

But all P(n) nodes in the network only handle the MP2P LSP tunnels. Nominally, this means that L(n) = S(PE) for all values of n. This would appear to be a great success with the number of LSPs cut to completely manageable levels.

However, the number of LSPs is not the only issue (although it may have some impact for some of the scaling concerns listed in Section 4). We are more interested in the amount of LSP state that is maintained by an LSR. This reflects the amount of storage required at the LSR, the amount of protocol processing, and the amount of information that needs to be managed.

In fact, we were also interested in this measure of scalability in the earlier sections of this document, but in those cases we could see a direct correlation between the number of LSPs and the amount of LSP state since transit LSPs had two pieces of state information (one on the incoming and one on the outgoing interface), and ingress or egress LSPs had just one piece of state.

We can quantify the amount of LSP state according to the number of LSP segments managed by an LSR. So (as above), in the case of a P2P LSP, an ingress or egress has one segment to maintain, while a transit has two segments. Similarly, for an MP2P LSP, an LSR must maintain one set of state information for each upstream segment (which, we can assume, is in a one-to-one relationship with the number of upstream neighbors) and exactly one downstream segment -- ingresses obviously have no upstream neighbors, and egresses have no downstream segments.

So we can start again on our examination of the scaling properties of MP2P LSPs using X(n) to represent the amount of LSP state held at each P(n) node.

8.3. Scaling Improvements for Snowflake Networks

At the PEs, there is only connectivity to one other network node: the P(2) node. But note that if P2P LSPs need to be used to allow disambiguation of data at the MP2P LSP egresses, then these P2P LSPs are tunneled within the MP2P LSPs. So X(PE) is:

X(PE) = 2*(S(PE) - 1) if no disambiguation is required,

and

X(PE) = 4*(S(PE) - 1) if disambiguation is required.

Each P(2) node has M(2) downstream PEs. The P(2) sees a single MP2P LSP targeted at each downstream PE with one downstream segment (to that PE) and M(2) - 1 upstream segments from the other subtended PEs. Additionally, each of these LSPs has an upstream segment from the one upstream P(1). This gives a total of M(2)*(1 + M(2)) LSP segments.

There are also LSPs running from the subtended PEs to every other PE in the network. There are S(PE) - M(2) such PEs, and the P(2) sees one upstream segment for each of these from each subtended PE. It also has one downstream segment for each of these LSPs. This gives (M(2) + 1)*(S(PE) - M(2)) LSP segments.

Thus:

```
X(2) = M(2)*(1 + M(2)) + (M(2) + 1)*(S(PE) - M(2))
     = S(PE)*(M(2) + 1)
```

Similarly, at each P(1) node there are M(1) downstream P(2) nodes and so a total of M(1)*M(2) downstream PEs. Each P(1) is connected in a full mesh with the other P(1) nodes and so has (S(1) - 1) neighbors.

The P(1) sees a single MP2P LSP targeted at each downstream PE. This has one downstream segment (to the P(2) to which the PE is connected) and M(1) - 1 upstream segments from the other subtended P(2) nodes. Additionally, each of these LSPs has an upstream segment from each of the P(1) neighbors. This gives a total number of LSP segments of M(1)*M(2)*(M(1) + S(1) - 1).

There are also LSPs running from each of the subtended PEs to every other PE in the network. There are S(PE) - M(1)M(2) such PEs, and the P(1) sees one upstream segment for each of these from each

subtended P(2) (since the aggregation from the subtended PEs has already happened at the P(2) nodes). It also has one downstream segment to the appropriate next hop P(1) neighbor for each of these LSPs. This gives (M(1) + 1)*(S(PE) - M(1)*M(2)) LSP segments.

