DECLARATION OF SANDY GINOZA
FOR IETF ADMINISTRATION LLC (IETF)

RFC 3031: Multiprotocol Label Switching Architecture

RFC 3272: Overview and Principles of Internet Traffic Engineering

RFC 3386: Network Hierarchy and Multilayer Survivability

RFC 3469: Framework for Multi-Protocol Label Switching (MPLS)-based Recovery

RFC 3916: Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)

RFC 3985: Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture

RFC 4447: Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)

RFC 4448: Encapsulation Methods for Transport of Ethernet over MPLS Networks

RFC 4619: Encapsulation Methods for Transport of Frame Relay over Multiprotocol Label Switching (MPLS) Networks

RFC 5659: An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge

I, Sandy Ginoza, hereby declare that all statements made herein are of my own knowledge and are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code:

1. I am an employee of Association Management Solutions, LLC (AMS), which acts under contract to the IETF Administration LLC (IETF) as the operator of the RFC Production Center. The RFC Production Center is part of the "RFC Editor" function, which prepares documents for publication and places files in an online repository for the authoritative Request for Comments (RFC) series of documents (RFC Series), and

preserves records relating to these documents. The RFC Series includes, among other things, the series of Internet standards developed by the IETF. I hold the position of RFC Production Center Director of Operations. I began employment with AMS on 6 January 2010.

2. My responsibilities as RFC Production Center Director of Operations include acting as the custodian of records relating to the RFC Series, and I am familiar with the record keeping practices relating to the RFC Series, including the creation and maintenance of such records.

3. I have held my position as RFC Production Center Director of Operations since 1 January 2025. Prior to my employment with AMS, I was an employee of the Information Sciences Institute at University of Southern California (ISI) (June 1999 - January 2010). With my employment in the RFC Editor project, I held various position titles, including Senior Editor and Director positions, before assuming my current title as Director of Operations.

4. The RFC Editor function was conducted by the Information Sciences Institute at the University of California ("ISI") under contract to the United States government prior to 1998. In 1998, the Internet Society (ISOC), in furtherance of its IETF activity, entered into the first in a series of contracts with ISI providing for ISI's performance of the RFC Editor function. Beginning in 2010, certain aspects of the RFC Editor function were assumed by the RFC Production Center operation of AMS under contract to ISOC (acting through its IETF function and, in particular, the IETF Administrative Oversight Committee (now the IETF Administration LLC)). At the beginning of 2025, the management of the RFC Production Center fully transitioned to IETF

Administration LLC. The business records of the RFC Editor function, as it was conducted by ISI, are currently housed with a cloud vendor under contract with IETF Administration LLC.

5. I make this declaration based on my personal knowledge and information contained in the business records of the RFC Editor as they are currently housed by AMS with a cloud vendor under contract with IETF Administration LLC, or in confirmation with other responsible RFC Editor personnel with such knowledge.

6. Prior to 1998, the RFC Editor's regular practice was to publish RFCs, making them available from a repository via FTP. When a new RFC was published, an announcement of its publication, with information on how to access the RFC, would be typically sent out within 24 hours of the publication.

7. Since 1998, the RFC Editor's regular practice was to publish RFCs, making them available on the RFC Editor website or via FTP. When a new RFC was published, an announcement of its publication, with information on how to access the RFC, would be typically sent out within 24 hours of the publication. The announcement would go out to all subscribers and a contemporaneous electronic record of the announcement is kept in the IETF mail archive that is available online.

8. Beginning in 1998, any RFC published on the RFC Editor website or via FTP was reasonably accessible to the public and was disseminated or otherwise available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable diligence could have located it. In particular, the RFCs were indexed and placed in a public repository.

9. The RFCs are kept in an online repository in the course of the RFC Editor's

regularly conducted activity and ordinary course of business. The records are made pursuant to established procedures and are relied upon by the RFC Editor in the performance of its functions.

10. It is the regular practice of the RFC Editor to make and keep the RFC records.

11. Based on the business records for the RFC Editor and the RFC Editor's course of conduct in publishing RFCs, I have determined that the publication date of RFC 3031 was no later than January, 2001, at which time it was reasonably accessible to the public either on the RFC Editor website or via FTP from a repository. An announcement of its publication also would have been sent out to subscribers within 24 hours of its publication. A copy of that RFC is attached to this declaration as **Exhibit 1**.

12. Based on the business records for the RFC Editor and the RFC Editor's course of conduct in publishing RFCs, I have determined that the publication date of RFC 3272 was no later than May, 2002, at which time it was reasonably accessible to the public either on the RFC Editor website or via FTP from a repository. An announcement of its publication also would have been sent out to subscribers within 24 hours of its publication. A copy of that RFC is attached to this declaration as **Exhibit 2**.

13. Based on the business records for the RFC Editor and the RFC Editor's course of conduct in publishing RFCs, I have determined that the publication date of RFC 3386 was no later than November, 2002, at which time it was reasonably accessible to the public either on the RFC Editor website or via FTP from a repository. An announcement of its publication also would have been sent out to subscribers within 24 hours of its publication. A copy of that RFC is attached to this declaration as **Exhibit 3.**

14. Based on the business records for the RFC Editor and the RFC Editor's course of conduct in publishing RFCs, I have determined that the publication date of RFC 3469 was no later than February, 2003, at which time it was reasonably accessible to the public either on the RFC Editor website or via FTP from a repository. An announcement of its publication also would have been sent out to subscribers within 24 hours of its publication. A copy of that RFC is attached to this declaration as **Exhibit 4**.

15. Based on the business records for the RFC Editor and the RFC Editor's course of conduct in publishing RFCs, I have determined that the publication date of RFC 3916 was no later than September, 2004, at which time it was reasonably accessible to the public either on the RFC Editor website or via FTP from a repository. An announcement of its publication also would have been sent out to subscribers within 24 hours of its publication. A copy of that RFC is attached to this declaration as **Exhibit 5**.

16. Based on the business records for the RFC Editor and the RFC Editor's course of conduct in publishing RFCs, I have determined that the publication date of RFC 3985 was no later than March, 2005, at which time it was reasonably accessible to the public either on the RFC Editor website or via FTP from a repository. An announcement of its publication also would have been sent out to subscribers within 24 hours of its publication. A copy of that RFC is attached to this declaration as **Exhibit 6**.

17. Based on the business records for the RFC Editor and the RFC Editor's course of conduct in publishing RFCs, I have determined that the publication date of RFC 4447 was no later than April, 2006, at which time it was reasonably accessible to the public either on the RFC Editor website or via FTP from a repository. An announcement of its publication also would have been sent out to subscribers within 24 hours of its

publication. A copy of that RFC is attached to this declaration as **Exhibit 7**.

18. Based on the business records for the RFC Editor and the RFC Editor's course of conduct in publishing RFCs, I have determined that the publication date of RFC 4448 was no later than April, 2006, at which time it was reasonably accessible to the public either on the RFC Editor website or via FTP from a repository. An announcement of its publication also would have been sent out to subscribers within 24 hours of its publication. A copy of that RFC is attached to this declaration as **Exhibit 8**.

19. Based on the business records for the RFC Editor and the RFC Editor's course of conduct in publishing RFCs, I have determined that the publication date of RFC 4619 was no later than September, 2006, at which time it was reasonably accessible to the public either on the RFC Editor website or via FTP from a repository. An announcement of its publication also would have been sent out to subscribers within 24 hours of its publication. A copy of that RFC is attached to this declaration as **Exhibit 9**.

20. Based on the business records for the RFC Editor and the RFC Editor's course of conduct in publishing RFCs, I have determined that the publication date of RFC 5659 was no later than October, 2009, at which time it was reasonably accessible to the public either on the RFC Editor website or via FTP from a repository. An announcement of its publication also would have been sent out to subscribers within 24 hours of its publication. A copy of that RFC is attached to this declaration as **Exhibit 10**.

[REMAINDER OF PAGE LEFT INTENTIONALLY BLANK]

PURSUANT TO SECTION 1746 OF TITLE 28 OF UNITED STATES CODE, I DECLARE UNDER PENALTY OF PERJURY UNDER THE LAWS OF THE UNITED STATES OF AMERICA THAT THE FOREGOING IS TRUE AND CORRECT AND THAT THE FOREGOING IS BASED UPON PERSONAL KNOWLEDGE AND INFORMATION AND IS BELIEVED TO BE TRUE.

Date: 29 July 2025

By: _____
Sandy Ginoza

Exhibit 1

Network Working Group
Request for Comments: 3031
Category: Standards Track

E. Rosen
Cisco Systems, Inc.
A. Viswanathan
Force10 Networks, Inc.
R. Callon
Juniper Networks, Inc.
January 2001

Multiprotocol Label Switching Architecture

Status of this Memo

Copyright Notice

Abstract

This document specifies the architecture for Multiprotocol Label Switching (MPLS).

Table of Contents

1. Specification

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

2. Introduction to MPLS

This document specifies the architecture for Multiprotocol Label Switching (MPLS).

Note that the use of MPLS for multicast is left for further study.

2.1. Overview

As a packet of a connectionless network layer protocol travels from one router to the next, each router makes an independent forwarding decision for that packet. That is, each router analyzes the packet's header, and each router runs a network layer routing algorithm. Each router independently chooses a next hop for the packet, based on its analysis of the packet's header and the results of running the routing algorithm.

Packet headers contain considerably more information than is needed simply to choose the next hop. Choosing the next hop can therefore be thought of as the composition of two functions. The first function partitions the entire set of possible packets into a set of "Forwarding Equivalence Classes (FECs)". The second maps each FEC to a next hop. Insofar as the forwarding decision is concerned, different packets which get mapped into the same FEC are indistinguishable. All packets which belong to a particular FEC and which travel from a particular node will follow the same path (or if certain kinds of multi-path routing are in use, they will all follow one of a set of paths associated with the FEC).

In conventional IP forwarding, a particular router will typically consider two packets to be in the same FEC if there is some address prefix X in that router's routing tables such that X is the "longest match" for each packet's destination address. As the packet traverses the network, each hop in turn reexamines the packet and assigns it to a FEC.

In MPLS, the assignment of a particular packet to a particular FEC is done just once, as the packet enters the network. The FEC to which the packet is assigned is encoded as a short fixed length value known as a "label". When a packet is forwarded to its next hop, the label is sent along with it; that is, the packets are "labeled" before they are forwarded.

At subsequent hops, there is no further analysis of the packet's network layer header. Rather, the label is used as an index into a table which specifies the next hop, and a new label. The old label is replaced with the new label, and the packet is forwarded to its next hop.

In the MPLS forwarding paradigm, once a packet is assigned to a FEC, no further header analysis is done by subsequent routers; all forwarding is driven by the labels. This has a number of advantages over conventional network layer forwarding.

- MPLS forwarding can be done by switches which are capable of doing label lookup and replacement, but are either not capable of analyzing the network layer headers, or are not capable of analyzing the network layer headers at adequate speed.

- Since a packet is assigned to a FEC when it enters the network, the ingress router may use, in determining the assignment, any information it has about the packet, even if that information cannot be gleaned from the network layer header. For example, packets arriving on different ports may be assigned to different FECs. Conventional forwarding, on the other hand, can only consider information which travels with the packet in the packet header.

- A packet that enters the network at a particular router can be labeled differently than the same packet entering the network at a different router, and as a result forwarding decisions that depend on the ingress router can be easily made. This cannot be done with conventional forwarding, since the identity of a packet's ingress router does not travel with the packet.

- The considerations that determine how a packet is assigned to a FEC can become ever more and more complicated, without any impact at all on the routers that merely forward labeled packets.

- Sometimes it is desirable to force a packet to follow a particular route which is explicitly chosen at or before the time the packet enters the network, rather than being chosen by the normal dynamic routing algorithm as the packet travels through the network. This may be done as a matter of policy, or to support traffic engineering. In conventional forwarding, this requires the packet to carry an encoding of its route along with it ("source routing"). In MPLS, a label can be used to represent the route, so that the identity of the explicit route need not be carried with the packet.

Some routers analyze a packet's network layer header not merely to choose the packet's next hop, but also to determine a packet's "precedence" or "class of service". They may then apply different discard thresholds or scheduling disciplines to different packets. MPLS allows (but does not require) the precedence or class of service to be fully or partially inferred from the label. In this case, one may say that the label represents the combination of a FEC and a precedence or class of service.

MPLS stands for "Multiprotocol" Label Switching, multiprotocol because its techniques are applicable to ANY network layer protocol. In this document, however, we focus on the use of IP as the network layer protocol.

A router which supports MPLS is known as a "Label Switching Router", or LSR.

2.2. Terminology

This section gives a general conceptual overview of the terms used in this document. Some of these terms are more precisely defined in later sections of the document.

DLCI	a label used in Frame Relay networks to identify frame relay circuits
forwarding equivalence class	a group of IP packets which are forwarded in the same manner (e.g., over the same path, with the same forwarding treatment)
frame merge	label merging, when it is applied to operation over frame based media, so that the potential problem of cell interleave is not an issue.
label	a short fixed length physically contiguous identifier which is used to identify a FEC, usually of local significance.
label merging	the replacement of multiple incoming labels for a particular FEC with a single outgoing label
label swap	the basic forwarding operation consisting of looking up an incoming label to determine the outgoing label, encapsulation, port, and other data handling information.
label swapping	a forwarding paradigm allowing streamlined forwarding of data by using labels to identify classes of data packets which are treated indistinguishably when forwarding.

label switched hop	the hop between two MPLS nodes, on which forwarding is done using labels.
label switched path	The path through one or more LSRs at one level of the hierarchy followed by a packets in a particular FEC.
label switching router	an MPLS node which is capable of forwarding native L3 packets
layer 2	the protocol layer under layer 3 (which therefore offers the services used by layer 3). Forwarding, when done by the swapping of short fixed length labels, occurs at layer 2 regardless of whether the label being examined is an ATM VPI/VCI, a frame relay DLCI, or an MPLS label.
layer 3	the protocol layer at which IP and its associated routing protocols operate link layer synonymous with layer 2
loop detection	a method of dealing with loops in which loops are allowed to be set up, and data may be transmitted over the loop, but the loop is later detected
loop prevention	a method of dealing with loops in which data is never transmitted over a loop
label stack	an ordered set of labels
merge point	a node at which label merging is done
MPLS domain	a contiguous set of nodes which operate MPLS routing and forwarding and which are also in one Routing or Administrative Domain
MPLS edge node	an MPLS node that connects an MPLS domain with a node which is outside of the domain, either because it does not run MPLS, and/or because it is in a different domain. Note that if an LSR has a neighboring host which is not running MPLS, that that LSR is an MPLS edge node.

MPLS egress node	an MPLS edge node in its role in handling traffic as it leaves an MPLS domain
MPLS ingress node	an MPLS edge node in its role in handling traffic as it enters an MPLS domain
MPLS label	a label which is carried in a packet header, and which represents the packet's FEC
MPLS node	a node which is running MPLS. An MPLS node will be aware of MPLS control protocols, will operate one or more L3 routing protocols, and will be capable of forwarding packets based on labels. An MPLS node may optionally be also capable of forwarding native L3 packets.
MultiProtocol Label Switching	an IETF working group and the effort associated with the working group
network layer	synonymous with layer 3
stack	synonymous with label stack
switched path	synonymous with label switched path
virtual circuit	a circuit used by a connection-oriented layer 2 technology such as ATM or Frame Relay, requiring the maintenance of state information in layer 2 switches.
VC merge	label merging where the MPLS label is carried in the ATM VCI field (or combined VPI/VCI field), so as to allow multiple VCs to merge into one single VC
VP merge	label merging where the MPLS label is carried din the ATM VPI field, so as to allow multiple VPs to be merged into one single VP. In this case two cells would have the same VCI value only if they originated from the same node. This allows cells from different sources to be distinguished via the VCI.

VPI/VCI	a label used in ATM networks to identify circuits

2.3. Acronyms and Abbreviations

ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
DLCI	Data Link Circuit Identifier
FEC	Forwarding Equivalence Class
FTN	FEC to NHLFE Map
IGP	Interior Gateway Protocol
ILM	Incoming Label Map
IP	Internet Protocol
LDP	Label Distribution Protocol
L2	Layer 2 L3 Layer 3
LSP	Label Switched Path
LSR	Label Switching Router
MPLS	MultiProtocol Label Switching
NHLFE	Next Hop Label Forwarding Entry
SVC	Switched Virtual Circuit
SVP	Switched Virtual Path
TTL	Time-To-Live
VC	Virtual Circuit
VCI	Virtual Circuit Identifier
VP	Virtual Path
VPI	Virtual Path Identifier

2.4. Acknowledgments

The ideas and text in this document have been collected from a number of sources and comments received. We would like to thank Rick Boivie, Paul Doolan, Nancy Feldman, Yakov Rekhter, Vijay Srinivasan, and George Swallow for their inputs and ideas.

3. MPLS Basics

In this section, we introduce some of the basic concepts of MPLS and describe the general approach to be used.

3.1. Labels

A label is a short, fixed length, locally significant identifier which is used to identify a FEC. The label which is put on a particular packet represents the Forwarding Equivalence Class to which that packet is assigned.

Most commonly, a packet is assigned to a FEC based (completely or partially) on its network layer destination address. However, the label is never an encoding of that address.

If Ru and Rd are LSRs, they may agree that when Ru transmits a packet to Rd, Ru will label with packet with label value L if and only if the packet is a member of a particular FEC F. That is, they can agree to a "binding" between label L and FEC F for packets moving from Ru to Rd. As a result of such an agreement, L becomes Ru's "outgoing label" representing FEC F, and L becomes Rd's "incoming label" representing FEC F.

Note that L does not necessarily represent FEC F for any packets other than those which are being sent from Ru to Rd. L is an arbitrary value whose binding to F is local to Ru and Rd.

When we speak above of packets "being sent" from Ru to Rd, we do not imply either that the packet originated at Ru or that its destination is Rd. Rather, we mean to include packets which are "transit packets" at one or both of the LSRs.

Sometimes it may be difficult or even impossible for Rd to tell, of an arriving packet carrying label L, that the label L was placed in the packet by Ru, rather than by some other LSR. (This will typically be the case when Ru and Rd are not direct neighbors.) In such cases, Rd must make sure that the binding from label to FEC is one-to-one. That is, Rd MUST NOT agree with Ru1 to bind L to FEC F1, while also agreeing with some other LSR Ru2 to bind L to a different FEC F2, UNLESS Rd can always tell, when it receives a packet with incoming label L, whether the label was put on the packet by Ru1 or whether it was put on by Ru2.

It is the responsibility of each LSR to ensure that it can uniquely interpret its incoming labels.

3.2. Upstream and Downstream LSRs

Suppose Ru and Rd have agreed to bind label L to FEC F, for packets sent from Ru to Rd. Then with respect to this binding, Ru is the "upstream LSR", and Rd is the "downstream LSR".

To say that one node is upstream and one is downstream with respect to a given binding means only that a particular label represents a particular FEC in packets travelling from the upstream node to the downstream node. This is NOT meant to imply that packets in that FEC would actually be routed from the upstream node to the downstream node.

3.3. Labeled Packet

A "labeled packet" is a packet into which a label has been encoded. In some cases, the label resides in an encapsulation header which exists specifically for this purpose. In other cases, the label may reside in an existing data link or network layer header, as long as there is a field which is available for that purpose. The particular encoding technique to be used must be agreed to by both the entity which encodes the label and the entity which decodes the label.

3.4. Label Assignment and Distribution

In the MPLS architecture, the decision to bind a particular label L to a particular FEC F is made by the LSR which is DOWNSTREAM with respect to that binding. The downstream LSR then informs the upstream LSR of the binding. Thus labels are "downstream-assigned", and label bindings are distributed in the "downstream to upstream" direction.

If an LSR has been designed so that it can only look up labels that fall into a certain numeric range, then it merely needs to ensure that it only binds labels that are in that range.

3.5. Attributes of a Label Binding

A particular binding of label L to FEC F, distributed by Rd to Ru, may have associated "attributes". If Ru, acting as a downstream LSR, also distributes a binding of a label to FEC F, then under certain conditions, it may be required to also distribute the corresponding attribute that it received from Rd.

3.6. Label Distribution Protocols

A label distribution protocol is a set of procedures by which one LSR informs another of the label/FEC bindings it has made. Two LSRs which use a label distribution protocol to exchange label/FEC binding information are known as "label distribution peers" with respect to the binding information they exchange. If two LSRs are label distribution peers, we will speak of there being a "label distribution adjacency" between them.

(N.B.: two LSRs may be label distribution peers with respect to some set of bindings, but not with respect to some other set of bindings.)

The label distribution protocol also encompasses any negotiations in which two label distribution peers need to engage in order to learn of each other’s MPLS capabilities.

THE ARCHITECTURE DOES NOT ASSUME THAT THERE IS ONLY A SINGLE LABEL DISTRIBUTION PROTOCOL. In fact, a number of different label distribution protocols are being standardized. Existing protocols have been extended so that label distribution can be piggybacked on them (see, e.g., [MPLS-BGP], [MPLS-RSVP-TUNNELS]). New protocols have also been defined for the explicit purpose of distributing labels (see, e.g., [MPLS-LDP], [MPLS-CR-LDP].

In this document, we try to use the acronym "LDP" to refer specifically to the protocol defined in [MPLS-LDP]; when speaking of label distribution protocols in general, we try to avoid the acronym.

3.7. Unsolicited Downstream vs. Downstream-on-Demand

The MPLS architecture allows an LSR to explicitly request, from its next hop for a particular FEC, a label binding for that FEC. This is known as "downstream-on-demand" label distribution.

The MPLS architecture also allows an LSR to distribute bindings to LSRs that have not explicitly requested them. This is known as "unsolicited downstream" label distribution.

It is expected that some MPLS implementations will provide only downstream-on-demand label distribution, and some will provide only unsolicited downstream label distribution, and some will provide both. Which is provided may depend on the characteristics of the interfaces which are supported by a particular implementation. However, both of these label distribution techniques may be used in the same network at the same time. On any given label distribution adjacency, the upstream LSR and the downstream LSR must agree on which technique is to be used.

3.8. Label Retention Mode

An LSR Ru may receive (or have received) a label binding for a particular FEC from an LSR Rd, even though Rd is not Ru’s next hop (or is no longer Ru’s next hop) for that FEC.

Ru then has the choice of whether to keep track of such bindings, or whether to discard such bindings. If Ru keeps track of such bindings, then it may immediately begin using the binding again if Rd eventually becomes its next hop for the FEC in question. If Ru discards such bindings, then if Rd later becomes the next hop, the binding will have to be reacquired.

If an LSR supports "Liberal Label Retention Mode", it maintains the bindings between a label and a FEC which are received from LSRs which are not its next hop for that FEC. If an LSR supports "Conservative Label Retention Mode", it discards such bindings.

Liberal label retention mode allows for quicker adaptation to routing changes, but conservative label retention mode though requires an LSR to maintain many fewer labels.

3.9. The Label Stack

So far, we have spoken as if a labeled packet carries only a single label. As we shall see, it is useful to have a more general model in which a labeled packet carries a number of labels, organized as a last-in, first-out stack. We refer to this as a "label stack".

Although, as we shall see, MPLS supports a hierarchy, the processing of a labeled packet is completely independent of the level of hierarchy. The processing is always based on the top label, without regard for the possibility that some number of other labels may have been "above it" in the past, or that some number of other labels may be below it at present.

An unlabeled packet can be thought of as a packet whose label stack is empty (i.e., whose label stack has depth 0).

If a packet's label stack is of depth m, we refer to the label at the bottom of the stack as the level 1 label, to the label above it (if such exists) as the level 2 label, and to the label at the top of the stack as the level m label.

The utility of the label stack will become clear when we introduce the notion of LSP Tunnel and the MPLS Hierarchy (section 3.27).

3.10. The Next Hop Label Forwarding Entry (NHLFE)

The "Next Hop Label Forwarding Entry" (NHLFE) is used when forwarding a labeled packet. It contains the following information:

1. the packet's next hop

2. the operation to perform on the packet's label stack; this is one of the following operations:

   a) replace the label at the top of the label stack with a specified new label

   b) pop the label stack

c) replace the label at the top of the label stack with a specified new label, and then push one or more specified new labels onto the label stack.

It may also contain:

d) the data link encapsulation to use when transmitting the packet

e) the way to encode the label stack when transmitting the packet

f) any other information needed in order to properly dispose of the packet.