Thus:

```
X(1) = M(1)*M(2)*(M(1) + S(1) - 1) +
       (M(1) + 1)*(S(PE) - M(1)*M(2))
     = M(1)*M(2)*(S(1) - 2) + S(PE)*(M(1) + 1)
```

So, for example, with S(1) = 10, M(1) = 10, and M(2) = 10, we see:

```
S(PE) = 1000
S(2)  = 100
X(PE) = 3996   (or 1998)
X(2)  = 11000
X(1)  = 11800
```

And similarly, with S(1) = 20, M(1) = 20, and M(2) = 5, we see:

```
S(PE) = 2000
S(2)  = 400
X(PE) = 5996   (or 2998)
X(2)  = 12000
X(1)  = 39800
```

8.3.1. Comparison with Other Scenarios

For comparison with the examples in Sections 5 and 6, we need to convert those LSP-based figures to our new measure of LSP state.

Observe that each LSP in Sections 5 and 6 generates two state units at a transit LSR and one at an ingress or egress. So we can provide conversions as follows:

Section 5 (flat snowflake network)

```
L(PE) = 2*(S(PE) - 1)
L(2)  = M(2)*(2*S(PE) - M(2) - 1)
L(1)  = M(1)*M(2)*(2*S(PE) - M(2)*(M(1) + 1))
X(PE) = 2*(S(PE) - 1)
X(2)  = 2*M(2)*(2*S(PE) - M(2) - 1)
X(1)  = 2*M(1)*M(2)*(2*S(PE) - M(2)*(M(1) + 1))
```

For the example with S(1) = 10, M(1) = 10, and M(2) = 10, this gives a comparison table as follows:

Count	Unmodified	MP2P
X(PE)	1998	3996
X(2)	39780	11000
X(1)	378000	11800

Clearly, this technique is a significant improvement over the flat network within the core of the network, although the PEs are more heavily stressed if disambiguation is required.

Section 6.1 (two-layer hierarchy snowflake network)

```
L(PE) = 2*(S(PE) - 1)
L(2)  = M(2)*(2*S(PE) - M(2) - 1) + 2*(S(2) - 1)
L(1)  = M(1)*(2*S(2) - M(1) - 1)
X(PE) = 2*(S(PE) - 1)
X(2)  = 2*M(2)*(2*S(PE) - M(2) - 1) + 2*(S(2) - 1)
X(1)  = 2*M(1)*(2*S(2) - M(1) - 1)
```

Note that in the computation of X(2) the hierarchical LSPs only add one state at each P(2) node.

For the same example with S(1) = 10, M(1) = 10, and M(2) = 10, this gives a comparison table as follows:

Count	2-Layer Hierarchy	MP2P
X(PE)	1998	3996
X(2)	39978	11000
X(1)	3780	11800

We can observe that the MP2P model is better at P(2), but the hierarchical model is better at P(1).

In fact, this comparison can be generalized to observe that the MP2P model produces its best effects toward the edge of the network, while the hierarchical model makes most impression at the core. However, the requirement for disambiguation of P2P LSPs tunneled within the MP2P LSPs does cause a double burden at the PEs.

8.4. Scaling Improvements for Ladder Networks

MP2P LSPs applied just within the ladder will not make a significant difference, but applying MP2P for all LSPs and at all nodes makes a very big difference without requiring any further configuration.

LSP state at a spar-node may be divided into those LSPs' segments that enter or leave the spar-node due to subtended PEs (local LSP segments), and those that enter or leave the spar-node due to remote PEs (remote segments).

The local segments may be counted as:

- E LSPs targeting local PEs
- (S(1)-1)*E*M(1) LSPs targeting remote PEs

The remote segments may be counted as:

- (S(1)-1)*E outgoing LSPs targeting remote PEs
- <= 3*S(1)*E incoming LSPs targeting any PE (there are precisely P(1) nodes attached to any other P(1) node)

Hence, using X(1) as a measure of LSP state rather than a count of LSPs, we get:

```
X(1) <= E + (S(1)-1)*E*M(1) + (S(1)-1)*E + 3*S(1)*E
     <= (4 + M(1))*S(1)*E - M(1)*E
```

The number of LSPs at the P(2) nodes is also improved. We may also count the LSP state in the same way so that there are:

- M(2) LSPs targeting local PEs,
- M(2)*(S(1)*E) LSPs from local PEs to all other PEs, and
- S(1)*E - M(2) LSPs to remote PEs.