Note that at a given LSR, the packet's "next hop" might be that LSR itself. In this case, the LSR would need to pop the top level label, and then "forward" the resulting packet to itself. It would then make another forwarding decision, based on what remains after the label stacked is popped. This may still be a labeled packet, or it may be the native IP packet.

This implies that in some cases the LSR may need to operate on the IP header in order to forward the packet.

If the packet's "next hop" is the current LSR, then the label stack operation MUST be to "pop the stack".

3.11. Incoming Label Map (ILM)

The "Incoming Label Map" (ILM) maps each incoming label to a set of NHLFEs. It is used when forwarding packets that arrive as labeled packets.

If the ILM maps a particular label to a set of NHLFEs that contains more than one element, exactly one element of the set must be chosen before the packet is forwarded. The procedures for choosing an element from the set are beyond the scope of this document. Having the ILM map a label to a set containing more than one NHLFE may be useful if, e.g., it is desired to do load balancing over multiple equal-cost paths.

3.12. FEC-to-NHLFE Map (FTN)

The "FEC-to-NHLFE" (FTN) maps each FEC to a set of NHLFEs. It is used when forwarding packets that arrive unlabeled, but which are to be labeled before being forwarded.

If the FTN maps a particular label to a set of NHLFEs that contains more than one element, exactly one element of the set must be chosen before the packet is forwarded. The procedures for choosing an element from the set are beyond the scope of this document. Having the FTN map a label to a set containing more than one NHLFE may be useful if, e.g., it is desired to do load balancing over multiple equal-cost paths.

3.13. Label Swapping

Label swapping is the use of the following procedures to forward a packet.

In order to forward a labeled packet, a LSR examines the label at the top of the label stack. It uses the ILM to map this label to an NHLFE. Using the information in the NHLFE, it determines where to forward the packet, and performs an operation on the packet's label stack. It then encodes the new label stack into the packet, and forwards the result.

In order to forward an unlabeled packet, a LSR analyzes the network layer header, to determine the packet's FEC. It then uses the FTN to map this to an NHLFE. Using the information in the NHLFE, it determines where to forward the packet, and performs an operation on the packet's label stack. (Popping the label stack would, of course, be illegal in this case.) It then encodes the new label stack into the packet, and forwards the result.

IT IS IMPORTANT TO NOTE THAT WHEN LABEL SWAPPING IS IN USE, THE NEXT HOP IS ALWAYS TAKEN FROM THE NHLFE; THIS MAY IN SOME CASES BE DIFFERENT FROM WHAT THE NEXT HOP WOULD BE IF MPLS WERE NOT IN USE.

3.14. Scope and Uniqueness of Labels

A given LSR Rd may bind label L1 to FEC F, and distribute that binding to label distribution peer Ru1. Rd may also bind label L2 to FEC F, and distribute that binding to label distribution peer Ru2. Whether or not L1 == L2 is not determined by the architecture; this is a local matter.

A given LSR Rd may bind label L to FEC F1, and distribute that binding to label distribution peer Ru1. Rd may also bind label L to FEC F2, and distribute that binding to label distribution peer Ru2. IF (AND ONLY IF) RD CAN TELL, WHEN IT RECEIVES A PACKET WHOSE TOP LABEL IS L, WHETHER THE LABEL WAS PUT THERE BY RU1 OR BY RU2, THEN THE ARCHITECTURE DOES NOT REQUIRE THAT F1 == F2. In such cases, we may say that Rd is using a different "label space" for the labels it distributes to Ru1 than for the labels it distributes to Ru2.

In general, Rd can only tell whether it was Ru1 or Ru2 that put the particular label value L at the top of the label stack if the following conditions hold:

- Ru1 and Ru2 are the only label distribution peers to which Rd distributed a binding of label value L, and

- Ru1 and Ru2 are each directly connected to Rd via a point-to-point interface.

When these conditions hold, an LSR may use labels that have "per interface" scope, i.e., which are only unique per interface. We may say that the LSR is using a "per-interface label space". When these conditions do not hold, the labels must be unique over the LSR which has assigned them, and we may say that the LSR is using a "per-platform label space."

If a particular LSR Rd is attached to a particular LSR Ru over two point-to-point interfaces, then Rd may distribute to Ru a binding of label L to FEC F1, as well as a binding of label L to FEC F2, F1 != F2, if and only if each binding is valid only for packets which Ru sends to Rd over a particular one of the interfaces. In all other cases, Rd MUST NOT distribute to Ru bindings of the same label value to two different FECs.

This prohibition holds even if the bindings are regarded as being at different "levels of hierarchy". In MPLS, there is no notion of having a different label space for different levels of the hierarchy; when interpreting a label, the level of the label is irrelevant.

The question arises as to whether it is possible for an LSR to use multiple per-platform label spaces, or to use multiple per-interface label spaces for the same interface. This is not prohibited by the architecture. However, in such cases the LSR must have some means, not specified by the architecture, of determining, for a particular incoming label, which label space that label belongs to. For example, [MPLS-SHIM] specifies that a different label space is used for unicast packets than for multicast packets, and uses a data link layer codepoint to distinguish the two label spaces.

3.15. Label Switched Path (LSP), LSP Ingress, LSP Egress

A "Label Switched Path (LSP) of level m" for a particular packet P is a sequence of routers,

<R1, ..., Rn>

with the following properties:

1. R1, the "LSP Ingress", is an LSR which pushes a label onto P's label stack, resulting in a label stack of depth m;

2. For all i, 1<i<n, P has a label stack of depth m when received by LSR Ri;

3. At no time during P's transit from R1 to R[n-1] does its label stack ever have a depth of less than m;

4. For all i, 1<i<n: Ri transmits P to R[i+1] by means of MPLS, i.e., by using the label at the top of the label stack (the level m label) as an index into an ILM;

5. For all i, 1<i<n: if a system S receives and forwards P after P is transmitted by Ri but before P is received by R[i+1] (e.g., Ri and R[i+1] might be connected via a switched data link subnetwork, and S might be one of the data link switches), then S's forwarding decision is not based on the level m label, or on the network layer header. This may be because:

   a) the decision is not based on the label stack or the network layer header at all;

   b) the decision is based on a label stack on which additional labels have been pushed (i.e., on a level m+k label, where k>0).

In other words, we can speak of the level m LSP for Packet P as the sequence of routers:

1. which begins with an LSR (an "LSP Ingress") that pushes on a level m label,

2. all of whose intermediate LSRs make their forwarding decision by label Switching on a level m label,

3. which ends (at an "LSP Egress") when a forwarding decision is made by label Switching on a level m-k label, where k>0, or when a forwarding decision is made by "ordinary", non-MPLS forwarding procedures.

A consequence (or perhaps a presupposition) of this is that whenever an LSR pushes a label onto an already labeled packet, it needs to make sure that the new label corresponds to a FEC whose LSP Egress is the LSR that assigned the label which is now second in the stack.

We will call a sequence of LSRs the "LSP for a particular FEC F" if it is an LSP of level m for a particular packet P when P's level m label is a label corresponding to FEC F.

Consider the set of nodes which may be LSP ingress nodes for FEC F. Then there is an LSP for FEC F which begins with each of those nodes. If a number of those LSPs have the same LSP egress, then one can consider the set of such LSPs to be a tree, whose root is the LSP egress. (Since data travels along this tree towards the root, this may be called a multipoint-to-point tree.) We can thus speak of the "LSP tree" for a particular FEC F.

3.16. Penultimate Hop Popping

Note that according to the definitions of section 3.15, if <R1, ..., Rn> is a level m LSP for packet P, P may be transmitted from R[n-1] to Rn with a label stack of depth m-1. That is, the label stack may be popped at the penultimate LSR of the LSP, rather than at the LSP Egress.

From an architectural perspective, this is perfectly appropriate. The purpose of the level m label is to get the packet to Rn. Once R[n-1] has decided to send the packet to Rn, the label no longer has any function, and need no longer be carried.

There is also a practical advantage to doing penultimate hop popping. If one does not do this, then when the LSP egress receives a packet, it first looks up the top label, and determines as a result of that lookup that it is indeed the LSP egress. Then it must pop the stack, and examine what remains of the packet. If there is another label on the stack, the egress will look this up and forward the packet based on this lookup. (In this case, the egress for the packet's level m LSP is also an intermediate node for its level m-1 LSP.) If there is no other label on the stack, then the packet is forwarded according to its network layer destination address. Note that this would require the egress to do TWO lookups, either two label lookups or a label lookup followed by an address lookup.

If, on the other hand, penultimate hop popping is used, then when the penultimate hop looks up the label, it determines:

- that it is the penultimate hop, and
- who the next hop is.

The penultimate node then pops the stack, and forwards the packet based on the information gained by looking up the label that was previously at the top of the stack. When the LSP egress receives the

packet, the label which is now at the top of the stack will be the label which it needs to look up in order to make its own forwarding decision. Or, if the packet was only carrying a single label, the LSP egress will simply see the network layer packet, which is just what it needs to see in order to make its forwarding decision.

This technique allows the egress to do a single lookup, and also requires only a single lookup by the penultimate node.

The creation of the forwarding "fastpath" in a label switching product may be greatly aided if it is known that only a single lookup is ever required:

- the code may be simplified if it can assume that only a single lookup is ever needed

- the code can be based on a "time budget" that assumes that only a single lookup is ever needed.

In fact, when penultimate hop popping is done, the LSP Egress need not even be an LSR.

However, some hardware switching engines may not be able to pop the label stack, so this cannot be universally required. There may also be some situations in which penultimate hop popping is not desirable. Therefore the penultimate node pops the label stack only if this is specifically requested by the egress node, OR if the next node in the LSP does not support MPLS. (If the next node in the LSP does support MPLS, but does not make such a request, the penultimate node has no way of knowing that it in fact is the penultimate node.)

An LSR which is capable of popping the label stack at all MUST do penultimate hop popping when so requested by its downstream label distribution peer.

Initial label distribution protocol negotiations MUST allow each LSR to determine whether its neighboring LSRS are capable of popping the label stack. A LSR MUST NOT request a label distribution peer to pop the label stack unless it is capable of doing so.

It may be asked whether the egress node can always interpret the top label of a received packet properly if penultimate hop popping is used. As long as the uniqueness and scoping rules of section 3.14 are obeyed, it is always possible to interpret the top label of a received packet unambiguously.

3.17. LSP Next Hop

The LSP Next Hop for a particular labeled packet in a particular LSR is the LSR which is the next hop, as selected by the NHLFE entry used for forwarding that packet.

The LSP Next Hop for a particular FEC is the next hop as selected by the NHLFE entry indexed by a label which corresponds to that FEC.

Note that the LSP Next Hop may differ from the next hop which would be chosen by the network layer routing algorithm. We will use the term "L3 next hop" when we refer to the latter.

3.18. Invalid Incoming Labels

What should an LSR do if it receives a labeled packet with a particular incoming label, but has no binding for that label? It is tempting to think that the labels can just be removed, and the packet forwarded as an unlabeled IP packet. However, in some cases, doing so could cause a loop. If the upstream LSR thinks the label is bound to an explicit route, and the downstream LSR doesn't think the label is bound to anything, and if the hop by hop routing of the unlabeled IP packet brings the packet back to the upstream LSR, then a loop is formed.

It is also possible that the label was intended to represent a route which cannot be inferred from the IP header.

Therefore, when a labeled packet is received with an invalid incoming label, it MUST be discarded, UNLESS it is determined by some means (not within the scope of the current document) that forwarding it unlabeled cannot cause any harm.

3.19. LSP Control: Ordered versus Independent

Some FECs correspond to address prefixes which are distributed via a dynamic routing algorithm. The setup of the LSPs for these FECs can be done in one of two ways: Independent LSP Control or Ordered LSP Control.

In Independent LSP Control, each LSR, upon noting that it recognizes a particular FEC, makes an independent decision to bind a label to that FEC and to distribute that binding to its label distribution peers. This corresponds to the way that conventional IP datagram routing works; each node makes an independent decision as to how to treat each packet, and relies on the routing algorithm to converge rapidly so as to ensure that each datagram is correctly delivered.

In Ordered LSP Control, an LSR only binds a label to a particular FEC if it is the egress LSR for that FEC, or if it has already received a label binding for that FEC from its next hop for that FEC.

If one wants to ensure that traffic in a particular FEC follows a path with some specified set of properties (e.g., that the traffic does not traverse any node twice, that a specified amount of resources are available to the traffic, that the traffic follows an explicitly specified path, etc.) ordered control must be used. With independent control, some LSRs may begin label switching a traffic in the FEC before the LSP is completely set up, and thus some traffic in the FEC may follow a path which does not have the specified set of properties. Ordered control also needs to be used if the recognition of the FEC is a consequence of the setting up of the corresponding LSP.

Ordered LSP setup may be initiated either by the ingress or the egress.

Ordered control and independent control are fully interoperable. However, unless all LSRs in an LSP are using ordered control, the overall effect on network behavior is largely that of independent control, since one cannot be sure that an LSP is not used until it is fully set up.

This architecture allows the choice between independent control and ordered control to be a local matter. Since the two methods interwork, a given LSR need support only one or the other. Generally speaking, the choice of independent versus ordered control does not appear to have any effect on the label distribution mechanisms which need to be defined.

3.20. Aggregation

One way of partitioning traffic into FECs is to create a separate FEC for each address prefix which appears in the routing table. However, within a particular MPLS domain, this may result in a set of FECs such that all traffic in all those FECs follows the same route. For example, a set of distinct address prefixes might all have the same egress node, and label swapping might be used only to get the the traffic to the egress node. In this case, within the MPLS domain, the union of those FECs is itself a FEC. This creates a choice: should a distinct label be bound to each component FEC, or should a single label be bound to the union, and that label applied to all traffic in the union?

The procedure of binding a single label to a union of FECs which is itself a FEC (within some domain), and of applying that label to all

traffic in the union, is known as "aggregation". The MPLS architecture allows aggregation. Aggregation may reduce the number of labels which are needed to handle a particular set of packets, and may also reduce the amount of label distribution control traffic needed.

Given a set of FECs which are "aggregatable" into a single FEC, it is possible to (a) aggregate them into a single FEC, (b) aggregate them into a set of FECs, or (c) not aggregate them at all. Thus we can speak of the "granularity" of aggregation, with (a) being the "coarsest granularity", and (c) being the "finest granularity".

When order control is used, each LSR should adopt, for a given set of FECs, the granularity used by its next hop for those FECs.

When independent control is used, it is possible that there will be two adjacent LSRs, Ru and Rd, which aggregate some set of FECs differently.

If Ru has finer granularity than Rd, this does not cause a problem. Ru distributes more labels for that set of FECs than Rd does. This means that when Ru needs to forward labeled packets in those FECs to Rd, it may need to map n labels into m labels, where n > m. As an option, Ru may withdraw the set of n labels that it has distributed, and then distribute a set of m labels, corresponding to Rd's level of granularity. This is not necessary to ensure correct operation, but it does result in a reduction of the number of labels distributed by Ru, and Ru is not gaining any particular advantage by distributing the larger number of labels. The decision whether to do this or not is a local matter.

If Ru has coarser granularity than Rd (i.e., Rd has distributed n labels for the set of FECs, while Ru has distributed m, where n > m), it has two choices:

- It may adopt Rd's finer level of granularity. This would require it to withdraw the m labels it has distributed, and distribute n labels. This is the preferred option.

- It may simply map its m labels into a subset of Rd's n labels, if it can determine that this will produce the same routing. For example, suppose that Ru applies a single label to all traffic that needs to pass through a certain egress LSR, whereas Rd binds a number of different labels to such traffic, depending on the individual destination addresses of the packets. If Ru knows the address of the egress router, and if Rd has bound a label to the FEC which is identified by that address, then Ru can simply apply that label.

In any event, every LSR needs to know (by configuration) what granularity to use for labels that it assigns. Where ordered control is used, this requires each node to know the granularity only for FECs which leave the MPLS network at that node. For independent control, best results may be obtained by ensuring that all LSRs are consistently configured to know the granularity for each FEC. However, in many cases this may be done by using a single level of granularity which applies to all FECs (such as "one label per IP prefix in the forwarding table", or "one label per egress node").

3.21. Route Selection

Route selection refers to the method used for selecting the LSP for a particular FEC. The proposed MPLS protocol architecture supports two options for Route Selection: (1) hop by hop routing, and (2) explicit routing.

Hop by hop routing allows each node to independently choose the next hop for each FEC. This is the usual mode today in existing IP networks. A "hop by hop routed LSP" is an LSP whose route is selected using hop by hop routing.

In an explicitly routed LSP, each LSR does not independently choose the next hop; rather, a single LSR, generally the LSP ingress or the LSP egress, specifies several (or all) of the LSRs in the LSP. If a single LSR specifies the entire LSP, the LSP is "strictly" explicitly routed. If a single LSR specifies only some of the LSP, the LSP is "loosely" explicitly routed.

The sequence of LSRs followed by an explicitly routed LSP may be chosen by configuration, or may be selected dynamically by a single node (for example, the egress node may make use of the topological information learned from a link state database in order to compute the entire path for the tree ending at that egress node).

Explicit routing may be useful for a number of purposes, such as policy routing or traffic engineering. In MPLS, the explicit route needs to be specified at the time that labels are assigned, but the explicit route does not have to be specified with each IP packet. This makes MPLS explicit routing much more efficient than the alternative of IP source routing.

The procedures for making use of explicit routes, either strict or loose, are beyond the scope of this document.

3.22. Lack of Outgoing Label

When a labeled packet is traveling along an LSP, it may occasionally happen that it reaches an LSR at which the ILM does not map the packet's incoming label into an NHLFE, even though the incoming label is itself valid. This can happen due to transient conditions, or due to an error at the LSR which should be the packet's next hop.

It is tempting in such cases to strip off the label stack and attempt to forward the packet further via conventional forwarding, based on its network layer header. However, in general this is not a safe procedure:

- If the packet has been following an explicitly routed LSP, this could result in a loop.

- The packet's network header may not contain enough information to enable this particular LSR to forward it correctly.

Unless it can be determined (through some means outside the scope of this document) that neither of these situations obtains, the only safe procedure is to discard the packet.

3.23. Time-to-Live (TTL)

In conventional IP forwarding, each packet carries a "Time To Live" (TTL) value in its header. Whenever a packet passes through a router, its TTL gets decremented by 1; if the TTL reaches 0 before the packet has reached its destination, the packet gets discarded.

This provides some level of protection against forwarding loops that may exist due to misconfigurations, or due to failure or slow convergence of the routing algorithm. TTL is sometimes used for other functions as well, such as multicast scoping, and supporting the "traceroute" command. This implies that there are two TTL-related issues that MPLS needs to deal with: (i) TTL as a way to suppress loops; (ii) TTL as a way to accomplish other functions, such as limiting the scope of a packet.

When a packet travels along an LSP, it SHOULD emerge with the same TTL value that it would have had if it had traversed the same sequence of routers without having been label switched. If the packet travels along a hierarchy of LSPs, the total number of LSR-hops traversed SHOULD be reflected in its TTL value when it emerges from the hierarchy of LSPs.

The way that TTL is handled may vary depending upon whether the MPLS label values are carried in an MPLS-specific "shim" header [MPLS-SHIM], or if the MPLS labels are carried in an L2 header, such as an ATM header [MPLS-ATM] or a frame relay header [MPLS-FRMRLY].

If the label values are encoded in a "shim" that sits between the data link and network layer headers, then this shim MUST have a TTL field that SHOULD be initially loaded from the network layer header TTL field, SHOULD be decremented at each LSR-hop, and SHOULD be copied into the network layer header TTL field when the packet emerges from its LSP.

If the label values are encoded in a data link layer header (e.g., the VPI/VCI field in ATM's AAL5 header), and the labeled packets are forwarded by an L2 switch (e.g., an ATM switch), and the data link layer (like ATM) does not itself have a TTL field, then it will not be possible to decrement a packet's TTL at each LSR-hop. An LSP segment which consists of a sequence of LSRs that cannot decrement a packet's TTL will be called a "non-TTL LSP segment".

When a packet emerges from a non-TTL LSP segment, it SHOULD however be given a TTL that reflects the number of LSR-hops it traversed. In the unicast case, this can be achieved by propagating a meaningful LSP length to ingress nodes, enabling the ingress to decrement the TTL value before forwarding packets into a non-TTL LSP segment.

Sometimes it can be determined, upon ingress to a non-TTL LSP segment, that a particular packet's TTL will expire before the packet reaches the egress of that non-TTL LSP segment. In this case, the LSR at the ingress to the non-TTL LSP segment must not label switch the packet. This means that special procedures must be developed to support traceroute functionality, for example, traceroute packets may be forwarded using conventional hop by hop forwarding.

3.24. Loop Control

On a non-TTL LSP segment, by definition, TTL cannot be used to protect against forwarding loops. The importance of loop control may depend on the particular hardware being used to provide the LSR functions along the non-TTL LSP segment.

Suppose, for instance, that ATM switching hardware is being used to provide MPLS switching functions, with the label being carried in the VPI/VCI field. Since ATM switching hardware cannot decrement TTL, there is no protection against loops. If the ATM hardware is capable of providing fair access to the buffer pool for incoming cells carrying different VPI/VCI values, this looping may not have any deleterious effect on other traffic. If the ATM hardware cannot

provide fair buffer access of this sort, however, then even transient loops may cause severe degradation of the LSR's total performance.

Even if fair buffer access can be provided, it is still worthwhile to have some means of detecting loops that last "longer than possible". In addition, even where TTL and/or per-VC fair queuing provides a means for surviving loops, it still may be desirable where practical to avoid setting up LSPs which loop. All LSRs that may attach to non-TTL LSP segments will therefore be required to support a common technique for loop detection; however, use of the loop detection technique is optional. The loop detection technique is specified in [MPLS-ATM] and [MPLS-LDP].

3.25. Label Encodings

In order to transmit a label stack along with the packet whose label stack it is, it is necessary to define a concrete encoding of the label stack. The architecture supports several different encoding techniques; the choice of encoding technique depends on the particular kind of device being used to forward labeled packets.

3.25.1. MPLS-specific Hardware and/or Software

If one is using MPLS-specific hardware and/or software to forward labeled packets, the most obvious way to encode the label stack is to define a new protocol to be used as a "shim" between the data link layer and network layer headers. This shim would really be just an encapsulation of the network layer packet; it would be "protocol-independent" such that it could be used to encapsulate any network layer. Hence we will refer to it as the "generic MPLS encapsulation".

The generic MPLS encapsulation would in turn be encapsulated in a data link layer protocol.

The MPLS generic encapsulation is specified in [MPLS-SHIM].

3.25.2. ATM Switches as LSRs

It will be noted that MPLS forwarding procedures are similar to those of legacy "label swapping" switches such as ATM switches. ATM switches use the input port and the incoming VPI/VCI value as the index into a "cross-connect" table, from which they obtain an output port and an outgoing VPI/VCI value. Therefore if one or more labels can be encoded directly into the fields which are accessed by these legacy switches, then the legacy switches can, with suitable software upgrades, be used as LSRs. We will refer to such devices as "ATM-LSRs".

There are three obvious ways to encode labels in the ATM cell header (presuming the use of AAL5):

1. SVC Encoding

   Use the VPI/VCI field to encode the label which is at the top of the label stack. This technique can be used in any network. With this encoding technique, each LSP is realized as an ATM SVC, and the label distribution protocol becomes the ATM "signaling" protocol. With this encoding technique, the ATM-LSRs cannot perform "push" or "pop" operations on the label stack.

2. SVP Encoding

   Use the VPI field to encode the label which is at the top of the label stack, and the VCI field to encode the second label on the stack, if one is present. This technique some advantages over the previous one, in that it permits the use of ATM "VP-switching". That is, the LSPs are realized as ATM SVPs, with the label distribution protocol serving as the ATM signaling protocol.

   However, this technique cannot always be used. If the network includes an ATM Virtual Path through a non-MPLS ATM network, then the VPI field is not necessarily available for use by MPLS.

   When this encoding technique is used, the ATM-LSR at the egress of the VP effectively does a "pop" operation.

3. SVP Multipoint Encoding

   Use the VPI field to encode the label which is at the top of the label stack, use part of the VCI field to encode the second label on the stack, if one is present, and use the remainder of the VCI field to identify the LSP ingress. If this technique is used, conventional ATM VP-switching capabilities can be used to provide multipoint-to-point VPs. Cells from different packets will then carry different VCI values. As we shall see in section 3.26, this enables us to do label merging, without running into any cell interleaving problems, on ATM switches which can provide multipoint-to-point VPs, but which do not have the VC merge capability.

   This technique depends on the existence of a capability for assigning 16-bit VCI values to each ATM switch such that no single VCI value is assigned to two different switches. (If an

adequate number of such values could be assigned to each switch, it would be possible to also treat the VCI value as the second label in the stack.)

If there are more labels on the stack than can be encoded in the ATM header, the ATM encodings must be combined with the generic encapsulation.

3.25.3. Interoperability among Encoding Techniques

If <R1, R2, R3> is a segment of a LSP, it is possible that R1 will use one encoding of the label stack when transmitting packet P to R2, but R2 will use a different encoding when transmitting a packet P to R3. In general, the MPLS architecture supports LSPs with different label stack encodings used on different hops. Therefore, when we discuss the procedures for processing a labeled packet, we speak in abstract terms of operating on the packet's label stack. When a labeled packet is received, the LSR must decode it to determine the current value of the label stack, then must operate on the label stack to determine the new value of the stack, and then encode the new value appropriately before transmitting the labeled packet to its next hop.

Unfortunately, ATM switches have no capability for translating from one encoding technique to another. The MPLS architecture therefore requires that whenever it is possible for two ATM switches to be successive LSRs along a level m LSP for some packet, that those two ATM switches use the same encoding technique.

Naturally there will be MPLS networks which contain a combination of ATM switches operating as LSRs, and other LSRs which operate using an MPLS shim header. In such networks there may be some LSRs which have ATM interfaces as well as "MPLS Shim" interfaces. This is one example of an LSR with different label stack encodings on different hops. Such an LSR may swap off an ATM encoded label stack on an incoming interface and replace it with an MPLS shim header encoded label stack on the outgoing interface.

3.26. Label Merging

Suppose that an LSR has bound multiple incoming labels to a particular FEC. When forwarding packets in that FEC, one would like to have a single outgoing label which is applied to all such packets. The fact that two different packets in the FEC arrived with different incoming labels is irrelevant; one would like to forward them with the same outgoing label. The capability to do so is known as "label merging".

Let us say that an LSR is capable of label merging if it can receive two packets from different incoming interfaces, and/or with different labels, and send both packets out the same outgoing interface with the same label. Once the packets are transmitted, the information that they arrived from different interfaces and/or with different incoming labels is lost.

Let us say that an LSR is not capable of label merging if, for any two packets which arrive from different interfaces, or with different labels, the packets must either be transmitted out different interfaces, or must have different labels. ATM-LSRs using the SVC or SVP Encodings cannot perform label merging. This is discussed in more detail in the next section.

If a particular LSR cannot perform label merging, then if two packets in the same FEC arrive with different incoming labels, they must be forwarded with different outgoing labels. With label merging, the number of outgoing labels per FEC need only be 1; without label merging, the number of outgoing labels per FEC could be as large as the number of nodes in the network.

With label merging, the number of incoming labels per FEC that a particular LSR needs is never be larger than the number of label distribution adjacencies. Without label merging, the number of incoming labels per FEC that a particular LSR needs is as large as the number of upstream nodes which forward traffic in the FEC to the LSR in question. In fact, it is difficult for an LSR to even determine how many such incoming labels it must support for a particular FEC.

The MPLS architecture accommodates both merging and non-merging LSRs, but allows for the fact that there may be LSRs which do not support label merging. This leads to the issue of ensuring correct interoperation between merging LSRs and non-merging LSRs. The issue is somewhat different in the case of datagram media versus the case of ATM. The different media types will therefore be discussed separately.

3.26.1. Non-merging LSRs

The MPLS forwarding procedures is very similar to the forwarding procedures used by such technologies as ATM and Frame Relay. That is, a unit of data arrives, a label (VPI/VCI or DLCI) is looked up in a "cross-connect table", on the basis of that lookup an output port is chosen, and the label value is rewritten. In fact, it is possible to use such technologies for MPLS forwarding; a label distribution protocol can be used as the "signalling protocol" for setting up the cross-connect tables.

Unfortunately, these technologies do not necessarily support the label merging capability. In ATM, if one attempts to perform label merging, the result may be the interleaving of cells from various packets. If cells from different packets get interleaved, it is impossible to reassemble the packets. Some Frame Relay switches use cell switching on their backplanes. These switches may also be incapable of supporting label merging, for the same reason -- cells of different packets may get interleaved, and there is then no way to reassemble the packets.

We propose to support two solutions to this problem. First, MPLS will contain procedures which allow the use of non-merging LSRs. Second, MPLS will support procedures which allow certain ATM switches to function as merging LSRs.

Since MPLS supports both merging and non-merging LSRs, MPLS also contains procedures to ensure correct interoperation between them.

3.26.2. Labels for Merging and Non-Merging LSRs

An upstream LSR which supports label merging needs to be sent only one label per FEC. An upstream neighbor which does not support label merging needs to be sent multiple labels per FEC. However, there is no way of knowing a priori how many labels it needs. This will depend on how many LSRs are upstream of it with respect to the FEC in question.

In the MPLS architecture, if a particular upstream neighbor does not support label merging, it is not sent any labels for a particular FEC unless it explicitly asks for a label for that FEC. The upstream neighbor may make multiple such requests, and is given a new label each time. When a downstream neighbor receives such a request from upstream, and the downstream neighbor does not itself support label merging, then it must in turn ask its downstream neighbor for another label for the FEC in question.

It is possible that there may be some nodes which support label merging, but can only merge a limited number of incoming labels into a single outgoing label. Suppose for example that due to some hardware limitation a node is capable of merging four incoming labels into a single outgoing label. Suppose however, that this particular node has six incoming labels arriving at it for a particular FEC. In this case, this node may merge these into two outgoing labels.

Whether label merging is applicable to explicitly routed LSPs is for further study.

3.26.3. Merge over ATM

3.26.3.1. Methods of Eliminating Cell Interleave

There are several methods that can be used to eliminate the cell interleaving problem in ATM, thereby allowing ATM switches to support stream merge:

1. VP merge, using the SVP Multipoint Encoding

   When VP merge is used, multiple virtual paths are merged into a virtual path, but packets from different sources are distinguished by using different VCIs within the VP.

2. VC merge

   When VC merge is used, switches are required to buffer cells from one packet until the entire packet is received (this may be determined by looking for the AAL5 end of frame indicator).

VP merge has the advantage that it is compatible with a higher percentage of existing ATM switch implementations. This makes it more likely that VP merge can be used in existing networks. Unlike VC merge, VP merge does not incur any delays at the merge points and also does not impose any buffer requirements. However, it has the disadvantage that it requires coordination of the VCI space within each VP. There are a number of ways that this can be accomplished. Selection of one or more methods is for further study.

This tradeoff between compatibility with existing equipment versus protocol complexity and scalability implies that it is desirable for the MPLS protocol to support both VP merge and VC merge. In order to do so each ATM switch participating in MPLS needs to know whether its immediate ATM neighbors perform VP merge, VC merge, or no merge.

3.26.3.2. Interoperation: VC Merge, VP Merge, and Non-Merge

The interoperation of the various forms of merging over ATM is most easily described by first describing the interoperation of VC merge with non-merge.

In the case where VC merge and non-merge nodes are interconnected the forwarding of cells is based in all cases on a VC (i.e., the concatenation of the VPI and VCI). For each node, if an upstream neighbor is doing VC merge then that upstream neighbor requires only a single VPI/VCI for a particular stream (this is analogous to the requirement for a single label in the case of operation over frame media). If the upstream neighbor is not doing merge, then the

neighbor will require a single VPI/VCI per stream for itself, plus enough VPI/VCIs to pass to its upstream neighbors. The number required will be determined by allowing the upstream nodes to request additional VPI/VCIs from their downstream neighbors (this is again analogous to the method used with frame merge).

A similar method is possible to support nodes which perform VP merge. In this case the VP merge node, rather than requesting a single VPI/VCI or a number of VPI/VCIs from its downstream neighbor, instead may request a single VP (identified by a VPI) but several VCIs within the VP. Furthermore, suppose that a non-merge node is downstream from two different VP merge nodes. This node may need to request one VPI/VCI (for traffic originating from itself) plus two VPs (one for each upstream node), each associated with a specified set of VCIs (as requested from the upstream node).

In order to support all of VP merge, VC merge, and non-merge, it is therefore necessary to allow upstream nodes to request a combination of zero or more VC identifiers (consisting of a VPI/VCI), plus zero or more VPs (identified by VPIs) each containing a specified number of VCs (identified by a set of VCIs which are significant within a VP). VP merge nodes would therefore request one VP, with a contained VCI for traffic that it originates (if appropriate) plus a VCI for each VC requested from above (regardless of whether or not the VC is part of a containing VP). VC merge node would request only a single VPI/VCI (since they can merge all upstream traffic into a single VC). Non-merge nodes would pass on any requests that they get from above, plus request a VPI/VCI for traffic that they originate (if appropriate).

3.27. Tunnels and Hierarchy

Sometimes a router Ru takes explicit action to cause a particular packet to be delivered to another router Rd, even though Ru and Rd are not consecutive routers on the Hop-by-hop path for that packet, and Rd is not the packet’s ultimate destination. For example, this may be done by encapsulating the packet inside a network layer packet whose destination address is the address of Rd itself. This creates a "tunnel" from Ru to Rd. We refer to any packet so handled as a "Tunneled Packet".

3.27.1. Hop-by-Hop Routed Tunnel

If a Tunneled Packet follows the Hop-by-hop path from Ru to Rd, we say that it is in an "Hop-by-Hop Routed Tunnel" whose "transmit endpoint" is Ru and whose "receive endpoint" is Rd.

3.27.2. Explicitly Routed Tunnel

If a Tunneled Packet travels from Ru to Rd over a path other than the Hop-by-hop path, we say that it is in an "Explicitly Routed Tunnel" whose "transmit endpoint" is Ru and whose "receive endpoint" is Rd. For example, we might send a packet through an Explicitly Routed Tunnel by encapsulating it in a packet which is source routed.

3.27.3. LSP Tunnels

It is possible to implement a tunnel as a LSP, and use label switching rather than network layer encapsulation to cause the packet to travel through the tunnel. The tunnel would be a LSP <R1, ..., Rn>, where R1 is the transmit endpoint of the tunnel, and Rn is the receive endpoint of the tunnel. This is called a "LSP Tunnel".

The set of packets which are to be sent though the LSP tunnel constitutes a FEC, and each LSR in the tunnel must assign a label to that FEC (i.e., must assign a label to the tunnel). The criteria for assigning a particular packet to an LSP tunnel is a local matter at the tunnel's transmit endpoint. To put a packet into an LSP tunnel, the transmit endpoint pushes a label for the tunnel onto the label stack and sends the labeled packet to the next hop in the tunnel.

If it is not necessary for the tunnel's receive endpoint to be able to determine which packets it receives through the tunnel, as discussed earlier, the label stack may be popped at the penultimate LSR in the tunnel.

A "Hop-by-Hop Routed LSP Tunnel" is a Tunnel that is implemented as an hop-by-hop routed LSP between the transmit endpoint and the receive endpoint.

An "Explicitly Routed LSP Tunnel" is a LSP Tunnel that is also an Explicitly Routed LSP.

3.27.4. Hierarchy: LSP Tunnels within LSPs

Consider a LSP <R1, R2, R3, R4>. Let us suppose that R1 receives unlabeled packet P, and pushes on its label stack the label to cause it to follow this path, and that this is in fact the Hop-by-hop path. However, let us further suppose that R2 and R3 are not directly connected, but are "neighbors" by virtue of being the endpoints of an LSP tunnel. So the actual sequence of LSRs traversed by P is <R1, R2, R21, R22, R23, R3, R4>.

When P travels from R1 to R2, it will have a label stack of depth 1. R2, switching on the label, determines that P must enter the tunnel. R2 first replaces the Incoming label with a label that is meaningful to R3. Then it pushes on a new label. This level 2 label has a value which is meaningful to R21. Switching is done on the level 2 label by R21, R22, R23. R23, which is the penultimate hop in the R2-R3 tunnel, pops the label stack before forwarding the packet to R3. When R3 sees packet P, P has only a level 1 label, having now exited the tunnel. Since R3 is the penultimate hop in P's level 1 LSP, it pops the label stack, and R4 receives P unlabeled.

The label stack mechanism allows LSP tunneling to nest to any depth.

3.27.5. Label Distribution Peering and Hierarchy

Suppose that packet P travels along a Level 1 LSP <R1, R2, R3, R4>, and when going from R2 to R3 travels along a Level 2 LSP <R2, R21, R22, R3>. From the perspective of the Level 2 LSP, R2's label distribution peer is R21. From the perspective of the Level 1 LSP, R2's label distribution peers are R1 and R3. One can have label distribution peers at each layer of hierarchy. We will see in sections 4.6 and 4.7 some ways to make use of this hierarchy. Note that in this example, R2 and R21 must be IGP neighbors, but R2 and R3 need not be.

When two LSRs are IGP neighbors, we will refer to them as "local label distribution peers". When two LSRs may be label distribution peers, but are not IGP neighbors, we will refer to them as "remote label distribution peers". In the above example, R2 and R21 are local label distribution peers, but R2 and R3 are remote label distribution peers.

The MPLS architecture supports two ways to distribute labels at different layers of the hierarchy: Explicit Peering and Implicit Peering.

One performs label distribution with one's local label distribution peer by sending label distribution protocol messages which are addressed to the peer. One can perform label distribution with one's remote label distribution peers in one of two ways:

1. Explicit Peering

   In explicit peering, one distributes labels to a peer by sending label distribution protocol messages which are addressed to the peer, exactly as one would do for local label distribution peers. This technique is most useful when the number of remote label distribution peers is small, or the

number of higher level label bindings is large, or the remote label distribution peers are in distinct routing areas or domains. Of course, one needs to know which labels to distribute to which peers; this is addressed in section 4.1.2.

Examples of the use of explicit peering is found in sections 4.2.1 and 4.6.

2. Implicit Peering

In Implicit Peering, one does not send label distribution protocol messages which are addressed to one's peer. Rather, to distribute higher level labels to ones remote label distribution peers, one encodes a higher level label as an attribute of a lower level label, and then distributes the lower level label, along with this attribute, to one's local label distribution peers. The local label distribution peers then propagate the information to their local label distribution peers. This process continues till the information reaches the remote peer.

This technique is most useful when the number of remote label distribution peers is large. Implicit peering does not require an n-square peering mesh to distribute labels to the remote label distribution peers because the information is piggybacked through the local label distribution peering. However, implicit peering requires the intermediate nodes to store information that they might not be directly interested in.

An example of the use of implicit peering is found in section 4.3.

3.28. Label Distribution Protocol Transport

A label distribution protocol is used between nodes in an MPLS network to establish and maintain the label bindings. In order for MPLS to operate correctly, label distribution information needs to be transmitted reliably, and the label distribution protocol messages pertaining to a particular FEC need to be transmitted in sequence. Flow control is also desirable, as is the capability to carry multiple label messages in a single datagram.

One way to meet these goals is to use TCP as the underlying transport, as is done in [MPLS-LDP] and [MPLS-BGP].

3.29. Why More than one Label Distribution Protocol?

This architecture does not establish hard and fast rules for choosing which label distribution protocol to use in which circumstances. However, it is possible to point out some of the considerations.

3.29.1. BGP and LDP

In many scenarios, it is desirable to bind labels to FECs which can be identified with routes to address prefixes (see section 4.1). If there is a standard, widely deployed routing algorithm which distributes those routes, it can be argued that label distribution is best achieved by piggybacking the label distribution on the distribution of the routes themselves.

For example, BGP distributes such routes, and if a BGP speaker needs to also distribute labels to its BGP peers, using BGP to do the label distribution (see [MPLS-BGP]) has a number of advantages. In particular, it permits BGP route reflectors to distribute labels, thus providing a significant scalability advantage over using LDP to distribute labels between BGP peers.

3.29.2. Labels for RSVP Flowspecs

When RSVP is used to set up resource reservations for particular flows, it can be desirable to label the packets in those flows, so that the RSVP filterspec does not need to be applied at each hop. It can be argued that having RSVP distribute the labels as part of its path/reservation setup process is the most efficient method of distributing labels for this purpose.

3.29.3. Labels for Explicitly Routed LSPs

In some applications of MPLS, particularly those related to traffic engineering, it is desirable to set up an explicitly routed path, from ingress to egress. It is also desirable to apply resource reservations along that path.

One can imagine two approaches to this:

- Start with an existing protocol that is used for setting up resource reservations, and extend it to support explicit routing and label distribution.

- Start with an existing protocol that is used for label distribution, and extend it to support explicit routing and resource reservations.

The first approach has given rise to the protocol specified in [MPLS-RSVP-TUNNELS], the second to the approach specified in [MPLS-CR-LDP].

3.30. Multicast

This section is for further study

4. Some Applications of MPLS

4.1. MPLS and Hop by Hop Routed Traffic

A number of uses of MPLS require that packets with a certain label be forwarded along the same hop-by-hop routed path that would be used for forwarding a packet with a specified address in its network layer destination address field.

4.1.1. Labels for Address Prefixes

In general, router R determines the next hop for packet P by finding the address prefix X in its routing table which is the longest match for P's destination address. That is, the packets in a given FEC are just those packets which match a given address prefix in R's routing table. In this case, a FEC can be identified with an address prefix.

Note that a packet P may be assigned to FEC F, and FEC F may be identified with address prefix X, even if P's destination address does not match X.

4.1.2. Distributing Labels for Address Prefixes

4.1.2.1. Label Distribution Peers for an Address Prefix

LSRs R1 and R2 are considered to be label distribution peers for address prefix X if and only if one of the following conditions holds:

1. R1's route to X is a route which it learned about via a particular instance of a particular IGP, and R2 is a neighbor of R1 in that instance of that IGP

2. R1's route to X is a route which it learned about by some instance of routing algorithm A1, and that route is redistributed into an instance of routing algorithm A2, and R2 is a neighbor of R1 in that instance of A2

3. R1 is the receive endpoint of an LSP Tunnel that is within another LSP, and R2 is a transmit endpoint of that tunnel, and R1 and R2 are participants in a common instance of an IGP, and are in the same IGP area (if the IGP in question has areas), and R1's route to X was learned via that IGP instance, or is redistributed by R1 into that IGP instance

4. R1's route to X is a route which it learned about via BGP, and R2 is a BGP peer of R1

In general, these rules ensure that if the route to a particular address prefix is distributed via an IGP, the label distribution peers for that address prefix are the IGP neighbors. If the route to a particular address prefix is distributed via BGP, the label distribution peers for that address prefix are the BGP peers. In other cases of LSP tunneling, the tunnel endpoints are label distribution peers.

4.1.2.2. Distributing Labels

In order to use MPLS for the forwarding of packets according to the hop-by-hop route corresponding to any address prefix, each LSR MUST:

1. bind one or more labels to each address prefix that appears in its routing table;

2. for each such address prefix X, use a label distribution protocol to distribute the binding of a label to X to each of its label distribution peers for X.

There is also one circumstance in which an LSR must distribute a label binding for an address prefix, even if it is not the LSR which bound that label to that address prefix:

3. If R1 uses BGP to distribute a route to X, naming some other LSR R2 as the BGP Next Hop to X, and if R1 knows that R2 has assigned label L to X, then R1 must distribute the binding between L and X to any BGP peer to which it distributes that route.

These rules ensure that labels corresponding to address prefixes which correspond to BGP routes are distributed to IGP neighbors if and only if the BGP routes are distributed into the IGP. Otherwise, the labels bound to BGP routes are distributed only to the other BGP speakers.

These rules are intended only to indicate which label bindings must be distributed by a given LSR to which other LSRs.

4.1.3. Using the Hop by Hop path as the LSP

If the hop-by-hop path that packet P needs to follow is <R1, ..., Rn>, then <R1, ..., Rn> can be an LSP as long as:

1. there is a single address prefix X, such that, for all i, 1<=i<n, X is the longest match in Ri's routing table for P's destination address;

2. for all i, 1<i<n, Ri has assigned a label to X and distributed that label to R[i-1].

Note that a packet's LSP can extend only until it encounters a router whose forwarding tables have a longer best match address prefix for the packet's destination address. At that point, the LSP must end and the best match algorithm must be performed again.

Suppose, for example, that packet P, with destination address 10.2.153.178 needs to go from R1 to R2 to R3. Suppose also that R2 advertises address prefix 10.2/16 to R1, but R3 advertises 10.2.153/23, 10.2.154/23, and 10.2/16 to R2. That is, R2 is advertising an "aggregated route" to R1. In this situation, packet P can be label Switched until it reaches R2, but since R2 has performed route aggregation, it must execute the best match algorithm to find P's FEC.

4.1.4. LSP Egress and LSP Proxy Egress

An LSR R is considered to be an "LSP Egress" LSR for address prefix X if and only if one of the following conditions holds:

1. R has an address Y, such that X is the address prefix in R's routing table which is the longest match for Y, or

2. R contains in its routing tables one or more address prefixes Y such that X is a proper initial substring of Y, but R's "LSP previous hops" for X do not contain any such address prefixes Y; that is, R is a "deaggregation point" for address prefix X.

An LSR R1 is considered to be an "LSP Proxy Egress" LSR for address prefix X if and only if:

1. R1's next hop for X is R2, and R1 and R2 are not label distribution peers with respect to X (perhaps because R2 does not support MPLS), or

2. R1 has been configured to act as an LSP Proxy Egress for X

The definition of LSP allows for the LSP Egress to be a node which does not support MPLS; in this case the penultimate node in the LSP is the Proxy Egress.

4.1.5. The Implicit NULL Label

The Implicit NULL label is a label with special semantics which an LSR can bind to an address prefix. If LSR Ru, by consulting its ILM, sees that labeled packet P must be forwarded next to Rd, but that Rd has distributed a binding of Implicit NULL to the corresponding address prefix, then instead of replacing the value of the label on top of the label stack, Ru pops the label stack, and then forwards the resulting packet to Rd.

LSR Rd distributes a binding between Implicit NULL and an address prefix X to LSR Ru if and only if:

1. the rules of Section 4.1.2 indicate that Rd distributes to Ru a label binding for X, and

2. Rd knows that Ru can support the Implicit NULL label (i.e., that it can pop the label stack), and

3. Rd is an LSP Egress (not proxy egress) for X.

This causes the penultimate LSR on a LSP to pop the label stack. This is quite appropriate; if the LSP Egress is an MPLS Egress for X, then if the penultimate LSR does not pop the label stack, the LSP Egress will need to look up the label, pop the label stack, and then look up the next label (or look up the L3 address, if no more labels are present). By having the penultimate LSR pop the label stack, the LSP Egress is saved the work of having to look up two labels in order to make its forwarding decision.

However, if the penultimate LSR is an ATM switch, it may not have the capability to pop the label stack. Hence a binding of Implicit NULL may be distributed only to LSRs which can support that function.

If the penultimate LSR in an LSP for address prefix X is an LSP Proxy Egress, it acts just as if the LSP Egress had distributed a binding of Implicit NULL for X.

4.1.6. Option: Egress-Targeted Label Assignment

There are situations in which an LSP Ingress, Ri, knows that packets of several different FECs must all follow the same LSP, terminating at, say, LSP Egress Re. In this case, proper routing can be achieved

by using a single label for all such FECs; it is not necessary to have a distinct label for each FEC. If (and only if) the following conditions hold:

1. the address of LSR Re is itself in the routing table as a "host route", and

2. there is some way for Ri to determine that Re is the LSP egress for all packets in a particular set of FECs

Then Ri may bind a single label to all FECS in the set. This is known as "Egress-Targeted Label Assignment."

How can LSR Ri determine that an LSR Re is the LSP Egress for all packets in a particular FEC? There are a number of possible ways:

- If the network is running a link state routing algorithm, and all nodes in the area support MPLS, then the routing algorithm provides Ri with enough information to determine the routers through which packets in that FEC must leave the routing domain or area.

- If the network is running BGP, Ri may be able to determine that the packets in a particular FEC must leave the network via some particular router which is the "BGP Next Hop" for that FEC.

- It is possible to use the label distribution protocol to pass information about which address prefixes are "attached" to which egress LSRs. This method has the advantage of not depending on the presence of link state routing.

If egress-targeted label assignment is used, the number of labels that need to be supported throughout the network may be greatly reduced. This may be significant if one is using legacy switching hardware to do MPLS, and the switching hardware can support only a limited number of labels.