So using X(2) as a measure of LSP state and not a count of LSPs, we have:

```
X(2) = M(2) + M(2)*(S(1)*E) + S(1)*E - M(2)
     = (M(2) + 1)*S(1)*E
```

Our examples from Section 5.2 give us the following numbers:

With S(1) = 6, M(1) = 10, and M(2) = 17, we see:

```
E     = 170
S(PE) = 1020
X(PE) = 2038
X(2)  = 18360
X(1) <= 12580
```

Alternatively, with S(1) = 10, M(1) = 10, and M(2) = 20, we see:

```
E     = 200
S(PE) = 2000
X(PE) = 3998
X(2)  = 42000
X(1) <= 26000
```

8.4.1. Comparison with Other Scenarios

The use of MP2P compares very favorably with all scaling scenarios. It is the only technique able to reduce the value of X(2), and it does this by a factor of almost two. The impact on X(1) is better than everything except the three-level hierarchy.

The following table provides a quick cross-reference for the figures for the example ladder networks. Note that the previous figures are modified to provide counts of LSP state rather than LSP numbers. Again, each LSP contributes one state at its end points and two states at transit nodes.

Thus, for the all cases we have:

```
X(PE) = 2*(S(PE) - 1) or
X(PE) = 4*(S(PE) - 1) if disambiguation is required.
```

In the unmodified (flat) case, we have:

```
X(2) = 2*(M(2)*(2*S(PE) - M(2) - 1))
X(1) = 2*(M(1)*M(2)*(2*S(PE) - M(2)*(M(1) + 1)))
```

In the two-level hierarchy, we have:

```
X(2) = 2*(2*M(2)*(S(PE) - 1) - M(2)*(M(2) - 1))
X(1) = S(1)*S(1) + 2*S(1) + 4*E*E*(S(1) - 1) - 2*E*M(2) - 2
```

In the three-level hierarchy, we have:

```
X(2) = 2*(2*M(2)*(S(PE) - 1) - M(2)*(M(2) - 1)) + 2*(S(1)*M(1) - 1)
X(1) = S(1)*S(1) + 2*S(1) + 4*M(1)*M(1)*S(1) - 2*M(1)(M(1) + 1) - 2
```

Example A: S(1) = 6, M(1) = 10, and M(2) = 17
Example B: S(1) = 10, M(1) = 10, and M(2) = 20

Example	Count	Unmodified	2-Level Hierarchy	3-Level Hierarchy	MP2P
A	X(2)	68748	68748	68866	18360
	X(1)	1554820	572266	2226	12580
B	X(2)	159160	159160	159358	42000
	X(1)	5032000	1433998	3898	26000

8.4.2. LSP State Compared with LSP Numbers

Recall that in Section 8.3, the true benefit of MP2P was analyzed with respect to the LSP segment state required, rather than the actual number of LSPs. This proved to be a more accurate comparison of the techniques because the MP2P LSPs require state on each branch of the LSP, so the saving is not linear with the reduced number of LSPs.

A similar analysis could be performed here for the ladder network. The net effect is that it increases the state by an order of two for all transit LSPs in the P2P models, and by a multiplier equal to the degree of a node in the MP2P model.

A rough estimate shows that, as with snowflake networks, MP2P provides better scaling than the one-level hierarchical model and is considerably better at the core. But MP2P compares less will with the two-level hierarchy especially in the core.

8.5. Issues with MP2P LSPs

The biggest challenges for MP2P LSPs are the provision of support in the control and data planes. To some extent, support must also be provided in the management plane.

Control plane support is just a matter of defining the protocols and procedures [MP2P-RSVP], although it must be clearly understood that this will introduce some complexity to the control plane.

Hardware issues may be a little more tricky. For example, the capacity of the upstream segments must never (allowing for statistical over-subscription) exceed the capacity of the downstream segment. Similarly, data planes must be equipped with sufficient buffers to handle incoming packet collisions.

The management plane will be impacted in several ways. Firstly, the management applications will need to handle LSPs with multiple senders. This means that, although the applications need to process fewer LSPs, they will be more complicated and will, in fact, need to

process the same number of ingresses and egresses. Other issues like diagnostics and OAM would also need to be enhanced to support MP2P, but might be borrowed heavily from LDP networks.

Lastly, note that when the MP2P solution is used, the receiver (the single egress PE of an MP2P tunnel) cannot use the incoming label as an indicator of the source of the data. Contrast this with P2P LSPs. Depending on deployment, this might not be an issue since the PE-PE connectivity may in any case be a tunnel with inner labels to discriminate the data flows.

In other deployments, it may be considered necessary to include additional PE-PE P2P LSPs and tunnel these through the MP2P LSPs. This would require the PEs to support twice as many LSPs. Since PEs are not usually as fully specified as P-routers, this may cause some concern; however, the use of penultimate hop popping on the MP2P LSPs might help to reduce this issue.

In all cases, care must be taken not to confuse the reduction in the number of LSPs with a reduction in the LSP state that is required. In fact, the discussion in Section 8.3 is slightly optimistic since LSP state toward the destination will probably need to include sender information and so will increase depending on the number of senders for the MP2P LSP. Section 8.4, on the other hand, counts LSP state rather than LSPs. This issue is clearly dependent on the protocol solution for MP2P RSVP-TE, which is out of scope for this document.

MPLS Fast Reroute (FRR) [RFC4090] is an attractive scheme for providing rapid local protection from node or link failures. Such a scheme has, however, not been designed for MP2P at the time of writing, so it remains to be seen how practical it could be, especially in the case of the failure of a merge node. Initial examination of this case suggests that FRR would not be a problem for MP2P, given that each flow can be handled separately.

As a final note, observe that the MP2P scenario presented in this document may be optimistic. MP2P LSP merging may be hard to achieve between LSPs with significantly different traffic and Quality of Service (QoS) parameters. Therefore, it may be necessary to increase the number of MP2P LSPs arriving at an egress.

9. Combined Models

There is nothing to prevent the combination of hierarchical and MP2P solutions within a network.

Note that if MP2P LSPs are tunneled through P2P FA LSPs across the core, none of the benefit of LSP merging is seen for the hops during which the MP2P LSPs are tunneled.

On the other hand, it is possible to construct solutions where MP2P FA LSPs are constructed within the network, resulting in savings from both modes of operation.

10. An Alternate Solution

A simple solution to reducing the number of LSP tunnels handled by any node in the network has been proposed. In this solution it is observed that part of the problem is caused purely by the total number of LSP in the network, and that this is a function of the number of PEs since a full mesh of PE-PE LSPs is required. The conclusion of this observation is to move the tunnel end-points further into the network so that, instead of having a full mesh of PE-PE tunnels, we have only a full mesh of P(n)-P(n) tunnels.

Obviously, there is no change in the physical network topology, so the PEs remain subtended to the P(n) nodes, and the consequence is that there is no TE on the links between PEs and P(n) nodes.

In this case, we have already done the hard work for computing the number of LSPs in the previous sections. The power of the analysis in the earlier sections is demonstrated by its applicability to this new model -- all we need to do is make minor changes to the formulae. This is most simply done by removing a layer from the network. We introduce the term "tunnel end-point" (TEP) and replace the P(n) nodes with TEPs. Thus, the example of a flat snowflake network in Figure 3 becomes as shown in Figure 7. Corresponding changes can be made to all of the sample topologies.