One possible approach would be to configure the network to use egress-targeted label assignment by default, but to configure particular LSRs to NOT use egress-targeted label assignment for one or more of the address prefixes for which it is an LSP egress. We impose the following rule:

- If a particular LSR is NOT an LSP Egress for some set of address prefixes, then it should assign labels to the address prefixes in the same way as is done by its LSP next hop for those address prefixes. That is, suppose Rd is Ru’s LSP next

hop for address prefixes X1 and X2. If Rd assigns the same label to X1 and X2, Ru should as well. If Rd assigns different labels to X1 and X2, then Ru should as well.

For example, suppose one wants to make egress-targeted label assignment the default, but to assign distinct labels to those address prefixes for which there are multiple possible LSP egresses (i.e., for those address prefixes which are multi-homed.) One can configure all LSRs to use egress-targeted label assignment, and then configure a handful of LSRs to assign distinct labels to those address prefixes which are multi-homed. For a particular multi-homed address prefix X, one would only need to configure this in LSRs which are either LSP Egresses or LSP Proxy Egresses for X.

It is important to note that if Ru and Rd are adjacent LSRs in an LSP for X1 and X2, forwarding will still be done correctly if Ru assigns distinct labels to X1 and X2 while Rd assigns just one label to the both of them. This just means that R1 will map different incoming labels to the same outgoing label, an ordinary occurrence.

Similarly, if Rd assigns distinct labels to X1 and X2, but Ru assigns to them both the label corresponding to the address of their LSP Egress or Proxy Egress, forwarding will still be done correctly. Ru will just map the incoming label to the label which Rd has assigned to the address of that LSP Egress.

4.2. MPLS and Explicitly Routed LSPs

There are a number of reasons why it may be desirable to use explicit routing instead of hop by hop routing. For example, this allows routes to be based on administrative policies, and allows the routes that LSPs take to be carefully designed to allow traffic engineering [MPLS-TRFENG].

4.2.1. Explicitly Routed LSP Tunnels

In some situations, the network administrators may desire to forward certain classes of traffic along certain pre-specified paths, where these paths differ from the Hop-by-hop path that the traffic would ordinarily follow. This can be done in support of policy routing, or in support of traffic engineering. The explicit route may be a configured one, or it may be determined dynamically by some means, e.g., by constraint-based routing.

MPLS allows this to be easily done by means of Explicitly Routed LSP Tunnels. All that is needed is:

1. A means of selecting the packets that are to be sent into the Explicitly Routed LSP Tunnel;

2. A means of setting up the Explicitly Routed LSP Tunnel;

3. A means of ensuring that packets sent into the Tunnel will not loop from the receive endpoint back to the transmit endpoint.

If the transmit endpoint of the tunnel wishes to put a labeled packet into the tunnel, it must first replace the label value at the top of the stack with a label value that was distributed to it by the tunnel's receive endpoint. Then it must push on the label which corresponds to the tunnel itself, as distributed to it by the next hop along the tunnel. To allow this, the tunnel endpoints should be explicit label distribution peers. The label bindings they need to exchange are of no interest to the LSRs along the tunnel.

4.3. Label Stacks and Implicit Peering

Suppose a particular LSR Re is an LSP proxy egress for 10 address prefixes, and it reaches each address prefix through a distinct interface.

One could assign a single label to all 10 address prefixes. Then Re is an LSP egress for all 10 address prefixes. This ensures that packets for all 10 address prefixes get delivered to Re. However, Re would then have to look up the network layer address of each such packet in order to choose the proper interface to send the packet on.

Alternatively, one could assign a distinct label to each interface. Then Re is an LSP proxy egress for the 10 address prefixes. This eliminates the need for Re to look up the network layer addresses in order to forward the packets. However, it can result in the use of a large number of labels.

An alternative would be to bind all 10 address prefixes to the same level 1 label (which is also bound to the address of the LSR itself), and then to bind each address prefix to a distinct level 2 label. The level 2 label would be treated as an attribute of the level 1 label binding, which we call the "Stack Attribute". We impose the following rules:

- When LSR Ru initially labels a hitherto unlabeled packet, if the longest match for the packet's destination address is X, and Ru's LSP next hop for X is Rd, and Rd has distributed to Ru a binding of label L1 to X, along with a stack attribute of L2, then

1. Ru must push L2 and then L1 onto the packet's label stack, and then forward the packet to Rd;

2. When Ru distributes label bindings for X to its label distribution peers, it must include L2 as the stack attribute.

3. Whenever the stack attribute changes (possibly as a result of a change in Ru's LSP next hop for X), Ru must distribute the new stack attribute.

Note that although the label value bound to X may be different at each hop along the LSP, the stack attribute value is passed unchanged, and is set by the LSP proxy egress.

Thus the LSP proxy egress for X becomes an "implicit peer" with each other LSR in the routing area or domain. In this case, explicit peering would be too unwieldy, because the number of peers would become too large.

4.4. MPLS and Multi-Path Routing

If an LSR supports multiple routes for a particular stream, then it may assign multiple labels to the stream, one for each route. Thus the reception of a second label binding from a particular neighbor for a particular address prefix should be taken as meaning that either label can be used to represent that address prefix.

If multiple label bindings for a particular address prefix are specified, they may have distinct attributes.

4.5. LSP Trees as Multipoint-to-Point Entities

Consider the case of packets P1 and P2, each of which has a destination address whose longest match, throughout a particular routing domain, is address prefix X. Suppose that the Hop-by-hop path for P1 is <R1, R2, R3>, and the Hop-by-hop path for P2 is <R4, R2, R3>. Let's suppose that R3 binds label L3 to X, and distributes this binding to R2. R2 binds label L2 to X, and distributes this binding to both R1 and R4. When R2 receives packet P1, its incoming label will be L2. R2 will overwrite L2 with L3, and send P1 to R3. When R2 receives packet P2, its incoming label will also be L2. R2 again overwrites L2 with L3, and send P2 on to R3.

Note then that when P1 and P2 are traveling from R2 to R3, they carry the same label, and as far as MPLS is concerned, they cannot be distinguished. Thus instead of talking about two distinct LSPs, <R1,

R2, R3> and <R4, R2, R3>, we might talk of a single "Multipoint-to-Point LSP Tree", which we might denote as <{R1, R4}, R2, R3>.

This creates a difficulty when we attempt to use conventional ATM switches as LSRs. Since conventional ATM switches do not support multipoint-to-point connections, there must be procedures to ensure that each LSP is realized as a point-to-point VC. However, if ATM switches which do support multipoint-to-point VCs are in use, then the LSPs can be most efficiently realized as multipoint-to-point VCs. Alternatively, if the SVP Multipoint Encoding (section 3.25.2) can be used, the LSPs can be realized as multipoint-to-point SVPs.

4.6. LSP Tunneling between BGP Border Routers

Consider the case of an Autonomous System, A, which carries transit traffic between other Autonomous Systems. Autonomous System A will have a number of BGP Border Routers, and a mesh of BGP connections among them, over which BGP routes are distributed. In many such cases, it is desirable to avoid distributing the BGP routes to routers which are not BGP Border Routers. If this can be avoided, the "route distribution load" on those routers is significantly reduced. However, there must be some means of ensuring that the transit traffic will be delivered from Border Router to Border Router by the interior routers.

This can easily be done by means of LSP Tunnels. Suppose that BGP routes are distributed only to BGP Border Routers, and not to the interior routers that lie along the Hop-by-hop path from Border Router to Border Router. LSP Tunnels can then be used as follows:

1. Each BGP Border Router distributes, to every other BGP Border Router in the same Autonomous System, a label for each address prefix that it distributes to that router via BGP.

2. The IGP for the Autonomous System maintains a host route for each BGP Border Router. Each interior router distributes its labels for these host routes to each of its IGP neighbors.

3. Suppose that:

   a) BGP Border Router B1 receives an unlabeled packet P,

   b) address prefix X in B1’s routing table is the longest match for the destination address of P,

   c) the route to X is a BGP route,

   d) the BGP Next Hop for X is B2,

e) B2 has bound label L1 to X, and has distributed this binding to B1,

f) the IGP next hop for the address of B2 is I1,

g) the address of B2 is in B1's and I1's IGP routing tables as a host route, and

h) I1 has bound label L2 to the address of B2, and distributed this binding to B1.

Then before sending packet P to I1, B1 must create a label stack for P, then push on label L1, and then push on label L2.

4. Suppose that BGP Border Router B1 receives a labeled Packet P, where the label on the top of the label stack corresponds to an address prefix, X, to which the route is a BGP route, and that conditions 3b, 3c, 3d, and 3e all hold. Then before sending packet P to I1, B1 must replace the label at the top of the label stack with L1, and then push on label L2.

With these procedures, a given packet P follows a level 1 LSP all of whose members are BGP Border Routers, and between each pair of BGP Border Routers in the level 1 LSP, it follows a level 2 LSP.

These procedures effectively create a Hop-by-Hop Routed LSP Tunnel between the BGP Border Routers.

Since the BGP border routers are exchanging label bindings for address prefixes that are not even known to the IGP routing, the BGP routers should become explicit label distribution peers with each other.

It is sometimes possible to create Hop-by-Hop Routed LSP Tunnels between two BGP Border Routers, even if they are not in the same Autonomous System. Suppose, for example, that B1 and B2 are in AS 1. Suppose that B3 is an EBGP neighbor of B2, and is in AS2. Finally, suppose that B2 and B3 are on some network which is common to both Autonomous Systems (a "Demilitarized Zone"). In this case, an LSP tunnel can be set up directly between B1 and B3 as follows:

- B3 distributes routes to B2 (using EBGP), optionally assigning labels to address prefixes;

- B2 redistributes those routes to B1 (using IBGP), indicating that the BGP next hop for each such route is B3. If B3 has assigned labels to address prefixes, B2 passes these labels along, unchanged, to B1.

- The IGP of AS1 has a host route for B3.

4.7. Other Uses of Hop-by-Hop Routed LSP Tunnels

The use of Hop-by-Hop Routed LSP Tunnels is not restricted to tunnels between BGP Next Hops. Any situation in which one might otherwise have used an encapsulation tunnel is one in which it is appropriate to use a Hop-by-Hop Routed LSP Tunnel. Instead of encapsulating the packet with a new header whose destination address is the address of the tunnel's receive endpoint, the label corresponding to the address prefix which is the longest match for the address of the tunnel's receive endpoint is pushed on the packet's label stack. The packet which is sent into the tunnel may or may not already be labeled.

If the transmit endpoint of the tunnel wishes to put a labeled packet into the tunnel, it must first replace the label value at the top of the stack with a label value that was distributed to it by the tunnel's receive endpoint. Then it must push on the label which corresponds to the tunnel itself, as distributed to it by the next hop along the tunnel. To allow this, the tunnel endpoints should be explicit label distribution peers. The label bindings they need to exchange are of no interest to the LSRs along the tunnel.

4.8. MPLS and Multicast

Multicast routing proceeds by constructing multicast trees. The tree along which a particular multicast packet must get forwarded depends in general on the packet's source address and its destination address. Whenever a particular LSR is a node in a particular multicast tree, it binds a label to that tree. It then distributes that binding to its parent on the multicast tree. (If the node in question is on a LAN, and has siblings on that LAN, it must also distribute the binding to its siblings. This allows the parent to use a single label value when multicasting to all children on the LAN.)

When a multicast labeled packet arrives, the NHLFE corresponding to the label indicates the set of output interfaces for that packet, as well as the outgoing label. If the same label encoding technique is used on all the outgoing interfaces, the very same packet can be sent to all the children.

5. Label Distribution Procedures (Hop-by-Hop)

In this section, we consider only label bindings that are used for traffic to be label switched along its hop-by-hop routed path. In these cases, the label in question will correspond to an address prefix in the routing table.

5.1. The Procedures for Advertising and Using labels

There are a number of different procedures that may be used to distribute label bindings. Some are executed by the downstream LSR, and some by the upstream LSR.

The downstream LSR must perform:

- The Distribution Procedure, and
- the Withdrawal Procedure.

The upstream LSR must perform:

- The Request Procedure, and
- the NotAvailable Procedure, and
- the Release Procedure, and
- the labelUse Procedure.

The MPLS architecture supports several variants of each procedure.

However, the MPLS architecture does not support all possible combinations of all possible variants. The set of supported combinations will be described in section 5.2, where the interoperability between different combinations will also be discussed.

5.1.1. Downstream LSR: Distribution Procedure

The Distribution Procedure is used by a downstream LSR to determine when it should distribute a label binding for a particular address prefix to its label distribution peers. The architecture supports four different distribution procedures.

Irrespective of the particular procedure that is used, if a label binding for a particular address prefix has been distributed by a downstream LSR Rd to an upstream LSR Ru, and if at any time the attributes (as defined above) of that binding change, then Rd must inform Ru of the new attributes.

If an LSR is maintaining multiple routes to a particular address prefix, it is a local matter as to whether that LSR binds multiple labels to the address prefix (one per route), and hence distributes multiple bindings.

5.1.1.1. PushUnconditional

Let Rd be an LSR. Suppose that:

1. X is an address prefix in Rd's routing table

2. Ru is a label distribution peer of Rd with respect to X

Whenever these conditions hold, Rd must bind a label to X and distribute that binding to Ru. It is the responsibility of Rd to keep track of the bindings which it has distributed to Ru, and to make sure that Ru always has these bindings.

This procedure would be used by LSRs which are performing unsolicited downstream label assignment in the Independent LSP Control Mode.

5.1.1.2. PushConditional

Let Rd be an LSR. Suppose that:

1. X is an address prefix in Rd's routing table

2. Ru is a label distribution peer of Rd with respect to X

3. Rd is either an LSP Egress or an LSP Proxy Egress for X, or Rd's L3 next hop for X is Rn, where Rn is distinct from Ru, and Rn has bound a label to X and distributed that binding to Rd.

Then as soon as these conditions all hold, Rd should bind a label to X and distribute that binding to Ru.

Whereas PushUnconditional causes the distribution of label bindings for all address prefixes in the routing table, PushConditional causes the distribution of label bindings only for those address prefixes for which one has received label bindings from one's LSP next hop, or for which one does not have an MPLS-capable L3 next hop.

This procedure would be used by LSRs which are performing unsolicited downstream label assignment in the Ordered LSP Control Mode.

5.1.1.3. PulledUnconditional

Let Rd be an LSR. Suppose that:

1. X is an address prefix in Rd's routing table

2. Ru is a label distribution peer of Rd with respect to X

3. Ru has explicitly requested that Rd bind a label to X and distribute the binding to Ru

Then Rd should bind a label to X and distribute that binding to Ru. Note that if X is not in Rd's routing table, or if Rd is not a label distribution peer of Ru with respect to X, then Rd must inform Ru that it cannot provide a binding at this time.

If Rd has already distributed a binding for address prefix X to Ru, and it receives a new request from Ru for a binding for address prefix X, it will bind a second label, and distribute the new binding to Ru. The first label binding remains in effect.

This procedure would be used by LSRs performing downstream-on-demand label distribution using the Independent LSP Control Mode.

5.1.1.4. PulledConditional

Let Rd be an LSR. Suppose that:

1. X is an address prefix in Rd's routing table
2. Ru is a label distribution peer of Rd with respect to X
3. Ru has explicitly requested that Rd bind a label to X and distribute the binding to Ru
4. Rd is either an LSP Egress or an LSP Proxy Egress for X, or Rd's L3 next hop for X is Rn, where Rn is distinct from Ru, and Rn has bound a label to X and distributed that binding to Rd

Then as soon as these conditions all hold, Rd should bind a label to X and distribute that binding to Ru. Note that if X is not in Rd's routing table and a binding for X is not obtainable via Rd's next hop for X, or if Rd is not a label distribution peer of Ru with respect to X, then Rd must inform Ru that it cannot provide a binding at this time.

However, if the only condition that fails to hold is that Rn has not yet provided a label to Rd, then Rd must defer any response to Ru until such time as it has receiving a binding from Rn.

If Rd has distributed a label binding for address prefix X to Ru, and at some later time, any attribute of the label binding changes, then Rd must redistribute the label binding to Ru, with the new attribute. It must do this even though Ru does not issue a new Request.

This procedure would be used by LSRs that are performing downstream-on-demand label allocation in the Ordered LSP Control Mode.

In section 5.2, we will discuss how to choose the particular procedure to be used at any given time, and how to ensure interoperability among LSRs that choose different procedures.

5.1.2. Upstream LSR: Request Procedure

The Request Procedure is used by the upstream LSR for an address prefix to determine when to explicitly request that the downstream LSR bind a label to that prefix and distribute the binding. There are three possible procedures that can be used.

5.1.2.1. RequestNever

Never make a request. This is useful if the downstream LSR uses the PushConditional procedure or the PushUnconditional procedure, but is not useful if the downstream LSR uses the PulledUnconditional procedure or the the PulledConditional procedures.

This procedure would be used by an LSR when unsolicited downstream label distribution and Liberal Label Retention Mode are being used.

5.1.2.2. RequestWhenNeeded

Make a request whenever the L3 next hop to the address prefix changes, or when a new address prefix is learned, and one doesn’t already have a label binding from that next hop for the given address prefix.

This procedure would be used by an LSR whenever Conservative Label Retention Mode is being used.

5.1.2.3. RequestOnRequest

Issue a request whenever a request is received, in addition to issuing a request when needed (as described in section 5.1.2.2). If Ru is not capable of being an LSP ingress, it may issue a request only when it receives a request from upstream.

If Rd receives such a request from Ru, for an address prefix for which Rd has already distributed Ru a label, Rd shall assign a new (distinct) label, bind it to X, and distribute that binding. (Whether Rd can distribute this binding to Ru immediately or not depends on the Distribution Procedure being used.)

This procedure would be used by an LSR which is doing downstream-on-demand label distribution, but is not doing label merging, e.g., an ATM-LSR which is not capable of VC merge.

5.1.3. Upstream LSR: NotAvailable Procedure

If Ru and Rd are respectively upstream and downstream label distribution peers for address prefix X, and Rd is Ru's L3 next hop for X, and Ru requests a binding for X from Rd, but Rd replies that it cannot provide a binding at this time, because it has no next hop for X, then the NotAvailable procedure determines how Ru responds. There are two possible procedures governing Ru's behavior:

5.1.3.1. RequestRetry

Ru should issue the request again at a later time. That is, the requester is responsible for trying again later to obtain the needed binding. This procedure would be used when downstream-on-demand label distribution is used.

5.1.3.2. RequestNoRetry

Ru should never reissue the request, instead assuming that Rd will provide the binding automatically when it is available. This is useful if Rd uses the PushUnconditional procedure or the PushConditional procedure, i.e., if unsolicited downstream label distribution is used.

Note that if Rd replies that it cannot provide a binding to Ru, because of some error condition, rather than because Rd has no next hop, the behavior of Ru will be governed by the error recovery conditions of the label distribution protocol, rather than by the NotAvailable procedure.

5.1.4. Upstream LSR: Release Procedure

Suppose that Rd is an LSR which has bound a label to address prefix X, and has distributed that binding to LSR Ru. If Rd does not happen to be Ru's L3 next hop for address prefix X, or has ceased to be Ru's L3 next hop for address prefix X, then Ru will not be using the label. The Release Procedure determines how Ru acts in this case. There are two possible procedures governing Ru's behavior:

5.1.4.1. ReleaseOnChange

Ru should release the binding, and inform Rd that it has done so. This procedure would be used to implement Conservative Label Retention Mode.

5.1.4.2. NoReleaseOnChange

Ru should maintain the binding, so that it can use it again immediately if Rd later becomes Ru's L3 next hop for X. This procedure would be used to implement Liberal Label Retention Mode.

5.1.5. Upstream LSR: labelUse Procedure

Suppose Ru is an LSR which has received label binding L for address prefix X from LSR Rd, and Ru is upstream of Rd with respect to X, and in fact Rd is Ru's L3 next hop for X.

Ru will make use of the binding if Rd is Ru's L3 next hop for X. If, at the time the binding is received by Ru, Rd is NOT Ru's L3 next hop for X, Ru does not make any use of the binding at that time. Ru may however start using the binding at some later time, if Rd becomes Ru's L3 next hop for X.

The labelUse Procedure determines just how Ru makes use of Rd's binding.

There are two procedures which Ru may use:

5.1.5.1. UseImmediate

Ru may put the binding into use immediately. At any time when Ru has a binding for X from Rd, and Rd is Ru's L3 next hop for X, Rd will also be Ru's LSP next hop for X. This procedure is used when loop detection is not in use.

5.1.5.2. UseIfLoopNotDetected

This procedure is the same as UseImmediate, unless Ru has detected a loop in the LSP. If a loop has been detected, Ru will discontinue the use of label L for forwarding packets to Rd.

This procedure is used when loop detection is in use.

This will continue until the next hop for X changes, or until the loop is no longer detected.

5.1.6. Downstream LSR: Withdraw Procedure

In this case, there is only a single procedure.

When LSR Rd decides to break the binding between label L and address prefix X, then this unbinding must be distributed to all LSRs to which the binding was distributed.

It is required that the unbinding of L from X be distributed by Rd to a LSR Ru before Rd distributes to Ru any new binding of L to any other address prefix Y, where X != Y. If Ru were to learn of the new binding of L to Y before it learned of the unbinding of L from X, and if packets matching both X and Y were forwarded by Ru to Rd, then for a period of time, Ru would label both packets matching X and packets matching Y with label L.

The distribution and withdrawal of label bindings is done via a label distribution protocol. All label distribution protocols require that a label distribution adjacency be established between two label distribution peers (except implicit peers). If LSR R1 has a label distribution adjacency to LSR R2, and has received label bindings from LSR R2 via that adjacency, then if adjacency is brought down by either peer (whether as a result of failure or as a matter of normal operation), all bindings received over that adjacency must be considered to have been withdrawn.

As long as the relevant label distribution adjacency remains in place, label bindings that are withdrawn must always be withdrawn explicitly. If a second label is bound to an address prefix, the result is not to implicitly withdraw the first label, but to bind both labels; this is needed to support multi-path routing. If a second address prefix is bound to a label, the result is not to implicitly withdraw the binding of that label to the first address prefix, but to use that label for both address prefixes.

5.2. MPLS Schemes: Supported Combinations of Procedures

Consider two LSRs, Ru and Rd, which are label distribution peers with respect to some set of address prefixes, where Ru is the upstream peer and Rd is the downstream peer.

The MPLS scheme which governs the interaction of Ru and Rd can be described as a quintuple of procedures: <Distribution Procedure, Request Procedure, NotAvailable Procedure, Release Procedure, labelUse Procedure>. (Since there is only one Withdraw Procedure, it need not be mentioned.) A "*" appearing in one of the positions is a wild-card, meaning that any procedure in that category may be present; an "N/A" appearing in a particular position indicates that no procedure in that category is needed.

Only the MPLS schemes which are specified below are supported by the MPLS Architecture. Other schemes may be added in the future, if a need for them is shown.

5.2.1. Schemes for LSRs that Support Label Merging

If Ru and Rd are label distribution peers, and both support label merging, one of the following schemes must be used:

1. <PushUnconditional, RequestNever, N/A, NoReleaseOnChange, UseImmediate>

   This is unsolicited downstream label distribution with independent control, liberal label retention mode, and no loop detection.

2. <PushUnconditional, RequestNever, N/A, NoReleaseOnChange, UseIfLoopNotDetected>

   This is unsolicited downstream label distribution with independent control, liberal label retention, and loop detection.

3. <PushConditional, RequestWhenNeeded, RequestNoRetry, ReleaseOnChange, *>

   This is unsolicited downstream label distribution with ordered control (from the egress) and conservative label retention mode. Loop detection is optional.

4. <PushConditional, RequestNever, N/A, NoReleaseOnChange, *>

   This is unsolicited downstream label distribution with ordered control (from the egress) and liberal label retention mode. Loop detection is optional.

5. <PulledConditional, RequestWhenNeeded, RequestRetry, ReleaseOnChange, *>

   This is downstream-on-demand label distribution with ordered control (initiated by the ingress), conservative label retention mode, and optional loop detection.

6. <PulledUnconditional, RequestWhenNeeded, N/A, ReleaseOnChange, UseImmediate>

   This is downstream-on-demand label distribution with independent control and conservative label retention mode, without loop detection.

7. <PulledUnconditional, RequestWhenNeeded, N/A, ReleaseOnChange, UseIfLoopNotDetected>

   This is downstream-on-demand label distribution with independent control and conservative label retention mode, with loop detection.

5.2.2. Schemes for LSRs that do not Support Label Merging

Suppose that R1, R2, R3, and R4 are ATM switches which do not support label merging, but are being used as LSRs. Suppose further that the L3 hop-by-hop path for address prefix X is <R1, R2, R3, R4>, and that packets destined for X can enter the network at any of these LSRs. Since there is no multipoint-to-point capability, the LSPs must be realized as point-to-point VCs, which means that there needs to be three such VCs for address prefix X: <R1, R2, R3, R4>, <R2, R3, R4>, and <R3, R4>.

Therefore, if R1 and R2 are MPLS peers, and either is an LSR which is implemented using conventional ATM switching hardware (i.e., no cell interleave suppression), or is otherwise incapable of performing label merging, the MPLS scheme in use between R1 and R2 must be one of the following:

1. <PulledConditional, RequestOnRequest, RequestRetry, ReleaseOnChange, *>

   This is downstream-on-demand label distribution with ordered control (initiated by the ingress), conservative label retention mode, and optional loop detection.

   The use of the RequestOnRequest procedure will cause R4 to distribute three labels for X to R3; R3 will distribute 2 labels for X to R2, and R2 will distribute one label for X to R1.

2. <PulledUnconditional, RequestOnRequest, N/A, ReleaseOnChange, UseImmediate>

   This is downstream-on-demand label distribution with independent control and conservative label retention mode, without loop detection.

3. <PulledUnconditional, RequestOnRequest, N/A, ReleaseOnChange, UseIfLoopNotDetected>

   This is downstream-on-demand label distribution with independent control and conservative label retention mode, with loop detection.

5.2.3. Interoperability Considerations

It is easy to see that certain quintuples do NOT yield viable MPLS schemes. For example:

- <PulledUnconditional, RequestNever, *, *, *>
  <PulledConditional, RequestNever, *, *, *>

  In these MPLS schemes, the downstream LSR Rd distributes label bindings to upstream LSR Ru only upon request from Ru, but Ru never makes any such requests. Obviously, these schemes are not viable, since they will not result in the proper distribution of label bindings.

  - <*, RequestNever, *, *, ReleaseOnChange>

    In these MPLS schemes, Rd releases bindings when it isn't using them, but it never asks for them again, even if it later has a need for them. These schemes thus do not ensure that label bindings get properly distributed.

In this section, we specify rules to prevent a pair of label distribution peers from adopting procedures which lead to infeasible MPLS Schemes. These rules require either the exchange of information between label distribution peers during the initialization of the label distribution adjacency, or a priori knowledge of the information (obtained through a means outside the scope of this document).

1. Each must state whether it supports label merging.

2. If Rd does not support label merging, Rd must choose either the PulledUnconditional procedure or the PulledConditional procedure. If Rd chooses PulledConditional, Ru is forced to use the RequestRetry procedure.

   That is, if the downstream LSR does not support label merging, its preferences take priority when the MPLS scheme is chosen.

3. If Ru does not support label merging, but Rd does, Ru must choose either the RequestRetry or RequestNoRetry procedure. This forces Rd to use the PulledConditional or PulledUnConditional procedure respectively.

   That is, if only one of the LSRs doesn't support label merging, its preferences take priority when the MPLS scheme is chosen.

4. If both Ru and Rd both support label merging, then the choice between liberal and conservative label retention mode belongs to Ru. That is, Ru gets to choose either to use RequestWhenNeeded/ReleaseOnChange (conservative) , or to use RequestNever/NoReleaseOnChange (liberal). However, the choice of "push" vs. "pull" and "conditional" vs. "unconditional" belongs to Rd. If Ru chooses liberal label retention mode, Rd can choose either PushUnconditional or PushConditional. If Ru chooses conservative label retention mode, Rd can choose PushConditional, PulledConditional, or PulledUnconditional.

   These choices together determine the MPLS scheme in use.

6. Security Considerations

Some routers may implement security procedures which depend on the network layer header being in a fixed place relative to the data link layer header. The MPLS generic encapsulation inserts a shim between the data link layer header and the network layer header. This may cause any such security procedures to fail.

An MPLS label has its meaning by virtue of an agreement between the LSR that puts the label in the label stack (the "label writer"), and the LSR that interprets that label (the "label reader"). If labeled packets are accepted from untrusted sources, or if a particular incoming label is accepted from an LSR to which that label has not been distributed, then packets may be routed in an illegitimate manner.

7. Intellectual Property

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

8. Authors' Addresses

Eric C. Rosen
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA, 01824

EMail: erosen@cisco.com

Arun Viswanathan
Force10 Networks, Inc.
1440 McCarthy Blvd.
Milpitas, CA 95035-7438

EMail: arun@force10networks.com

Ross Callon
Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA

EMail: rcallon@juniper.net

9. References

[MPLS-ATM] Davie, B., Lawrence, J., McCloghrie, K., Rekhter, Y., Rosen, E., Swallow, G. and P. Doolan, "MPLS using LDP and ATM VC Switching", RFC 3035, January 2001.

[MPLS-BGP] "Carrying Label Information in BGP-4", Rekhter, Rosen, Work in Progress.

[MPLS-CR-LDP] "Constraint-Based LSP Setup using LDP", Jamoussi, Editor, Work in Progress.

[MPLS-FRMRLY] Conta, A., Doolan, P. and A. Malis, "Use of Label Switching on Frame Relay Networks Specification", RFC 3034, January 2001.

[MPLS-LDP] Andersson, L., Doolan, P., Feldman, N., Fredette, A. and B. Thomas, "LDP Specification", RFC 3036, January 2001.

[MPLS-RSVP-TUNNELS] "Extensions to RSVP for LSP Tunnels", Awduche, Berger, Gan, Li, Swallow, Srinvasan, Work in Progress.

[MPLS-SHIM] Rosen, E., Rekhter, Y., Tappan, D., Fedorkow, G., Farinacci, D. and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.

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

10. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

Exhibit 2

Network Working Group
Request for Comments: 3272
Category: Informational

D. Awduche
Movaz Networks
A. Chiu
Celion Networks
A. Elwalid
I. Widjaja
Lucent Technologies
X. Xiao
Redback Networks

May 2002

Overview and Principles of Internet Traffic Engineering

Status of this Memo

Copyright Notice

Abstract

This memo describes the principles of Traffic Engineering (TE) in the Internet. The document is intended to promote better understanding of the issues surrounding traffic engineering in IP networks, and to provide a common basis for the development of traffic engineering capabilities for the Internet. The principles, architectures, and methodologies for performance evaluation and performance optimization of operational IP networks are discussed throughout this document.

Table of Contents

1.0 Introduction

This memo describes the principles of Internet traffic engineering. The objective of the document is to articulate the general issues and principles for Internet traffic engineering; and where appropriate to provide recommendations, guidelines, and options for the development of online and offline Internet traffic engineering capabilities and support systems.

This document can aid service providers in devising and implementing traffic engineering solutions for their networks. Networking hardware and software vendors will also find this document helpful in the development of mechanisms and support systems for the Internet environment that support the traffic engineering function.

This document provides a terminology for describing and understanding common Internet traffic engineering concepts. This document also provides a taxonomy of known traffic engineering styles. In this context, a traffic engineering style abstracts important aspects from a traffic engineering methodology. Traffic engineering styles can be viewed in different ways depending upon the specific context in which they are used and the specific purpose which they serve. The combination of styles and views results in a natural taxonomy of traffic engineering systems.

Even though Internet traffic engineering is most effective when applied end-to-end, the initial focus of this document document is intra-domain traffic engineering (that is, traffic engineering within a given autonomous system). However, because a preponderance of Internet traffic tends to be inter-domain (originating in one autonomous system and terminating in another), this document provides an overview of aspects pertaining to inter-domain traffic engineering.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

1.1. What is Internet Traffic Engineering?

Internet traffic engineering is defined as that aspect of Internet network engineering dealing with the issue of performance evaluation and performance optimization of operational IP networks. Traffic Engineering encompasses the application of technology and scientific principles to the measurement, characterization, modeling, and control of Internet traffic [RFC-2702, AWD2].

Enhancing the performance of an operational network, at both the traffic and resource levels, are major objectives of Internet traffic engineering. This is accomplished by addressing traffic oriented performance requirements, while utilizing network resources economically and reliably. Traffic oriented performance measures include delay, delay variation, packet loss, and throughput.

An important objective of Internet traffic engineering is to facilitate reliable network operations [RFC-2702]. Reliable network operations can be facilitated by providing mechanisms that enhance network integrity and by embracing policies emphasizing network survivability. This results in a minimization of the vulnerability of the network to service outages arising from errors, faults, and failures occurring within the infrastructure.

The Internet exists in order to transfer information from source nodes to destination nodes. Accordingly, one of the most significant functions performed by the Internet is the routing of traffic from ingress nodes to egress nodes. Therefore, one of the most distinctive functions performed by Internet traffic engineering is the control and optimization of the routing function, to steer traffic through the network in the most effective way.

Ultimately, it is the performance of the network as seen by end users of network services that is truly paramount. This crucial point should be considered throughout the development of traffic engineering mechanisms and policies. The characteristics visible to end users are the emergent properties of the network, which are the characteristics of the network when viewed as a whole. A central goal of the service provider, therefore, is to enhance the emergent properties of the network while taking economic considerations into account.

The importance of the above observation regarding the emergent properties of networks is that special care must be taken when choosing network performance measures to optimize. Optimizing the wrong measures may achieve certain local objectives, but may have

disastrous consequences on the emergent properties of the network and thereby on the quality of service perceived by end-users of network services.

A subtle, but practical advantage of the systematic application of traffic engineering concepts to operational networks is that it helps to identify and structure goals and priorities in terms of enhancing the quality of service delivered to end-users of network services. The application of traffic engineering concepts also aids in the measurement and analysis of the achievement of these goals.

The optimization aspects of traffic engineering can be achieved through capacity management and traffic management. As used in this document, capacity management includes capacity planning, routing control, and resource management. Network resources of particular interest include link bandwidth, buffer space, and computational resources. Likewise, as used in this document, traffic management includes (1) nodal traffic control functions such as traffic conditioning, queue management, scheduling, and (2) other functions that regulate traffic flow through the network or that arbitrate access to network resources between different packets or between different traffic streams.

The optimization objectives of Internet traffic engineering should be viewed as a continual and iterative process of network performance improvement and not simply as a one time goal. Traffic engineering also demands continual development of new technologies and new methodologies for network performance enhancement.

The optimization objectives of Internet traffic engineering may change over time as new requirements are imposed, as new technologies emerge, or as new insights are brought to bear on the underlying problems. Moreover, different networks may have different optimization objectives, depending upon their business models, capabilities, and operating constraints. The optimization aspects of traffic engineering are ultimately concerned with network control regardless of the specific optimization goals in any particular environment.

Thus, the optimization aspects of traffic engineering can be viewed from a control perspective. The aspect of control within the Internet traffic engineering arena can be pro-active and/or reactive. In the pro-active case, the traffic engineering control system takes preventive action to obviate predicted unfavorable future network states. It may also take perfective action to induce a more desirable state in the future. In the reactive case, the control system responds correctively and perhaps adaptively to events that have already transpired in the network.

The control dimension of Internet traffic engineering responds at multiple levels of temporal resolution to network events. Certain aspects of capacity management, such as capacity planning, respond at very coarse temporal levels, ranging from days to possibly years. The introduction of automatically switched optical transport networks (e.g., based on the Multi-protocol Lambda Switching concepts) could significantly reduce the lifecycle for capacity planning by expediting provisioning of optical bandwidth. Routing control functions operate at intermediate levels of temporal resolution, ranging from milliseconds to days. Finally, the packet level processing functions (e.g., rate shaping, queue management, and scheduling) operate at very fine levels of temporal resolution, ranging from picoseconds to milliseconds while responding to the real-time statistical behavior of traffic. The subsystems of Internet traffic engineering control include: capacity augmentation, routing control, traffic control, and resource control (including control of service policies at network elements). When capacity is to be augmented for tactical purposes, it may be desirable to devise a deployment plan that expedites bandwidth provisioning while minimizing installation costs.

Inputs into the traffic engineering control system include network state variables, policy variables, and decision variables.

One major challenge of Internet traffic engineering is the realization of automated control capabilities that adapt quickly and cost effectively to significant changes in a network's state, while still maintaining stability.

Another critical dimension of Internet traffic engineering is network performance evaluation, which is important for assessing the effectiveness of traffic engineering methods, and for monitoring and verifying compliance with network performance goals. Results from performance evaluation can be used to identify existing problems, guide network re-optimization, and aid in the prediction of potential future problems.

Performance evaluation can be achieved in many different ways. The most notable techniques include analytical methods, simulation, and empirical methods based on measurements. When analytical methods or simulation are used, network nodes and links can be modeled to capture relevant operational features such as topology, bandwidth, buffer space, and nodal service policies (link scheduling, packet prioritization, buffer management, etc.). Analytical traffic models can be used to depict dynamic and behavioral traffic characteristics, such as burstiness, statistical distributions, and dependence.

Performance evaluation can be quite complicated in practical network contexts. A number of techniques can be used to simplify the analysis, such as abstraction, decomposition, and approximation. For example, simplifying concepts such as effective bandwidth and effective buffer [Elwalid] may be used to approximate nodal behaviors at the packet level and simplify the analysis at the connection level. Network analysis techniques using, for example, queuing models and approximation schemes based on asymptotic and decomposition techniques can render the analysis even more tractable. In particular, an emerging set of concepts known as network calculus [CRUZ] based on deterministic bounds may simplify network analysis relative to classical stochastic techniques. When using analytical techniques, care should be taken to ensure that the models faithfully reflect the relevant operational characteristics of the modeled network entities.

Simulation can be used to evaluate network performance or to verify and validate analytical approximations. Simulation can, however, be computationally costly and may not always provide sufficient insights. An appropriate approach to a given network performance evaluation problem may involve a hybrid combination of analytical techniques, simulation, and empirical methods.

As a general rule, traffic engineering concepts and mechanisms must be sufficiently specific and well defined to address known requirements, but simultaneously flexible and extensible to accommodate unforeseen future demands.

1.2. Scope

The scope of this document is intra-domain traffic engineering; that is, traffic engineering within a given autonomous system in the Internet. This document will discuss concepts pertaining to intra-domain traffic control, including such issues as routing control, micro and macro resource allocation, and the control coordination problems that arise consequently.

This document will describe and characterize techniques already in use or in advanced development for Internet traffic engineering. The way these techniques fit together will be discussed and scenarios in which they are useful will be identified.

While this document considers various intra-domain traffic engineering approaches, it focuses more on traffic engineering with MPLS. Traffic engineering based upon manipulation of IGP metrics is not addressed in detail. This topic may be addressed by other working group document(s).

Although the emphasis is on intra-domain traffic engineering, in Section 7.0, an overview of the high level considerations pertaining to inter-domain traffic engineering will be provided. Inter-domain Internet traffic engineering is crucial to the performance enhancement of the global Internet infrastructure.

Whenever possible, relevant requirements from existing IETF documents and other sources will be incorporated by reference.

1.3 Terminology

This subsection provides terminology which is useful for Internet traffic engineering. The definitions presented apply to this document. These terms may have other meanings elsewhere.

- Baseline analysis:
  A study conducted to serve as a baseline for comparison to the actual behavior of the network.

- Busy hour:
  A one hour period within a specified interval of time (typically 24 hours) in which the traffic load in a network or sub-network is greatest.

- Bottleneck:
  A network element whose input traffic rate tends to be greater than its output rate.

- Congestion:
  A state of a network resource in which the traffic incident on the resource exceeds its output capacity over an interval of time.

- Congestion avoidance:
  An approach to congestion management that attempts to obviate the occurrence of congestion.

- Congestion control:
  An approach to congestion management that attempts to remedy congestion problems that have already occurred.

- Constraint-based routing:
  A class of routing protocols that take specified traffic attributes, network constraints, and policy constraints into account when making routing decisions. Constraint-based routing is applicable to traffic aggregates as well as flows. It is a generalization of QoS routing.

- Demand side congestion management:
  A congestion management scheme that addresses congestion problems by regulating or conditioning offered load.

- Effective bandwidth:
  The minimum amount of bandwidth that can be assigned to a flow or traffic aggregate in order to deliver 'acceptable service quality' to the flow or traffic aggregate.

- Egress traffic:
  Traffic exiting a network or network element.

- Hot-spot:
  A network element or subsystem which is in a state of congestion.

- Ingress traffic:
  Traffic entering a network or network element.

- Inter-domain traffic:
  Traffic that originates in one Autonomous system and terminates in another.

- Loss network:
  A network that does not provide adequate buffering for traffic, so that traffic entering a busy resource within the network will be dropped rather than queued.

- Metric:
  A parameter defined in terms of standard units of measurement.

- Measurement Methodology:
  A repeatable measurement technique used to derive one or more metrics of interest.

- Network Survivability:
  The capability to provide a prescribed level of QoS for existing services after a given number of failures occur within the network.

- Offline traffic engineering:
  A traffic engineering system that exists outside of the network.

- Online traffic engineering:
  A traffic engineering system that exists within the network, typically implemented on or as adjuncts to operational network elements.

- Performance measures:
  Metrics that provide quantitative or qualitative measures of the performance of systems or subsystems of interest.

- Performance management:
  A systematic approach to improving effectiveness in the accomplishment of specific networking goals related to performance improvement.

- Performance Metric:
  A performance parameter defined in terms of standard units of measurement.

- Provisioning:
  The process of assigning or configuring network resources to meet certain requests.

- QoS routing:
  Class of routing systems that selects paths to be used by a flow based on the QoS requirements of the flow.

- Service Level Agreement:
  A contract between a provider and a customer that guarantees specific levels of performance and reliability at a certain cost.

- Stability:
  An operational state in which a network does not oscillate in a disruptive manner from one mode to another mode.

- Supply side congestion management:
  A congestion management scheme that provisions additional network resources to address existing and/or anticipated congestion problems.

- Transit traffic:
  Traffic whose origin and destination are both outside of the network under consideration.

- Traffic characteristic:
  A description of the temporal behavior or a description of the attributes of a given traffic flow or traffic aggregate.

- Traffic engineering system:
  A collection of objects, mechanisms, and protocols that are used conjunctively to accomplish traffic engineering objectives.

- Traffic flow:
  A stream of packets between two end-points that can be characterized in a certain way. A micro-flow has a more specific definition: A micro-flow is a stream of packets with the same source and destination addresses, source and destination ports, and protocol ID.

- Traffic intensity:
  A measure of traffic loading with respect to a resource capacity over a specified period of time. In classical telephony systems, traffic intensity is measured in units of Erlang.

- Traffic matrix:
  A representation of the traffic demand between a set of origin and destination abstract nodes. An abstract node can consist of one or more network elements.

- Traffic monitoring:
  The process of observing traffic characteristics at a given point in a network and collecting the traffic information for analysis and further action.

- Traffic trunk:
  An aggregation of traffic flows belonging to the same class which are forwarded through a common path. A traffic trunk may be characterized by an ingress and egress node, and a set of attributes which determine its behavioral characteristics and requirements from the network.

2.0 Background

The Internet has quickly evolved into a very critical communications infrastructure, supporting significant economic, educational, and social activities. Simultaneously, the delivery of Internet communications services has become very competitive and end-users are demanding very high quality service from their service providers. Consequently, performance optimization of large scale IP networks, especially public Internet backbones, have become an important problem. Network performance requirements are multi-dimensional, complex, and sometimes contradictory; making the traffic engineering problem very challenging.

The network must convey IP packets from ingress nodes to egress nodes efficiently, expeditiously, and economically. Furthermore, in a multiclass service environment (e.g., Diffserv capable networks), the resource sharing parameters of the network must be appropriately determined and configured according to prevailing policies and service models to resolve resource contention issues arising from mutual interference between packets traversing through the network. Thus, consideration must be given to resolving competition for network resources between traffic streams belonging to the same service class (intra-class contention resolution) and traffic streams belonging to different classes (inter-class contention resolution).

2.1 Context of Internet Traffic Engineering

The context of Internet traffic engineering pertains to the scenarios where traffic engineering is used. A traffic engineering methodology establishes appropriate rules to resolve traffic performance issues occurring in a specific context. The context of Internet traffic engineering includes:

(1) A network context defining the universe of discourse, and in particular the situations in which the traffic engineering problems occur. The network context includes network structure, network policies, network characteristics, network constraints, network quality attributes, and network optimization criteria.

(2) A problem context defining the general and concrete issues that traffic engineering addresses. The problem context includes identification, abstraction of relevant features, representation, formulation, specification of the requirements on the solution space, and specification of the desirable features of acceptable solutions.

(3) A solution context suggesting how to address the issues identified by the problem context. The solution context includes analysis, evaluation of alternatives, prescription, and resolution.

(4) An implementation and operational context in which the solutions are methodologically instantiated. The implementation and operational context includes planning, organization, and execution.

The context of Internet traffic engineering and the different problem scenarios are discussed in the following subsections.

2.2 Network Context

IP networks range in size from small clusters of routers situated within a given location, to thousands of interconnected routers, switches, and other components distributed all over the world.

Conceptually, at the most basic level of abstraction, an IP network can be represented as a distributed dynamical system consisting of: (1) a set of interconnected resources which provide transport services for IP traffic subject to certain constraints, (2) a demand system representing the offered load to be transported through the network, and (3) a response system consisting of network processes, protocols, and related mechanisms which facilitate the movement of traffic through the network [see also AWD2].

The network elements and resources may have specific characteristics restricting the manner in which the demand is handled. Additionally, network resources may be equipped with traffic control mechanisms superintending the way in which the demand is serviced. Traffic control mechanisms may, for example, be used to control various packet processing activities within a given resource, arbitrate contention for access to the resource by different packets, and regulate traffic behavior through the resource. A configuration management and provisioning system may allow the settings of the traffic control mechanisms to be manipulated by external or internal entities in order to exercise control over the way in which the network elements respond to internal and external stimuli.

The details of how the network provides transport services for packets are specified in the policies of the network administrators and are installed through network configuration management and policy based provisioning systems. Generally, the types of services provided by the network also depends upon the technology and characteristics of the network elements and protocols, the prevailing service and utility models, and the ability of the network administrators to translate policies into network configurations.

Contemporary Internet networks have three significant characteristics: (1) they provide real-time services, (2) they have become mission critical, and (3) their operating environments are very dynamic. The dynamic characteristics of IP networks can be attributed in part to fluctuations in demand, to the interaction between various network protocols and processes, to the rapid evolution of the infrastructure which demands the constant inclusion of new technologies and new network elements, and to transient and persistent impairments which occur within the system.

Packets contend for the use of network resources as they are conveyed through the network. A network resource is considered to be congested if the arrival rate of packets exceed the output capacity of the resource over an interval of time. Congestion may result in some of the arrival packets being delayed or even dropped.

Congestion increases transit delays, delay variation, packet loss, and reduces the predictability of network services. Clearly, congestion is a highly undesirable phenomenon.

Combating congestion at a reasonable cost is a major objective of Internet traffic engineering.

Efficient sharing of network resources by multiple traffic streams is a basic economic premise for packet switched networks in general and for the Internet in particular. A fundamental challenge in network operation, especially in a large scale public IP network, is to increase the efficiency of resource utilization while minimizing the possibility of congestion.

Increasingly, the Internet will have to function in the presence of different classes of traffic with different service requirements. The advent of Differentiated Services [RFC-2475] makes this requirement particularly acute. Thus, packets may be grouped into behavior aggregates such that each behavior aggregate may have a common set of behavioral characteristics or a common set of delivery requirements. In practice, the delivery requirements of a specific set of packets may be specified explicitly or implicitly. Two of the most important traffic delivery requirements are capacity constraints and QoS constraints.

Capacity constraints can be expressed statistically as peak rates, mean rates, burst sizes, or as some deterministic notion of effective bandwidth. QoS requirements can be expressed in terms of (1) integrity constraints such as packet loss and (2) in terms of temporal constraints such as timing restrictions for the delivery of each packet (delay) and timing restrictions for the delivery of consecutive packets belonging to the same traffic stream (delay variation).

2.3 Problem Context

Fundamental problems exist in association with the operation of a network described by the simple model of the previous subsection. This subsection reviews the problem context in relation to the traffic engineering function.

The identification, abstraction, representation, and measurement of network features relevant to traffic engineering is a significant issue.

One particularly important class of problems concerns how to explicitly formulate the problems that traffic engineering attempts to solve, how to identify the requirements on the solution space, how to specify the desirable features of good solutions, how to actually solve the problems, and how to measure and characterize the effectiveness of the solutions.