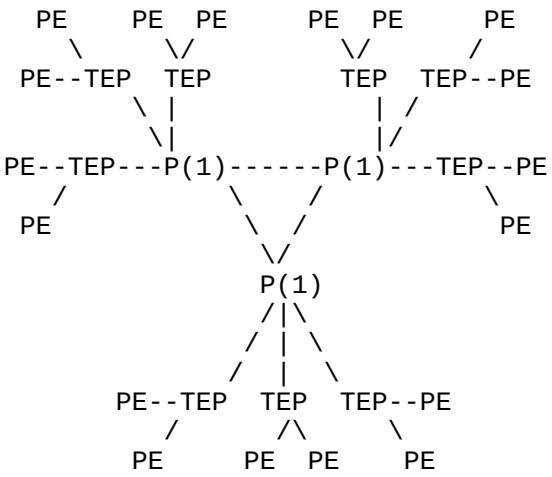

Figure 7 : An Example Snowflake Network with Tunnel End-Points

To perform the scaling calculations we need only replace the PE counts in the formulae with TEP counts, and observe that there is one fewer layer in the network. For example, in the flat snowflake network shown in Figure 7, we can see that the number of LSPs seen at a TEP is:

L(TEP) = 2*(S(TPE) - 1)

In our sample networks, S(TPE) is typically of the order of 50 or 100 (the original values of S(2)), so L(TEP) is less than 200, which is quite manageable.

Similarly, the number of LSPs handled by a P(1) node can be derived from the original formula for the number of LSPs seen at a P(2) node, since all we have done is reduce n in P(n) from 2 to 1. So our new formula is:

L(1) = M(1)*(2*S(TEP) - M(1) - 1)

With figures for M(1) = 10 and S(TEP) = 100, this gives us L(1) = 1890. This is also very manageable.

10.1. Pros and Cons of the Alternate Solution

On the face of it, this alternate solution seems very attractive. Simply by contracting the edges of the tunnels into the network, we have shown a dramatic reduction in the number of tunnels needed, and there is no requirement to apply any additional scaling techniques.

But what of the PE-P(n) links? In the earlier sections of this document, we have assumed that there was some requirement for PE-PE LSPs with TE properties that extended to the PE-P(n) links at both ends of each LSP. That means that there was a requirement to provide reservation-based QoS on those links, to be able to discriminate traffic flows for priority-based treatment, and to be able to distinguish applications and sources that send data based on the LSPs that carry the data.

It might be argued that, since the PE-P(n) links do not offer any routing options (each such link provides the only access to the network for a PE), most of the benefits of tunnels are lost on these peripheral links. However, TE is not just about routing. Just as important are the abilities to make resource reservations, to prioritize traffic, and to discriminate between traffic from different applications, customers, or VPNs.

Furthermore, in multihoming scenarios where each PE is connected to more than one P(n) or where a PE has multiple links to a single P(n), there may be a desire to pre-select the link to be used and to direct the traffic to that link using a PE-PE LSP. Note that multihoming has not been considered in this document.

Operationally, P(n)-P(n) LSPs offer the additional management overhead that is seen for hierarchical LSPs described in Section 6. That is, the LSPs have to be configured and established through additional configuration or management operations that are not carried out at the PEs. As described in Section 6, automesh [RFC4972] could be used to ease this task. But it must be noted that, as mentioned above, some of the key uses of tunnels require that traffic is classified and placed in an appropriate tunnel according to its traffic class, end-points, originating application, and customer (such as client VPN). This information may not be readily available for each packet at the P(n) nodes since it is PE-based information. Of course, it is possible to conceive of techniques to make this information available, such as assigning a different label for each class of traffic, but this gives rise to the original problem of larger numbers of LSPs.

Our conclusion is, therefore, that this alternate technique may be suitable for the general distribution of traffic based solely on the destination, or on a combination of the destination and key fields carried in the IP header. In this case, it can provide a very satisfactory answer to the scaling issues in an MPLS-TE network. But if more sophisticated packet classification and discrimination is required, this technique will make the desired function hard to

achieve, and the trade-off between scaling and feature-level will swing too far towards solving the scaling issue at the expense of delivery of function to the customer.

11. Management Considerations

The management issues of the models presented in this document have been discussed in-line. No one solution is without its management overhead.

Note, however, that scalability of management tools is one of the motivators for this work and that network scaling solutions that reduce the active management of LSPs at the cost of additional effort to manage the more static elements of the network represent a benefit. That is, it is worth the additional effort to set up MP2P or FA LSPs if it means that the network can be scaled to a larger size without being constrained by the management tools.

The MP2P technique may prove harder to debug through OAM methods than the FA LSP approach.

12. Security Considerations

The techniques described in this document use existing or yet-to-be-defined signaling protocol extensions and are subject to the security provided by those extensions. Note that we are talking about tunneling techniques used within the network and that both approaches are vulnerable to the creation of bogus tunnels that deliver data to an egress or consume network resources.

The fact that the MP2P technique may prove harder to debug through OAM methods than the FA LSP approach is a security concern since it is important to be able to detect misconnections.

General issues of the relationship between scaling and security are covered in Section 1.1, but the details are beyond the scope of this document. Readers are referred to [MPLS-SEC] for details of MPLS security techniques.

13. Recommendations

The analysis in this document suggests that the ability to signal MP2P MPLS-TE LSPs is a desirable addition to the operator’s MPLS-TE toolkit.

At this stage, no further recommendations are made, but it would be valuable to consult more widely to discover:

- The concerns of other service providers with respect to network scalability.

- More opinions on the realistic constraints to the network parameters listed in Section 4.

- Desirable values for the cost-effectiveness of the network (parameter K).

- The applicability, manageability, and support for the two techniques described.

- The feasibility of combining the two techniques, as discussed in Section 9.

- The level of concern over the loss of functionality that would occur if the alternate solution described in Section 10 was adopted.

14. Acknowledgements

The authors are grateful to Jean-Louis Le Roux for discussions and review input. Thanks to Ben Niven-Jenkins, JP Vasseur, Loa Andersson, Anders Gavler, Ben Campbell, and Tim Polk for their comments. Thanks to Dave Allen for useful discussion of the math.

15. Normative References

[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.

16. Informative References

[RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, April 2001.

[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.

[RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, May 2002.

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

[RFC3985] Bryant, S., Ed., and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, March 2005.

[RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.

[RFC4110] Callon, R. and M. Suzuki, "A Framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs)", RFC 4110, July 2005.

[RFC4972] Vasseur, JP., Ed., Leroux, JL., Ed., Yasukawa, S., Previdi, S., Psenak, P., and P. Mabbey, "Routing Extensions for Discovery of Multiprotocol (MPLS) Label Switch Router (LSR) Traffic Engineering (TE) Mesh Membership", RFC 4972, July 2007.

[RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, October 2007.

[MP2P-RSVP] Yasukawa, Y., "Supporting Multipoint-to-Point Label Switched Paths in Multiprotocol Label Switching Traffic Engineering", Work in Progress, October 2008.

[MPLS-SEC] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", Work in Progress, November 2008.

Authors' Addresses

Seisho Yasukawa
NTT Corporation
9-11, Midori-Cho 3-Chome
Musashino-Shi, Tokyo 180-8585 Japan
Phone: +81 422 59 4769
EMail: s.yasukawa@hco.ntt.co.jp

Adrian Farrel
Old Dog Consulting
EMail: adrian@olddog.co.uk

Olufemi Komolafe
Cisco Systems
96 Commercial Street
Edinburgh
EH6 6LX
United Kingdom
EMail: femi@cisco.com