Another class of problems concerns how to measure and estimate relevant network state parameters. Effective traffic engineering relies on a good estimate of the offered traffic load as well as a view of the underlying topology and associated resource constraints. A network-wide view of the topology is also a must for offline planning.

Still another class of problems concerns how to characterize the state of the network and how to evaluate its performance under a variety of scenarios. The performance evaluation problem is two-fold. One aspect of this problem relates to the evaluation of the system level performance of the network. The other aspect relates to the evaluation of the resource level performance, which restricts attention to the performance analysis of individual network resources. In this memo, we refer to the system level characteristics of the network as the "macro-states" and the resource level characteristics as the "micro-states." The system level characteristics are also known as the emergent properties of the network as noted earlier. Correspondingly, we shall refer to the traffic engineering schemes dealing with network performance optimization at the systems level as "macro-TE" and the schemes that optimize at the individual resource level as "micro-TE." Under certain circumstances, the system level performance can be derived from the resource level performance using appropriate rules of composition, depending upon the particular performance measures of interest.

Another fundamental class of problems concerns how to effectively optimize network performance. Performance optimization may entail translating solutions to specific traffic engineering problems into network configurations. Optimization may also entail some degree of resource management control, routing control, and/or capacity augmentation.

As noted previously, congestion is an undesirable phenomena in operational networks. Therefore, the next subsection addresses the issue of congestion and its ramifications within the problem context of Internet traffic engineering.

2.3.1 Congestion and its Ramifications

Congestion is one of the most significant problems in an operational IP context. A network element is said to be congested if it experiences sustained overload over an interval of time. Congestion almost always results in degradation of service quality to end users. Congestion control schemes can include demand side policies and supply side policies. Demand side policies may restrict access to congested resources and/or dynamically regulate the demand to alleviate the overload situation. Supply side policies may expand or augment network capacity to better accommodate offered traffic. Supply side policies may also re-allocate network resources by redistributing traffic over the infrastructure. Traffic redistribution and resource re-allocation serve to increase the 'effective capacity' seen by the demand.

The emphasis of this memo is primarily on congestion management schemes falling within the scope of the network, rather than on congestion management systems dependent upon sensitivity and adaptivity from end-systems. That is, the aspects that are considered in this memo with respect to congestion management are those solutions that can be provided by control entities operating on the network and by the actions of network administrators and network operations systems.

2.4 Solution Context

The solution context for Internet traffic engineering involves analysis, evaluation of alternatives, and choice between alternative courses of action. Generally the solution context is predicated on making reasonable inferences about the current or future state of the network, and subsequently making appropriate decisions that may involve a preference between alternative sets of action. More specifically, the solution context demands reasonable estimates of traffic workload, characterization of network state, deriving solutions to traffic engineering problems which may be implicitly or explicitly formulated, and possibly instantiating a set of control actions. Control actions may involve the manipulation of parameters associated with routing, control over tactical capacity acquisition, and control over the traffic management functions.

The following list of instruments may be applicable to the solution context of Internet traffic engineering.

(1) A set of policies, objectives, and requirements (which may be context dependent) for network performance evaluation and performance optimization.

(2) A collection of online and possibly offline tools and mechanisms for measurement, characterization, modeling, and control of Internet traffic and control over the placement and allocation of network resources, as well as control over the mapping or distribution of traffic onto the infrastructure.

(3) A set of constraints on the operating environment, the network protocols, and the traffic engineering system itself.

(4) A set of quantitative and qualitative techniques and methodologies for abstracting, formulating, and solving traffic engineering problems.

(5) A set of administrative control parameters which may be manipulated through a Configuration Management (CM) system. The CM system itself may include a configuration control subsystem, a configuration repository, a configuration accounting subsystem, and a configuration auditing subsystem.

(6) A set of guidelines for network performance evaluation, performance optimization, and performance improvement.

Derivation of traffic characteristics through measurement and/or estimation is very useful within the realm of the solution space for traffic engineering. Traffic estimates can be derived from customer subscription information, traffic projections, traffic models, and from actual empirical measurements. The empirical measurements may be performed at the traffic aggregate level or at the flow level in order to derive traffic statistics at various levels of detail. Measurements at the flow level or on small traffic aggregates may be performed at edge nodes, where traffic enters and leaves the network. Measurements at large traffic aggregate levels may be performed within the core of the network where potentially numerous traffic flows may be in transit concurrently.

To conduct performance studies and to support planning of existing and future networks, a routing analysis may be performed to determine the path(s) the routing protocols will choose for various traffic demands, and to ascertain the utilization of network resources as traffic is routed through the network. The routing analysis should capture the selection of paths through the network, the assignment of

traffic across multiple feasible routes, and the multiplexing of IP traffic over traffic trunks (if such constructs exists) and over the underlying network infrastructure. A network topology model is a necessity for routing analysis. A network topology model may be extracted from network architecture documents, from network designs, from information contained in router configuration files, from routing databases, from routing tables, or from automated tools that discover and depict network topology information. Topology information may also be derived from servers that monitor network state, and from servers that perform provisioning functions.

Routing in operational IP networks can be administratively controlled at various levels of abstraction including the manipulation of BGP attributes and manipulation of IGP metrics. For path oriented technologies such as MPLS, routing can be further controlled by the manipulation of relevant traffic engineering parameters, resource parameters, and administrative policy constraints. Within the context of MPLS, the path of an explicit label switched path (LSP) can be computed and established in various ways including: (1) manually, (2) automatically online using constraint-based routing processes implemented on label switching routers, and (3) automatically offline using constraint-based routing entities implemented on external traffic engineering support systems.

2.4.1 Combating the Congestion Problem

Minimizing congestion is a significant aspect of Internet traffic engineering. This subsection gives an overview of the general approaches that have been used or proposed to combat congestion problems.

Congestion management policies can be categorized based upon the following criteria (see e.g., [YARE95] for a more detailed taxonomy of congestion control schemes): (1) Response time scale which can be characterized as long, medium, or short; (2) reactive versus preventive which relates to congestion control and congestion avoidance; and (3) supply side versus demand side congestion management schemes. These aspects are discussed in the following paragraphs.

(1) Congestion Management based on Response Time Scales

- Long (weeks to months): Capacity planning works over a relatively long time scale to expand network capacity based on estimates or forecasts of future traffic demand and traffic distribution. Since router and link provisioning take time and are generally expensive, these upgrades are typically carried out in the weeks-to-months or even years time scale.

- Medium (minutes to days): Several control policies fall within the medium time scale category. Examples include: (1) Adjusting IGP and/or BGP parameters to route traffic away or towards certain segments of the network; (2) Setting up and/or adjusting some explicitly routed label switched paths (ER-LSPs) in MPLS networks to route some traffic trunks away from possibly congested resources or towards possibly more favorable routes; (3) re-configuring the logical topology of the network to make it correlate more closely with the spatial traffic distribution using for example some underlying path-oriented technology such as MPLS LSPs, ATM PVCs, or optical channel trails. Many of these adaptive medium time scale response schemes rely on a measurement system that monitors changes in traffic distribution, traffic shifts, and network resource utilization and subsequently provides feedback to the online and/or offline traffic engineering mechanisms and tools which employ this feedback information to trigger certain control actions to occur within the network. The traffic engineering mechanisms and tools can be implemented in a distributed fashion or in a centralized fashion, and may have a hierarchical structure or a flat structure. The comparative merits of distributed and centralized control structures for networks are well known. A centralized scheme may have global visibility into the network state and may produce potentially more optimal solutions. However, centralized schemes are prone to single points of failure and may not scale as well as distributed schemes. Moreover, the information utilized by a centralized scheme may be stale and may not reflect the actual state of the network. It is not an objective of this memo to make a recommendation between distributed and centralized schemes. This is a choice that network administrators must make based on their specific needs.

- Short (picoseconds to minutes): This category includes packet level processing functions and events on the order of several round trip times. It includes router mechanisms such as passive and active buffer management. These mechanisms are used to control congestion and/or signal congestion to end systems so that they can adaptively regulate the rate at which traffic is injected into the network. One of the most popular active queue management schemes, especially for TCP traffic, is Random Early Detection (RED) [FLJA93], which supports congestion avoidance by controlling the average queue size. During congestion (but before the queue is filled), the RED scheme chooses arriving packets to "mark" according to a probabilistic algorithm which takes into account the average queue size. For a router that does not utilize explicit congestion notification (ECN) see e.g., [FLOY94], the marked packets can simply be dropped to signal the inception of congestion to end systems. On the other hand, if the router supports ECN, then it can set the ECN field in the packet header. Several variations of RED have been proposed to support different drop precedence levels in multi-class environments [RFC-

2597], e.g., RED with In and Out (RIO) and Weighted RED. There is general consensus that RED provides congestion avoidance performance which is not worse than traditional Tail-Drop (TD) queue management (drop arriving packets only when the queue is full). Importantly, however, RED reduces the possibility of global synchronization and improves fairness among different TCP sessions. However, RED by itself can not prevent congestion and unfairness caused by sources unresponsive to RED, e.g., UDP traffic and some misbehaved greedy connections. Other schemes have been proposed to improve the performance and fairness in the presence of unresponsive traffic. Some of these schemes were proposed as theoretical frameworks and are typically not available in existing commercial products. Two such schemes are Longest Queue Drop (LQD) and Dynamic Soft Partitioning with Random Drop (RND) [SLDC98].

(2) Congestion Management: Reactive versus Preventive Schemes

- Reactive: reactive (recovery) congestion management policies react to existing congestion problems to improve it. All the policies described in the long and medium time scales above can be categorized as being reactive especially if the policies are based on monitoring and identifying existing congestion problems, and on the initiation of relevant actions to ease a situation.

- Preventive: preventive (predictive/avoidance) policies take proactive action to prevent congestion based on estimates and predictions of future potential congestion problems. Some of the policies described in the long and medium time scales fall into this category. They do not necessarily respond immediately to existing congestion problems. Instead forecasts of traffic demand and workload distribution are considered and action may be taken to prevent potential congestion problems in the future. The schemes described in the short time scale (e.g., RED and its variations, ECN, LQD, and RND) are also used for congestion avoidance since dropping or marking packets before queues actually overflow would trigger corresponding TCP sources to slow down.

(3) Congestion Management: Supply Side versus Demand Side Schemes

- Supply side: supply side congestion management policies increase the effective capacity available to traffic in order to control or obviate congestion. This can be accomplished by augmenting capacity. Another way to accomplish this is to minimize congestion by having a relatively balanced distribution of traffic over the network. For example, capacity planning should aim to provide a physical topology and associated link bandwidths that match estimated traffic workload and traffic distribution based on forecasting (subject to budgetary and other constraints). However, if actual traffic distribution does

not match the topology derived from capacity panning (due to forecasting errors or facility constraints for example), then the traffic can be mapped onto the existing topology using routing control mechanisms, using path oriented technologies (e.g., MPLS LSPs and optical channel trails) to modify the logical topology, or by using some other load redistribution mechanisms.

- Demand side: demand side congestion management policies control or regulate the offered traffic to alleviate congestion problems. For example, some of the short time scale mechanisms described earlier (such as RED and its variations, ECN, LQD, and RND) as well as policing and rate shaping mechanisms attempt to regulate the offered load in various ways. Tariffs may also be applied as a demand side instrument. To date, however, tariffs have not been used as a means of demand side congestion management within the Internet.

In summary, a variety of mechanisms can be used to address congestion problems in IP networks. These mechanisms may operate at multiple time-scales.

2.5 Implementation and Operational Context

The operational context of Internet traffic engineering is characterized by constant change which occur at multiple levels of abstraction. The implementation context demands effective planning, organization, and execution. The planning aspects may involve determining prior sets of actions to achieve desired objectives. Organizing involves arranging and assigning responsibility to the various components of the traffic engineering system and coordinating the activities to accomplish the desired TE objectives. Execution involves measuring and applying corrective or perfective actions to attain and maintain desired TE goals.

3.0 Traffic Engineering Process Model(s)

This section describes a generic process model that captures the high level practical aspects of Internet traffic engineering in an operational context. The process model is described as a sequence of actions that a traffic engineer, or more generally a traffic engineering system, must perform to optimize the performance of an operational network (see also [RFC-2702, AWD2]). The process model described here represents the broad activities common to most traffic engineering methodologies although the details regarding how traffic engineering is executed may differ from network to network. This process model may be enacted explicitly or implicitly, by an automaton and/or by a human.

The traffic engineering process model is iterative [AWD2]. The four phases of the process model described below are repeated continually.

The first phase of the TE process model is to define the relevant control policies that govern the operation of the network. These policies may depend upon many factors including the prevailing business model, the network cost structure, the operating constraints, the utility model, and optimization criteria.

The second phase of the process model is a feedback mechanism involving the acquisition of measurement data from the operational network. If empirical data is not readily available from the network, then synthetic workloads may be used instead which reflect either the prevailing or the expected workload of the network. Synthetic workloads may be derived by estimation or extrapolation using prior empirical data. Their derivation may also be obtained using mathematical models of traffic characteristics or other means.

The third phase of the process model is to analyze the network state and to characterize traffic workload. Performance analysis may be proactive and/or reactive. Proactive performance analysis identifies potential problems that do not exist, but could manifest in the future. Reactive performance analysis identifies existing problems, determines their cause through diagnosis, and evaluates alternative approaches to remedy the problem, if necessary. A number of quantitative and qualitative techniques may be used in the analysis process, including modeling based analysis and simulation. The analysis phase of the process model may involve investigating the concentration and distribution of traffic across the network or relevant subsets of the network, identifying the characteristics of the offered traffic workload, identifying existing or potential bottlenecks, and identifying network pathologies such as ineffective link placement, single points of failures, etc. Network pathologies may result from many factors including inferior network architecture, inferior network design, and configuration problems. A traffic matrix may be constructed as part of the analysis process. Network analysis may also be descriptive or prescriptive.

The fourth phase of the TE process model is the performance optimization of the network. The performance optimization phase involves a decision process which selects and implements a set of actions from a set of alternatives. Optimization actions may include the use of appropriate techniques to either control the offered traffic or to control the distribution of traffic across the network. Optimization actions may also involve adding additional links or increasing link capacity, deploying additional hardware such as routers and switches, systematically adjusting parameters associated with routing such as IGP metrics and BGP attributes, and adjusting

traffic management parameters. Network performance optimization may also involve starting a network planning process to improve the network architecture, network design, network capacity, network technology, and the configuration of network elements to accommodate current and future growth.

3.1 Components of the Traffic Engineering Process Model

The key components of the traffic engineering process model include a measurement subsystem, a modeling and analysis subsystem, and an optimization subsystem. The following subsections examine these components as they apply to the traffic engineering process model.

3.2 Measurement

Measurement is crucial to the traffic engineering function. The operational state of a network can be conclusively determined only through measurement. Measurement is also critical to the optimization function because it provides feedback data which is used by traffic engineering control subsystems. This data is used to adaptively optimize network performance in response to events and stimuli originating within and outside the network. Measurement is also needed to determine the quality of network services and to evaluate the effectiveness of traffic engineering policies. Experience suggests that measurement is most effective when acquired and applied systematically.

When developing a measurement system to support the traffic engineering function in IP networks, the following questions should be carefully considered: Why is measurement needed in this particular context? What parameters are to be measured? How should the measurement be accomplished? Where should the measurement be performed? When should the measurement be performed? How frequently should the monitored variables be measured? What level of measurement accuracy and reliability is desirable? What level of measurement accuracy and reliability is realistically attainable? To what extent can the measurement system permissibly interfere with the monitored network components and variables? What is the acceptable cost of measurement? The answers to these questions will determine the measurement tools and methodologies appropriate in any given traffic engineering context.

It should also be noted that there is a distinction between measurement and evaluation. Measurement provides raw data concerning state parameters and variables of monitored network elements. Evaluation utilizes the raw data to make inferences regarding the monitored system.

Measurement in support of the TE function can occur at different levels of abstraction. For example, measurement can be used to derive packet level characteristics, flow level characteristics, user or customer level characteristics, traffic aggregate characteristics, component level characteristics, and network wide characteristics.

3.3 Modeling, Analysis, and Simulation

Modeling and analysis are important aspects of Internet traffic engineering. Modeling involves constructing an abstract or physical representation which depicts relevant traffic characteristics and network attributes.

A network model is an abstract representation of the network which captures relevant network features, attributes, and characteristics, such as link and nodal attributes and constraints. A network model may facilitate analysis and/or simulation which can be used to predict network performance under various conditions as well as to guide network expansion plans.

In general, Internet traffic engineering models can be classified as either structural or behavioral. Structural models focus on the organization of the network and its components. Behavioral models focus on the dynamics of the network and the traffic workload. Modeling for Internet traffic engineering may also be formal or informal.

Accurate behavioral models for traffic sources are particularly useful for analysis. Development of behavioral traffic source models that are consistent with empirical data obtained from operational networks is a major research topic in Internet traffic engineering. These source models should also be tractable and amenable to analysis. The topic of source models for IP traffic is a research topic and is therefore outside the scope of this document. Its importance, however, must be emphasized.

Network simulation tools are extremely useful for traffic engineering. Because of the complexity of realistic quantitative analysis of network behavior, certain aspects of network performance studies can only be conducted effectively using simulation. A good network simulator can be used to mimic and visualize network characteristics under various conditions in a safe and non-disruptive manner. For example, a network simulator may be used to depict congested resources and hot spots, and to provide hints regarding possible solutions to network performance problems. A good simulator may also be used to validate the effectiveness of planned solutions to network issues without the need to tamper with the operational network, or to commence an expensive network upgrade which may not

achieve the desired objectives. Furthermore, during the process of network planning, a network simulator may reveal pathologies such as single points of failure which may require additional redundancy, and potential bottlenecks and hot spots which may require additional capacity.

Routing simulators are especially useful in large networks. A routing simulator may identify planned links which may not actually be used to route traffic by the existing routing protocols. Simulators can also be used to conduct scenario based and perturbation based analysis, as well as sensitivity studies. Simulation results can be used to initiate appropriate actions in various ways. For example, an important application of network simulation tools is to investigate and identify how best to make the network evolve and grow, in order to accommodate projected future demands.

3.4 Optimization

Network performance optimization involves resolving network issues by transforming such issues into concepts that enable a solution, identification of a solution, and implementation of the solution. Network performance optimization can be corrective or perfective. In corrective optimization, the goal is to remedy a problem that has occurred or that is incipient. In perfective optimization, the goal is to improve network performance even when explicit problems do not exist and are not anticipated.

Network performance optimization is a continual process, as noted previously. Performance optimization iterations may consist of real-time optimization sub-processes and non-real-time network planning sub-processes. The difference between real-time optimization and network planning is primarily in the relative time-scale in which they operate and in the granularity of actions. One of the objectives of a real-time optimization sub-process is to control the mapping and distribution of traffic over the existing network infrastructure to avoid and/or relieve congestion, to assure satisfactory service delivery, and to optimize resource utilization. Real-time optimization is needed because random incidents such as fiber cuts or shifts in traffic demand will occur irrespective of how well a network is designed. These incidents can cause congestion and other problems to manifest in an operational network. Real-time optimization must solve such problems in small to medium time-scales ranging from micro-seconds to minutes or hours. Examples of real-time optimization include queue management, IGP/BGP metric tuning, and using technologies such as MPLS explicit LSPs to change the paths of some traffic trunks [XIAO].

One of the functions of the network planning sub-process is to initiate actions to systematically evolve the architecture, technology, topology, and capacity of a network. When a problem exists in the network, real-time optimization should provide an immediate remedy. Because a prompt response is necessary, the real-time solution may not be the best possible solution. Network planning may subsequently be needed to refine the solution and improve the situation. Network planning is also required to expand the network to support traffic growth and changes in traffic distribution over time. As previously noted, a change in the topology and/or capacity of the network may be the outcome of network planning.

Clearly, network planning and real-time performance optimization are mutually complementary activities. A well-planned and designed network makes real-time optimization easier, while a systematic approach to real-time network performance optimization allows network planning to focus on long term issues rather than tactical considerations. Systematic real-time network performance optimization also provides valuable inputs and insights toward network planning.

Stability is an important consideration in real-time network performance optimization. This aspect will be repeatedly addressed throughout this memo.

4.0 Historical Review and Recent Developments

This section briefly reviews different traffic engineering approaches proposed and implemented in telecommunications and computer networks. The discussion is not intended to be comprehensive. It is primarily intended to illuminate pre-existing perspectives and prior art concerning traffic engineering in the Internet and in legacy telecommunications networks.

4.1 Traffic Engineering in Classical Telephone Networks

This subsection presents a brief overview of traffic engineering in telephone networks which often relates to the way user traffic is steered from an originating node to the terminating node. This subsection presents a brief overview of this topic. A detailed description of the various routing strategies applied in telephone networks is included in the book by G. Ash [ASH2].

The early telephone network relied on static hierarchical routing, whereby routing patterns remained fixed independent of the state of the network or time of day. The hierarchy was intended to accommodate overflow traffic, improve network reliability via

alternate routes, and prevent call looping by employing strict hierarchical rules. The network was typically over-provisioned since a given fixed route had to be dimensioned so that it could carry user traffic during a busy hour of any busy day. Hierarchical routing in the telephony network was found to be too rigid upon the advent of digital switches and stored program control which were able to manage more complicated traffic engineering rules.

Dynamic routing was introduced to alleviate the routing inflexibility in the static hierarchical routing so that the network would operate more efficiently. This resulted in significant economic gains [HUSS87]. Dynamic routing typically reduces the overall loss probability by 10 to 20 percent (compared to static hierarchical routing). Dynamic routing can also improve network resilience by recalculating routes on a per-call basis and periodically updating routes.

There are three main types of dynamic routing in the telephone network. They are time-dependent routing, state-dependent routing (SDR), and event dependent routing (EDR).

In time-dependent routing, regular variations in traffic loads (such as time of day or day of week) are exploited in pre-planned routing tables. In state-dependent routing, routing tables are updated online according to the current state of the network (e.g., traffic demand, utilization, etc.). In event dependent routing, routing changes are incepted by events (such as call setups encountering congested or blocked links) whereupon new paths are searched out using learning models. EDR methods are real-time adaptive, but they do not require global state information as does SDR. Examples of EDR schemes include the dynamic alternate routing (DAR) from BT, the state-and-time dependent routing (STR) from NTT, and the success-to-the-top (STT) routing from AT&T.

Dynamic non-hierarchical routing (DNHR) is an example of dynamic routing that was introduced in the AT&T toll network in the 1980's to respond to time-dependent information such as regular load variations as a function of time. Time-dependent information in terms of load may be divided into three time scales: hourly, weekly, and yearly. Correspondingly, three algorithms are defined to pre-plan the routing tables. The network design algorithm operates over a year-long interval while the demand servicing algorithm operates on a weekly basis to fine tune link sizes and routing tables to correct forecast errors on the yearly basis. At the smallest time scale, the routing algorithm is used to make limited adjustments based on daily traffic variations. Network design and demand servicing are computed using offline calculations. Typically, the calculations require extensive searches on possible routes. On the other hand, routing may need

online calculations to handle crankback. DNHR adopts a "two-link" approach whereby a path can consist of two links at most. The routing algorithm presents an ordered list of route choices between an originating switch and a terminating switch. If a call overflows, a via switch (a tandem exchange between the originating switch and the terminating switch) would send a crankback signal to the originating switch. This switch would then select the next route, and so on, until there are no alternative routes available in which the call is blocked.

4.2 Evolution of Traffic Engineering in Packet Networks

This subsection reviews related prior work that was intended to improve the performance of data networks. Indeed, optimization of the performance of data networks started in the early days of the ARPANET. Other early commercial networks such as SNA also recognized the importance of performance optimization and service differentiation.

In terms of traffic management, the Internet has been a best effort service environment until recently. In particular, very limited traffic management capabilities existed in IP networks to provide differentiated queue management and scheduling services to packets belonging to different classes.

In terms of routing control, the Internet has employed distributed protocols for intra-domain routing. These protocols are highly scalable and resilient. However, they are based on simple algorithms for path selection which have very limited functionality to allow flexible control of the path selection process.

In the following subsections, the evolution of practical traffic engineering mechanisms in IP networks and its predecessors are reviewed.

4.2.1 Adaptive Routing in the ARPANET

The early ARPANET recognized the importance of adaptive routing where routing decisions were based on the current state of the network [MCQ80]. Early minimum delay routing approaches forwarded each packet to its destination along a path for which the total estimated transit time was the smallest. Each node maintained a table of network delays, representing the estimated delay that a packet would experience along a given path toward its destination. The minimum delay table was periodically transmitted by a node to its neighbors. The shortest path, in terms of hop count, was also propagated to give the connectivity information.

One drawback to this approach is that dynamic link metrics tend to create "traffic magnets" causing congestion to be shifted from one location of a network to another location, resulting in oscillation and network instability.

4.2.2 Dynamic Routing in the Internet

The Internet evolved from the APARNET and adopted dynamic routing algorithms with distributed control to determine the paths that packets should take en-route to their destinations. The routing algorithms are adaptations of shortest path algorithms where costs are based on link metrics. The link metric can be based on static or dynamic quantities. The link metric based on static quantities may be assigned administratively according to local criteria. The link metric based on dynamic quantities may be a function of a network congestion measure such as delay or packet loss.

It was apparent early that static link metric assignment was inadequate because it can easily lead to unfavorable scenarios in which some links become congested while others remain lightly loaded. One of the many reasons for the inadequacy of static link metrics is that link metric assignment was often done without considering the traffic matrix in the network. Also, the routing protocols did not take traffic attributes and capacity constraints into account when making routing decisions. This results in traffic concentration being localized in subsets of the network infrastructure and potentially causing congestion. Even if link metrics are assigned in accordance with the traffic matrix, unbalanced loads in the network can still occur due to a number factors including:

- Resources may not be deployed in the most optimal locations from a routing perspective.

- Forecasting errors in traffic volume and/or traffic distribution.

- Dynamics in traffic matrix due to the temporal nature of traffic patterns, BGP policy change from peers, etc.

The inadequacy of the legacy Internet interior gateway routing system is one of the factors motivating the interest in path oriented technology with explicit routing and constraint-based routing capability such as MPLS.

4.2.3 ToS Routing

Type-of-Service (ToS) routing involves different routes going to the same destination with selection dependent upon the ToS field of an IP packet [RFC-2474]. The ToS classes may be classified as low delay and high throughput. Each link is associated with multiple link costs and each link cost is used to compute routes for a particular ToS. A separate shortest path tree is computed for each ToS. The shortest path algorithm must be run for each ToS resulting in very expensive computation. Classical ToS-based routing is now outdated as the IP header field has been replaced by a Diffserv field. Effective traffic engineering is difficult to perform in classical ToS-based routing because each class still relies exclusively on shortest path routing which results in localization of traffic concentration within the network.

4.2.4 Equal Cost Multi-Path

Equal Cost Multi-Path (ECMP) is another technique that attempts to address the deficiency in the Shortest Path First (SPF) interior gateway routing systems [RFC-2328]. In the classical SPF algorithm, if two or more shortest paths exist to a given destination, the algorithm will choose one of them. The algorithm is modified slightly in ECMP so that if two or more equal cost shortest paths exist between two nodes, the traffic between the nodes is distributed among the multiple equal-cost paths. Traffic distribution across the equal-cost paths is usually performed in one of two ways: (1) packet-based in a round-robin fashion, or (2) flow-based using hashing on source and destination IP addresses and possibly other fields of the IP header. The first approach can easily cause out-of-order packets while the second approach is dependent upon the number and distribution of flows. Flow-based load sharing may be unpredictable in an enterprise network where the number of flows is relatively small and less heterogeneous (for example, hashing may not be uniform), but it is generally effective in core public networks where the number of flows is large and heterogeneous.

In ECMP, link costs are static and bandwidth constraints are not considered, so ECMP attempts to distribute the traffic as equally as possible among the equal-cost paths independent of the congestion status of each path. As a result, given two equal-cost paths, it is possible that one of the paths will be more congested than the other. Another drawback of ECMP is that load sharing cannot be achieved on multiple paths which have non-identical costs.

4.2.5 Nimrod

Nimrod is a routing system developed to provide heterogeneous service specific routing in the Internet, while taking multiple constraints into account [RFC-1992]. Essentially, Nimrod is a link state routing protocol which supports path oriented packet forwarding. It uses the concept of maps to represent network connectivity and services at multiple levels of abstraction. Mechanisms are provided to allow restriction of the distribution of routing information.

Even though Nimrod did not enjoy deployment in the public Internet, a number of key concepts incorporated into the Nimrod architecture, such as explicit routing which allows selection of paths at originating nodes, are beginning to find applications in some recent constraint-based routing initiatives.

4.3 Overlay Model

In the overlay model, a virtual-circuit network, such as ATM, frame relay, or WDM, provides virtual-circuit connectivity between routers that are located at the edges of a virtual-circuit cloud. In this mode, two routers that are connected through a virtual circuit see a direct adjacency between themselves independent of the physical route taken by the virtual circuit through the ATM, frame relay, or WDM network. Thus, the overlay model essentially decouples the logical topology that routers see from the physical topology that the ATM, frame relay, or WDM network manages. The overlay model based on ATM or frame relay enables a network administrator or an automaton to employ traffic engineering concepts to perform path optimization by re-configuring or rearranging the virtual circuits so that a virtual circuit on a congested or sub-optimal physical link can be re-routed to a less congested or more optimal one. In the overlay model, traffic engineering is also employed to establish relationships between the traffic management parameters (e.g., PCR, SCR, and MBS for ATM) of the virtual-circuit technology and the actual traffic that traverses each circuit. These relationships can be established based upon known or projected traffic profiles, and some other factors.

The overlay model using IP over ATM requires the management of two separate networks with different technologies (IP and ATM) resulting in increased operational complexity and cost. In the fully-meshed overlay model, each router would peer to every other router in the network, so that the total number of adjacencies is a quadratic function of the number of routers. Some of the issues with the overlay model are discussed in [AWD2].

4.4 Constrained-Based Routing

Constraint-based routing refers to a class of routing systems that compute routes through a network subject to the satisfaction of a set of constraints and requirements. In the most general setting, constraint-based routing may also seek to optimize overall network performance while minimizing costs.

The constraints and requirements may be imposed by the network itself or by administrative policies. Constraints may include bandwidth, hop count, delay, and policy instruments such as resource class attributes. Constraints may also include domain specific attributes of certain network technologies and contexts which impose restrictions on the solution space of the routing function. Path oriented technologies such as MPLS have made constraint-based routing feasible and attractive in public IP networks.

The concept of constraint-based routing within the context of MPLS traffic engineering requirements in IP networks was first defined in [RFC-2702].

Unlike QoS routing (for example, see [RFC-2386] and [MA]) which generally addresses the issue of routing individual traffic flows to satisfy prescribed flow based QoS requirements subject to network resource availability, constraint-based routing is applicable to traffic aggregates as well as flows and may be subject to a wide variety of constraints which may include policy restrictions.

4.5 Overview of Other IETF Projects Related to Traffic Engineering

This subsection reviews a number of IETF activities pertinent to Internet traffic engineering. These activities are primarily intended to evolve the IP architecture to support new service definitions which allow preferential or differentiated treatment to be accorded to certain types of traffic.

4.5.1 Integrated Services

The IETF Integrated Services working group developed the integrated services (Intserv) model. This model requires resources, such as bandwidth and buffers, to be reserved a priori for a given traffic flow to ensure that the quality of service requested by the traffic flow is satisfied. The integrated services model includes additional components beyond those used in the best-effort model such as packet classifiers, packet schedulers, and admission control. A packet classifier is used to identify flows that are to receive a certain level of service. A packet scheduler handles the scheduling of

service to different packet flows to ensure that QoS commitments are met. Admission control is used to determine whether a router has the necessary resources to accept a new flow.

Two services have been defined under the Integrated Services model: guaranteed service [RFC-2212] and controlled-load service [RFC-2211].

The guaranteed service can be used for applications requiring bounded packet delivery time. For this type of application, data that is delivered to the application after a pre-defined amount of time has elapsed is usually considered worthless. Therefore, guaranteed service was intended to provide a firm quantitative bound on the end-to-end packet delay for a flow. This is accomplished by controlling the queuing delay on network elements along the data flow path. The guaranteed service model does not, however, provide bounds on jitter (inter-arrival times between consecutive packets).

The controlled-load service can be used for adaptive applications that can tolerate some delay but are sensitive to traffic overload conditions. This type of application typically functions satisfactorily when the network is lightly loaded but its performance degrades significantly when the network is heavily loaded. Controlled-load service, therefore, has been designed to provide approximately the same service as best-effort service in a lightly loaded network regardless of actual network conditions. Controlled-load service is described qualitatively in that no target values of delay or loss are specified.

The main issue with the Integrated Services model has been scalability [RFC-2998], especially in large public IP networks which may potentially have millions of active micro-flows in transit concurrently.

A notable feature of the Integrated Services model is that it requires explicit signaling of QoS requirements from end systems to routers [RFC-2753]. The Resource Reservation Protocol (RSVP) performs this signaling function and is a critical component of the Integrated Services model. The RSVP protocol is described next.

4.5.2 RSVP

RSVP is a soft state signaling protocol [RFC-2205]. It supports receiver initiated establishment of resource reservations for both multicast and unicast flows. RSVP was originally developed as a signaling protocol within the integrated services framework for applications to communicate QoS requirements to the network and for the network to reserve relevant resources to satisfy the QoS requirements [RFC-2205].

Under RSVP, the sender or source node sends a PATH message to the receiver with the same source and destination addresses as the traffic which the sender will generate. The PATH message contains: (1) a sender Tspec specifying the characteristics of the traffic, (2) a sender Template specifying the format of the traffic, and (3) an optional Adspec which is used to support the concept of one pass with advertising" (OPWA) [RFC-2205]. Every intermediate router along the path forwards the PATH Message to the next hop determined by the routing protocol. Upon receiving a PATH Message, the receiver responds with a RESV message which includes a flow descriptor used to request resource reservations. The RESV message travels to the sender or source node in the opposite direction along the path that the PATH message traversed. Every intermediate router along the path can reject or accept the reservation request of the RESV message. If the request is rejected, the rejecting router will send an error message to the receiver and the signaling process will terminate. If the request is accepted, link bandwidth and buffer space are allocated for the flow and the related flow state information is installed in the router.

One of the issues with the original RSVP specification was Scalability. This is because reservations were required for micro-flows, so that the amount of state maintained by network elements tends to increase linearly with the number of micro-flows. These issues are described in [RFC-2961].

Recently, RSVP has been modified and extended in several ways to mitigate the scaling problems. As a result, it is becoming a versatile signaling protocol for the Internet. For example, RSVP has been extended to reserve resources for aggregation of flows, to set up MPLS explicit label switched paths, and to perform other signaling functions within the Internet. There are also a number of proposals to reduce the amount of refresh messages required to maintain established RSVP sessions [RFC-2961].

A number of IETF working groups have been engaged in activities related to the RSVP protocol. These include the original RSVP working group, the MPLS working group, the Resource Allocation Protocol working group, and the Policy Framework working group.

4.5.3 Differentiated Services

The goal of the Differentiated Services (Diffserv) effort within the IETF is to devise scalable mechanisms for categorization of traffic into behavior aggregates, which ultimately allows each behavior aggregate to be treated differently, especially when there is a shortage of resources such as link bandwidth and buffer space [RFC-2475]. One of the primary motivations for the Diffserv effort was to

devise alternative mechanisms for service differentiation in the Internet that mitigate the scalability issues encountered with the Intserv model.

The IETF Diffserv working group has defined a Differentiated Services field in the IP header (DS field). The DS field consists of six bits of the part of the IP header formerly known as TOS octet. The DS field is used to indicate the forwarding treatment that a packet should receive at a node [RFC-2474]. The Diffserv working group has also standardized a number of Per-Hop Behavior (PHB) groups. Using the PHBs, several classes of services can be defined using different classification, policing, shaping, and scheduling rules.

For an end-user of network services to receive Differentiated Services from its Internet Service Provider (ISP), it may be necessary for the user to have a Service Level Agreement (SLA) with the ISP. An SLA may explicitly or implicitly specify a Traffic Conditioning Agreement (TCA) which defines classifier rules as well as metering, marking, discarding, and shaping rules.

Packets are classified, and possibly policed and shaped at the ingress to a Diffserv network. When a packet traverses the boundary between different Diffserv domains, the DS field of the packet may be re-marked according to existing agreements between the domains.

Differentiated Services allows only a finite number of service classes to be indicated by the DS field. The main advantage of the Diffserv approach relative to the Intserv model is scalability. Resources are allocated on a per-class basis and the amount of state information is proportional to the number of classes rather than to the number of application flows.

It should be obvious from the previous discussion that the Diffserv model essentially deals with traffic management issues on a per hop basis. The Diffserv control model consists of a collection of micro-TE control mechanisms. Other traffic engineering capabilities, such as capacity management (including routing control), are also required in order to deliver acceptable service quality in Diffserv networks. The concept of Per Domain Behaviors has been introduced to better capture the notion of differentiated services across a complete domain [RFC-3086].

4.5.4 MPLS

MPLS is an advanced forwarding scheme which also includes extensions to conventional IP control plane protocols. MPLS extends the Internet routing model and enhances packet forwarding and path control [RFC-3031].

At the ingress to an MPLS domain, label switching routers (LSRs) classify IP packets into forwarding equivalence classes (FECs) based on a variety of factors, including, e.g., a combination of the information carried in the IP header of the packets and the local routing information maintained by the LSRs. An MPLS label is then prepended to each packet according to their forwarding equivalence classes. In a non-ATM/FR environment, the label is 32 bits long and contains a 20-bit label field, a 3-bit experimental field (formerly known as Class-of-Service or CoS field), a 1-bit label stack indicator and an 8-bit TTL field. In an ATM (FR) environment, the label consists of information encoded in the VCI/VPI (DLCI) field. An MPLS capable router (an LSR) examines the label and possibly the experimental field and uses this information to make packet forwarding decisions.

An LSR makes forwarding decisions by using the label prepended to packets as the index into a local next hop label forwarding entry (NHLFE). The packet is then processed as specified in the NHLFE. The incoming label may be replaced by an outgoing label, and the packet may be switched to the next LSR. This label-switching process is very similar to the label (VCI/VPI) swapping process in ATM networks. Before a packet leaves an MPLS domain, its MPLS label may be removed. A Label Switched Path (LSP) is the path between an ingress LSRs and an egress LSRs through which a labeled packet traverses. The path of an explicit LSP is defined at the originating (ingress) node of the LSP. MPLS can use a signaling protocol such as RSVP or LDP to set up LSPs.

MPLS is a very powerful technology for Internet traffic engineering because it supports explicit LSPs which allow constraint-based routing to be implemented efficiently in IP networks [AWD2]. The requirements for traffic engineering over MPLS are described in [RFC-2702]. Extensions to RSVP to support instantiation of explicit LSP are discussed in [RFC-3209]. Extensions to LDP, known as CR-LDP, to support explicit LSPs are presented in [JAM].

4.5.5 IP Performance Metrics

The IETF IP Performance Metrics (IPPM) working group has been developing a set of standard metrics that can be used to monitor the quality, performance, and reliability of Internet services. These metrics can be applied by network operators, end-users, and independent testing groups to provide users and service providers with a common understanding of the performance and reliability of the Internet component 'clouds' they use/provide [RFC-2330]. The criteria for performance metrics developed by the IPPM WG are described in [RFC-2330]. Examples of performance metrics include one-way packet

loss [RFC-2680], one-way delay [RFC-2679], and connectivity measures between two nodes [RFC-2678]. Other metrics include second-order measures of packet loss and delay.

Some of the performance metrics specified by the IPPM WG are useful for specifying Service Level Agreements (SLAs). SLAs are sets of service level objectives negotiated between users and service providers, wherein each objective is a combination of one or more performance metrics, possibly subject to certain constraints.

4.5.6 Flow Measurement

The IETF Real Time Flow Measurement (RTFM) working group has produced an architecture document defining a method to specify traffic flows as well as a number of components for flow measurement (meters, meter readers, manager) [RFC-2722]. A flow measurement system enables network traffic flows to be measured and analyzed at the flow level for a variety of purposes. As noted in RFC 2722, a flow measurement system can be very useful in the following contexts: (1) understanding the behavior of existing networks, (2) planning for network development and expansion, (3) quantification of network performance, (4) verifying the quality of network service, and (5) attribution of network usage to users.

A flow measurement system consists of meters, meter readers, and managers. A meter observes packets passing through a measurement point, classifies them into certain groups, accumulates certain usage data (such as the number of packets and bytes for each group), and stores the usage data in a flow table. A group may represent a user application, a host, a network, a group of networks, etc. A meter reader gathers usage data from various meters so it can be made available for analysis. A manager is responsible for configuring and controlling meters and meter readers. The instructions received by a meter from a manager include flow specification, meter control parameters, and sampling techniques. The instructions received by a meter reader from a manager include the address of the meter whose date is to be collected, the frequency of data collection, and the types of flows to be collected.

4.5.7 Endpoint Congestion Management

[RFC-3124] is intended to provide a set of congestion control mechanisms that transport protocols can use. It is also intended to develop mechanisms for unifying congestion control across a subset of an endpoint’s active unicast connections (called a congestion group). A congestion manager continuously monitors the state of the path for

each congestion group under its control. The manager uses that information to instruct a scheduler on how to partition bandwidth among the connections of that congestion group.

4.6 Overview of ITU Activities Related to Traffic Engineering

This section provides an overview of prior work within the ITU-T pertaining to traffic engineering in traditional telecommunications networks.

ITU-T Recommendations E.600 [ITU-E600], E.701 [ITU-E701], and E.801 [ITU-E801] address traffic engineering issues in traditional telecommunications networks. Recommendation E.600 provides a vocabulary for describing traffic engineering concepts, while E.701 defines reference connections, Grade of Service (GOS), and traffic parameters for ISDN. Recommendation E.701 uses the concept of a reference connection to identify representative cases of different types of connections without describing the specifics of their actual realizations by different physical means. As defined in Recommendation E.600, "a connection is an association of resources providing means for communication between two or more devices in, or attached to, a telecommunication network." Also, E.600 defines "a resource as any set of physically or conceptually identifiable entities within a telecommunication network, the use of which can be unambiguously determined" [ITU-E600]. There can be different types of connections as the number and types of resources in a connection may vary.

Typically, different network segments are involved in the path of a connection. For example, a connection may be local, national, or international. The purposes of reference connections are to clarify and specify traffic performance issues at various interfaces between different network domains. Each domain may consist of one or more service provider networks.

Reference connections provide a basis to define grade of service (GoS) parameters related to traffic engineering within the ITU-T framework. As defined in E.600, "GoS refers to a number of traffic engineering variables which are used to provide a measure of the adequacy of a group of resources under specified conditions." These GoS variables may be probability of loss, dial tone, delay, etc. They are essential for network internal design and operation as well as for component performance specification.

GoS is different from quality of service (QoS) in the ITU framework. QoS is the performance perceivable by a telecommunication service user and expresses the user’s degree of satisfaction of the service. QoS parameters focus on performance aspects observable at the service

access points and network interfaces, rather than their causes within the network. GoS, on the other hand, is a set of network oriented measures which characterize the adequacy of a group of resources under specified conditions. For a network to be effective in serving its users, the values of both GoS and QoS parameters must be related, with GoS parameters typically making a major contribution to the QoS.

Recommendation E.600 stipulates that a set of GoS parameters must be selected and defined on an end-to-end basis for each major service category provided by a network to assist the network provider with improving efficiency and effectiveness of the network. Based on a selected set of reference connections, suitable target values are assigned to the selected GoS parameters under normal and high load conditions. These end-to-end GoS target values are then apportioned to individual resource components of the reference connections for dimensioning purposes.

4.7 Content Distribution

The Internet is dominated by client-server interactions, especially Web traffic (in the future, more sophisticated media servers may become dominant). The location and performance of major information servers has a significant impact on the traffic patterns within the Internet as well as on the perception of service quality by end users.

A number of dynamic load balancing techniques have been devised to improve the performance of replicated information servers. These techniques can cause spatial traffic characteristics to become more dynamic in the Internet because information servers can be dynamically picked based upon the location of the clients, the location of the servers, the relative utilization of the servers, the relative performance of different networks, and the relative performance of different parts of a network. This process of assignment of distributed servers to clients is called Traffic Directing. It functions at the application layer.

Traffic Directing schemes that allocate servers in multiple geographically dispersed locations to clients may require empirical network performance statistics to make more effective decisions. In the future, network measurement systems may need to provide this type of information. The exact parameters needed are not yet defined.

When congestion exists in the network, Traffic Directing and Traffic Engineering systems should act in a coordinated manner. This topic is for further study.

The issues related to location and replication of information servers, particularly web servers, are important for Internet traffic engineering because these servers contribute a substantial proportion of Internet traffic.

5.0 Taxonomy of Traffic Engineering Systems

This section presents a short taxonomy of traffic engineering systems. A taxonomy of traffic engineering systems can be constructed based on traffic engineering styles and views as listed below:

- Time-dependent vs State-dependent vs Event-dependent
- Offline vs Online
- Centralized vs Distributed
- Local vs Global Information
- Prescriptive vs Descriptive
- Open Loop vs Closed Loop
- Tactical vs Strategic

These classification systems are described in greater detail in the following subsections of this document.

5.1 Time-Dependent Versus State-Dependent Versus Event Dependent

Traffic engineering methodologies can be classified as time-dependent, or state-dependent, or event-dependent. All TE schemes are considered to be dynamic in this document. Static TE implies that no traffic engineering methodology or algorithm is being applied.

In the time-dependent TE, historical information based on periodic variations in traffic, (such as time of day), is used to pre-program routing plans and other TE control mechanisms. Additionally, customer subscription or traffic projection may be used. Pre-programmed routing plans typically change on a relatively long time scale (e.g., diurnal). Time-dependent algorithms do not attempt to adapt to random variations in traffic or changing network conditions. An example of a time-dependent algorithm is a global centralized optimizer where the input to the system is a traffic matrix and multi-class QoS requirements as described [MR99].

State-dependent TE adapts the routing plans for packets based on the current state of the network. The current state of the network provides additional information on variations in actual traffic (i.e., perturbations from regular variations) that could not be predicted using historical information. Constraint-based routing is

an example of state-dependent TE operating in a relatively long time scale. An example operating in a relatively short time scale is a load-balancing algorithm described in [MATE].

The state of the network can be based on parameters such as utilization, packet delay, packet loss, etc. These parameters can be obtained in several ways. For example, each router may flood these parameters periodically or by means of some kind of trigger to other routers. Another approach is for a particular router performing adaptive TE to send probe packets along a path to gather the state of that path. Still another approach is for a management system to gather relevant information from network elements.

Expeditious and accurate gathering and distribution of state information is critical for adaptive TE due to the dynamic nature of network conditions. State-dependent algorithms may be applied to increase network efficiency and resilience. Time-dependent algorithms are more suitable for predictable traffic variations. On the other hand, state-dependent algorithms are more suitable for adapting to the prevailing network state.

Event-dependent TE methods can also be used for TE path selection. Event-dependent TE methods are distinct from time-dependent and state-dependent TE methods in the manner in which paths are selected. These algorithms are adaptive and distributed in nature and typically use learning models to find good paths for TE in a network. While state-dependent TE models typically use available-link-bandwidth (ALB) flooding for TE path selection, event-dependent TE methods do not require ALB flooding. Rather, event-dependent TE methods typically search out capacity by learning models, as in the success-to-the-top (STT) method. ALB flooding can be resource intensive, since it requires link bandwidth to carry LSAs, processor capacity to process LSAs, and the overhead can limit area/autonomous system (AS) size. Modeling results suggest that event-dependent TE methods could lead to a reduction in ALB flooding overhead without loss of network throughput performance [ASH3].

5.2 Offline Versus Online

Traffic engineering requires the computation of routing plans. The computation may be performed offline or online. The computation can be done offline for scenarios where routing plans need not be executed in real-time. For example, routing plans computed from forecast information may be computed offline. Typically, offline computation is also used to perform extensive searches on multi-dimensional solution spaces.

Online computation is required when the routing plans must adapt to changing network conditions as in state-dependent algorithms. Unlike offline computation (which can be computationally demanding), online computation is geared toward relative simple and fast calculations to select routes, fine-tune the allocations of resources, and perform load balancing.

5.3 Centralized Versus Distributed

Centralized control has a central authority which determines routing plans and perhaps other TE control parameters on behalf of each router. The central authority collects the network-state information from all routers periodically and returns the routing information to the routers. The routing update cycle is a critical parameter directly impacting the performance of the network being controlled. Centralized control may need high processing power and high bandwidth control channels.

Distributed control determines route selection by each router autonomously based on the routers view of the state of the network. The network state information may be obtained by the router using a probing method or distributed by other routers on a periodic basis using link state advertisements. Network state information may also be disseminated under exceptional conditions.

5.4 Local Versus Global

Traffic engineering algorithms may require local or global network-state information.

Local information pertains to the state of a portion of the domain. Examples include the bandwidth and packet loss rate of a particular path. Local state information may be sufficient for certain instances of distributed-controlled TEs.

Global information pertains to the state of the entire domain undergoing traffic engineering. Examples include a global traffic matrix and loading information on each link throughout the domain of interest. Global state information is typically required with centralized control. Distributed TE systems may also need global information in some cases.

5.5 Prescriptive Versus Descriptive

TE systems may also be classified as prescriptive or descriptive.

Prescriptive traffic engineering evaluates alternatives and recommends a course of action. Prescriptive traffic engineering can be further categorized as either corrective or perfective. Corrective TE prescribes a course of action to address an existing or predicted anomaly. Perfective TE prescribes a course of action to evolve and improve network performance even when no anomalies are evident.

Descriptive traffic engineering, on the other hand, characterizes the state of the network and assesses the impact of various policies without recommending any particular course of action.

5.6 Open-Loop Versus Closed-Loop

Open-loop traffic engineering control is where control action does not use feedback information from the current network state. The control action may use its own local information for accounting purposes, however.

Closed-loop traffic engineering control is where control action utilizes feedback information from the network state. The feedback information may be in the form of historical information or current measurement.

5.7 Tactical vs Strategic

Tactical traffic engineering aims to address specific performance problems (such as hot-spots) that occur in the network from a tactical perspective, without consideration of overall strategic imperatives. Without proper planning and insights, tactical TE tends to be ad hoc in nature.

Strategic traffic engineering approaches the TE problem from a more organized and systematic perspective, taking into consideration the immediate and longer term consequences of specific policies and actions.

6.0 Recommendations for Internet Traffic Engineering

This section describes high level recommendations for traffic engineering in the Internet. These recommendations are presented in general terms.

The recommendations describe the capabilities needed to solve a traffic engineering problem or to achieve a traffic engineering objective. Broadly speaking, these recommendations can be categorized as either functional and non-functional recommendations.

Functional recommendations for Internet traffic engineering describe the functions that a traffic engineering system should perform. These functions are needed to realize traffic engineering objectives by addressing traffic engineering problems.

Non-functional recommendations for Internet traffic engineering relate to the quality attributes or state characteristics of a traffic engineering system. These recommendations may contain conflicting assertions and may sometimes be difficult to quantify precisely.

6.1 Generic Non-functional Recommendations

The generic non-functional recommendations for Internet traffic engineering include: usability, automation, scalability, stability, visibility, simplicity, efficiency, reliability, correctness, maintainability, extensibility, interoperability, and security. In a given context, some of these recommendations may be critical while others may be optional. Therefore, prioritization may be required during the development phase of a traffic engineering system (or components thereof) to tailor it to a specific operational context.

In the following paragraphs, some of the aspects of the non-functional recommendations for Internet traffic engineering are summarized.

Usability: Usability is a human factor aspect of traffic engineering systems. Usability refers to the ease with which a traffic engineering system can be deployed and operated. In general, it is desirable to have a TE system that can be readily deployed in an existing network. It is also desirable to have a TE system that is easy to operate and maintain.

Automation: Whenever feasible, a traffic engineering system should automate as many traffic engineering functions as possible to minimize the amount of human effort needed to control and analyze operational networks. Automation is particularly imperative in large scale public networks because of the high cost of the human aspects of network operations and the high risk of network problems caused by human errors. Automation may entail the incorporation of automatic feedback and intelligence into some components of the traffic engineering system.

Scalability: Contemporary public networks are growing very fast with respect to network size and traffic volume. Therefore, a TE system should be scalable to remain applicable as the network evolves. In particular, a TE system should remain functional as the network expands with regard to the number of routers and links, and with

respect to the traffic volume. A TE system should have a scalable architecture, should not adversely impair other functions and processes in a network element, and should not consume too much network resources when collecting and distributing state information or when exerting control.

Stability: Stability is a very important consideration in traffic engineering systems that respond to changes in the state of the network. State-dependent traffic engineering methodologies typically mandate a tradeoff between responsiveness and stability. It is strongly recommended that when tradeoffs are warranted between responsiveness and stability, that the tradeoff should be made in favor of stability (especially in public IP backbone networks).

Flexibility: A TE system should be flexible to allow for changes in optimization policy. In particular, a TE system should provide sufficient configuration options so that a network administrator can tailor the TE system to a particular environment. It may also be desirable to have both online and offline TE subsystems which can be independently enabled and disabled. TE systems that are used in multi-class networks should also have options to support class based performance evaluation and optimization.

Visibility: As part of the TE system, mechanisms should exist to collect statistics from the network and to analyze these statistics to determine how well the network is functioning. Derived statistics such as traffic matrices, link utilization, latency, packet loss, and other performance measures of interest which are determined from network measurements can be used as indicators of prevailing network conditions. Other examples of status information which should be observed include existing functional routing information (additionally, in the context of MPLS existing LSP routes), etc.

Simplicity: Generally, a TE system should be as simple as possible. More importantly, the TE system should be relatively easy to use (i.e., clean, convenient, and intuitive user interfaces). Simplicity in user interface does not necessarily imply that the TE system will use naive algorithms. When complex algorithms and internal structures are used, such complexities should be hidden as much as possible from the network administrator through the user interface.

Interoperability: Whenever feasible, traffic engineering systems and their components should be developed with open standards based interfaces to allow interoperation with other systems and components.

Security: Security is a critical consideration in traffic engineering systems. Such traffic engineering systems typically exert control over certain functional aspects of the network to achieve the desired

performance objectives. Therefore, adequate measures must be taken to safeguard the integrity of the traffic engineering system. Adequate measures must also be taken to protect the network from vulnerabilities that originate from security breaches and other impairments within the traffic engineering system.

The remainder of this section will focus on some of the high level functional recommendations for traffic engineering.

6.2 Routing Recommendations

Routing control is a significant aspect of Internet traffic engineering. Routing impacts many of the key performance measures associated with networks, such as throughput, delay, and utilization. Generally, it is very difficult to provide good service quality in a wide area network without effective routing control. A desirable routing system is one that takes traffic characteristics and network constraints into account during route selection while maintaining stability.

Traditional shortest path first (SPF) interior gateway protocols are based on shortest path algorithms and have limited control capabilities for traffic engineering [RFC-2702, AWD2]. These limitations include :

1. The well known issues with pure SPF protocols, which do not take network constraints and traffic characteristics into account during route selection. For example, since IGPs always use the shortest paths (based on administratively assigned link metrics) to forward traffic, load sharing cannot be accomplished among paths of different costs. Using shortest paths to forward traffic conserves network resources, but may cause the following problems: 1) If traffic from a source to a destination exceeds the capacity of a link along the shortest path, the link (hence the shortest path) becomes congested while a longer path between these two nodes may be under-utilized; 2) the shortest paths from different sources can overlap at some links. If the total traffic from the sources exceeds the capacity of any of these links, congestion will occur. Problems can also occur because traffic demand changes over time but network topology and routing configuration cannot be changed as rapidly. This causes the network topology and routing configuration to become sub-optimal over time, which may result in persistent congestion problems.

2. The Equal-Cost Multi-Path (ECMP) capability of SPF IGPs supports sharing of traffic among equal cost paths between two nodes. However, ECMP attempts to divide the traffic as equally as possible among the equal cost shortest paths. Generally, ECMP

does not support configurable load sharing ratios among equal cost paths. The result is that one of the paths may carry significantly more traffic than other paths because it may also carry traffic from other sources. This situation can result in congestion along the path that carries more traffic.

3. Modifying IGP metrics to control traffic routing tends to have network-wide effect. Consequently, undesirable and unanticipated traffic shifts can be triggered as a result. Recent work described in Section 8.0 may be capable of better control [FT00, FT01].

Because of these limitations, new capabilities are needed to enhance the routing function in IP networks. Some of these capabilities have been described elsewhere and are summarized below.

Constraint-based routing is desirable to evolve the routing architecture of IP networks, especially public IP backbones with complex topologies [RFC-2702]. Constraint-based routing computes routes to fulfill requirements subject to constraints. Constraints may include bandwidth, hop count, delay, and administrative policy instruments such as resource class attributes [RFC-2702, RFC-2386]. This makes it possible to select routes that satisfy a given set of requirements subject to network and administrative policy constraints. Routes computed through constraint-based routing are not necessarily the shortest paths. Constraint-based routing works best with path oriented technologies that support explicit routing, such as MPLS.

Constraint-based routing can also be used as a way to redistribute traffic onto the infrastructure (even for best effort traffic). For example, if the bandwidth requirements for path selection and reservable bandwidth attributes of network links are appropriately defined and configured, then congestion problems caused by uneven traffic distribution may be avoided or reduced. In this way, the performance and efficiency of the network can be improved.

A number of enhancements are needed to conventional link state IGPs, such as OSPF and IS-IS, to allow them to distribute additional state information required for constraint-based routing. These extensions to OSPF were described in [KATZ] and to IS-IS in [SMIT]. Essentially, these enhancements require the propagation of additional information in link state advertisements. Specifically, in addition to normal link-state information, an enhanced IGP is required to propagate topology state information needed for constraint-based routing. Some of the additional topology state information include link attributes such as reservable bandwidth and link resource class attribute (an administratively specified property of the link). The

resource class attribute concept was defined in [RFC-2702]. The additional topology state information is carried in new TLVs and sub-TLVs in IS-IS, or in the Opaque LSA in OSPF [SMIT, KATZ].

An enhanced link-state IGP may flood information more frequently than a normal IGP. This is because even without changes in topology, changes in reservable bandwidth or link affinity can trigger the enhanced IGP to initiate flooding. A tradeoff is typically required between the timeliness of the information flooded and the flooding frequency to avoid excessive consumption of link bandwidth and computational resources, and more importantly, to avoid instability.

In a TE system, it is also desirable for the routing subsystem to make the load splitting ratio among multiple paths (with equal cost or different cost) configurable. This capability gives network administrators more flexibility in the control of traffic distribution across the network. It can be very useful for avoiding/relieving congestion in certain situations. Examples can be found in [XIAO].

The routing system should also have the capability to control the routes of subsets of traffic without affecting the routes of other traffic if sufficient resources exist for this purpose. This capability allows a more refined control over the distribution of traffic across the network. For example, the ability to move traffic from a source to a destination away from its original path to another path (without affecting other traffic paths) allows traffic to be moved from resource-poor network segments to resource-rich segments. Path oriented technologies such as MPLS inherently support this capability as discussed in [AWD2].

Additionally, the routing subsystem should be able to select different paths for different classes of traffic (or for different traffic behavior aggregates) if the network supports multiple classes of service (different behavior aggregates).

6.3 Traffic Mapping Recommendations

Traffic mapping pertains to the assignment of traffic workload onto pre-established paths to meet certain requirements. Thus, while constraint-based routing deals with path selection, traffic mapping deals with the assignment of traffic to established paths which may have been selected by constraint-based routing or by some other means. Traffic mapping can be performed by time-dependent or state-dependent mechanisms, as described in Section 5.1.

An important aspect of the traffic mapping function is the ability to establish multiple paths between an originating node and a destination node, and the capability to distribute the traffic between the two nodes across the paths according to some policies. A pre-condition for this scheme is the existence of flexible mechanisms to partition traffic and then assign the traffic partitions onto the parallel paths. This requirement was noted in [RFC-2702]. When traffic is assigned to multiple parallel paths, it is recommended that special care should be taken to ensure proper ordering of packets belonging to the same application (or micro-flow) at the destination node of the parallel paths.

As a general rule, mechanisms that perform the traffic mapping functions should aim to map the traffic onto the network infrastructure to minimize congestion. If the total traffic load cannot be accommodated, or if the routing and mapping functions cannot react fast enough to changing traffic conditions, then a traffic mapping system may rely on short time scale congestion control mechanisms (such as queue management, scheduling, etc.) to mitigate congestion. Thus, mechanisms that perform the traffic mapping functions should complement existing congestion control mechanisms. In an operational network, it is generally desirable to map the traffic onto the infrastructure such that intra-class and inter-class resource contention are minimized.

When traffic mapping techniques that depend on dynamic state feedback (e.g., MATE and such like) are used, special care must be taken to guarantee network stability.

6.4 Measurement Recommendations

The importance of measurement in traffic engineering has been discussed throughout this document. Mechanisms should be provided to measure and collect statistics from the network to support the traffic engineering function. Additional capabilities may be needed to help in the analysis of the statistics. The actions of these mechanisms should not adversely affect the accuracy and integrity of the statistics collected. The mechanisms for statistical data acquisition should also be able to scale as the network evolves.

Traffic statistics may be classified according to long-term or short-term time scales. Long-term time scale traffic statistics are very useful for traffic engineering. Long-term time scale traffic statistics may capture or reflect periodicity in network workload (such as hourly, daily, and weekly variations in traffic profiles) as well as traffic trends. Aspects of the monitored traffic statistics may also depict class of service characteristics for a network supporting multiple classes of service. Analysis of the long-term

traffic statistics MAY yield secondary statistics such as busy hour characteristics, traffic growth patterns, persistent congestion problems, hot-spot, and imbalances in link utilization caused by routing anomalies.

A mechanism for constructing traffic matrices for both long-term and short-term traffic statistics should be in place. In multi-service IP networks, the traffic matrices may be constructed for different service classes. Each element of a traffic matrix represents a statistic of traffic flow between a pair of abstract nodes. An abstract node may represent a router, a collection of routers, or a site in a VPN.

Measured traffic statistics should provide reasonable and reliable indicators of the current state of the network on the short-term scale. Some short term traffic statistics may reflect link utilization and link congestion status. Examples of congestion indicators include excessive packet delay, packet loss, and high resource utilization. Examples of mechanisms for distributing this kind of information include SNMP, probing techniques, FTP, IGP link state advertisements, etc.

6.5 Network Survivability

Network survivability refers to the capability of a network to maintain service continuity in the presence of faults. This can be accomplished by promptly recovering from network impairments and maintaining the required QoS for existing services after recovery. Survivability has become an issue of great concern within the Internet community due to the increasing demands to carry mission critical traffic, real-time traffic, and other high priority traffic over the Internet. Survivability can be addressed at the device level by developing network elements that are more reliable; and at the network level by incorporating redundancy into the architecture, design, and operation of networks. It is recommended that a philosophy of robustness and survivability should be adopted in the architecture, design, and operation of traffic engineering that control IP networks (especially public IP networks). Because different contexts may demand different levels of survivability, the mechanisms developed to support network survivability should be flexible so that they can be tailored to different needs.

Failure protection and restoration capabilities have become available from multiple layers as network technologies have continued to improve. At the bottom of the layered stack, optical networks are now capable of providing dynamic ring and mesh restoration functionality at the wavelength level as well as traditional protection functionality. At the SONET/SDH layer survivability

capability is provided with Automatic Protection Switching (APS) as well as self-healing ring and mesh architectures. Similar functionality is provided by layer 2 technologies such as ATM (generally with slower mean restoration times). Rerouting is traditionally used at the IP layer to restore service following link and node outages. Rerouting at the IP layer occurs after a period of routing convergence which may require seconds to minutes to complete. Some new developments in the MPLS context make it possible to achieve recovery at the IP layer prior to convergence [SHAR].

To support advanced survivability requirements, path-oriented technologies such a MPLS can be used to enhance the survivability of IP networks in a potentially cost effective manner. The advantages of path oriented technologies such as MPLS for IP restoration becomes even more evident when class based protection and restoration capabilities are required.

Recently, a common suite of control plane protocols has been proposed for both MPLS and optical transport networks under the acronym Multi-protocol Lambda Switching [AWD1]. This new paradigm of Multi-protocol Lambda Switching will support even more sophisticated mesh restoration capabilities at the optical layer for the emerging IP over WDM network architectures.

Another important aspect regarding multi-layer survivability is that technologies at different layers provide protection and restoration capabilities at different temporal granularities (in terms of time scales) and at different bandwidth granularity (from packet-level to wavelength level). Protection and restoration capabilities can also be sensitive to different service classes and different network utility models.

The impact of service outages varies significantly for different service classes depending upon the effective duration of the outage. The duration of an outage can vary from milliseconds (with minor service impact) to seconds (with possible call drops for IP telephony and session time-outs for connection oriented transactions) to minutes and hours (with potentially considerable social and business impact).

Coordinating different protection and restoration capabilities across multiple layers in a cohesive manner to ensure network survivability is maintained at reasonable cost is a challenging task. Protection and restoration coordination across layers may not always be feasible, because networks at different layers may belong to different administrative domains.

The following paragraphs present some of the general recommendations for protection and restoration coordination.

- Protection and restoration capabilities from different layers should be coordinated whenever feasible and appropriate to provide network survivability in a flexible and cost effective manner. Minimization of function duplication across layers is one way to achieve the coordination. Escalation of alarms and other fault indicators from lower to higher layers may also be performed in a coordinated manner. A temporal order of restoration trigger timing at different layers is another way to coordinate multi-layer protection/restoration.

- Spare capacity at higher layers is often regarded as working traffic at lower layers. Placing protection/restoration functions in many layers may increase redundancy and robustness, but it should not result in significant and avoidable inefficiencies in network resource utilization.

- It is generally desirable to have protection and restoration schemes that are bandwidth efficient.

- Failure notification throughout the network should be timely and reliable.

- Alarms and other fault monitoring and reporting capabilities should be provided at appropriate layers.

6.5.1 Survivability in MPLS Based Networks

MPLS is an important emerging technology that enhances IP networks in terms of features, capabilities, and services. Because MPLS is path-oriented, it can potentially provide faster and more predictable protection and restoration capabilities than conventional hop by hop routed IP systems. This subsection describes some of the basic aspects and recommendations for MPLS networks regarding protection and restoration. See [SHAR] for a more comprehensive discussion on MPLS based recovery.

Protection types for MPLS networks can be categorized as link protection, node protection, path protection, and segment protection.

- Link Protection: The objective for link protection is to protect an LSP from a given link failure. Under link protection, the path of the protection or backup LSP (the secondary LSP) is disjoint from the path of the working or operational LSP at the particular link over which protection is required. When the protected link fails, traffic on the working LSP is switched over to the

protection LSP at the head-end of the failed link. This is a local repair method which can be fast. It might be more appropriate in situations where some network elements along a given path are less reliable than others.

- Node Protection: The objective of LSP node protection is to protect an LSP from a given node failure. Under node protection, the path of the protection LSP is disjoint from the path of the working LSP at the particular node to be protected. The secondary path is also disjoint from the primary path at all links associated with the node to be protected. When the node fails, traffic on the working LSP is switched over to the protection LSP at the upstream LSR directly connected to the failed node.

- Path Protection: The goal of LSP path protection is to protect an LSP from failure at any point along its routed path. Under path protection, the path of the protection LSP is completely disjoint from the path of the working LSP. The advantage of path protection is that the backup LSP protects the working LSP from all possible link and node failures along the path, except for failures that might occur at the ingress and egress LSRs, or for correlated failures that might impact both working and backup paths simultaneously. Additionally, since the path selection is end-to-end, path protection might be more efficient in terms of resource usage than link or node protection. However, path protection may be slower than link and node protection in general.

- Segment Protection: An MPLS domain may be partitioned into multiple protection domains whereby a failure in a protection domain is rectified within that domain. In cases where an LSP traverses multiple protection domains, a protection mechanism within a domain only needs to protect the segment of the LSP that lies within the domain. Segment protection will generally be faster than path protection because recovery generally occurs closer to the fault.

6.5.2 Protection Option

Another issue to consider is the concept of protection options. The protection option uses the notation m:n protection, where m is the number of protection LSPs used to protect n working LSPs. Feasible protection options follow.

- 1:1: one working LSP is protected/restored by one protection LSP.

- 1:n: one protection LSP is used to protect/restore n working LSPs.

- n:1: one working LSP is protected/restored by n protection LSPs, possibly with configurable load splitting ratio. When more than one protection LSP is used, it may be desirable to share the traffic across the protection LSPs when the working LSP fails to satisfy the bandwidth requirement of the traffic trunk associated with the working LSP. This may be especially useful when it is not feasible to find one path that can satisfy the bandwidth requirement of the primary LSP.

- 1+1: traffic is sent concurrently on both the working LSP and the protection LSP. In this case, the egress LSR selects one of the two LSPs based on a local traffic integrity decision process, which compares the traffic received from both the working and the protection LSP and identifies discrepancies. It is unlikely that this option would be used extensively in IP networks due to its resource utilization inefficiency. However, if bandwidth becomes plentiful and cheap, then this option might become quite viable and attractive in IP networks.

6.6 Traffic Engineering in Diffserv Environments

This section provides an overview of the traffic engineering features and recommendations that are specifically pertinent to Differentiated Services (Diffserv) [RFC-2475] capable IP networks.

Increasing requirements to support multiple classes of traffic, such as best effort and mission critical data, in the Internet calls for IP networks to differentiate traffic according to some criteria, and to accord preferential treatment to certain types of traffic. Large numbers of flows can be aggregated into a few behavior aggregates based on some criteria in terms of common performance requirements in terms of packet loss ratio, delay, and jitter; or in terms of common fields within the IP packet headers.

As Diffserv evolves and becomes deployed in operational networks, traffic engineering will be critical to ensuring that SLAs defined within a given Diffserv service model are met. Classes of service (CoS) can be supported in a Diffserv environment by concatenating per-hop behaviors (PHBs) along the routing path, using service provisioning mechanisms, and by appropriately configuring edge functionality such as traffic classification, marking, policing, and shaping. PHB is the forwarding behavior that a packet receives at a DS node (a Diffserv-compliant node). This is accomplished by means of buffer management and packet scheduling mechanisms. In this context, packets belonging to a class are those that are members of a corresponding ordering aggregate.

Traffic engineering can be used as a compliment to Diffserv mechanisms to improve utilization of network resources, but not as a necessary element in general. When traffic engineering is used, it can be operated on an aggregated basis across all service classes [RFC-3270] or on a per service class basis. The former is used to provide better distribution of the aggregate traffic load over the network resources. (See [RFC-3270] for detailed mechanisms to support aggregate traffic engineering.) The latter case is discussed below since it is specific to the Diffserv environment, with so called Diffserv-aware traffic engineering [DIFF_TE].

For some Diffserv networks, it may be desirable to control the performance of some service classes by enforcing certain relationships between the traffic workload contributed by each service class and the amount of network resources allocated or provisioned for that service class. Such relationships between demand and resource allocation can be enforced using a combination of, for example: (1) traffic engineering mechanisms on a per service class basis that enforce the desired relationship between the amount of traffic contributed by a given service class and the resources allocated to that class, and (2) mechanisms that dynamically adjust the resources allocated to a given service class to relate to the amount of traffic contributed by that service class.

It may also be desirable to limit the performance impact of high priority traffic on relatively low priority traffic. This can be achieved by, for example, controlling the percentage of high priority traffic that is routed through a given link. Another way to accomplish this is to increase link capacities appropriately so that lower priority traffic can still enjoy adequate service quality. When the ratio of traffic workload contributed by different service classes vary significantly from router to router, it may not suffice to rely exclusively on conventional IGP routing protocols or on traffic engineering mechanisms that are insensitive to different service classes. Instead, it may be desirable to perform traffic engineering, especially routing control and mapping functions, on a per service class basis. One way to accomplish this in a domain that supports both MPLS and Diffserv is to define class specific LSPs and to map traffic from each class onto one or more LSPs that correspond to that service class. An LSP corresponding to a given service class can then be routed and protected/restored in a class dependent manner, according to specific policies.

Performing traffic engineering on a per class basis may require certain per-class parameters to be distributed. Note that it is common to have some classes share some aggregate constraint (e.g., maximum bandwidth requirement) without enforcing the constraint on each individual class. These classes then can be grouped into a

class-type and per-class-type parameters can be distributed instead to improve scalability. It also allows better bandwidth sharing between classes in the same class-type. A class-type is a set of classes that satisfy the following two conditions:

1) Classes in the same class-type have common aggregate requirements to satisfy required performance levels.

2) There is no requirement to be enforced at the level of individual class in the class-type. Note that it is still possible, nevertheless, to implement some priority policies for classes in the same class-type to permit preferential access to the class-type bandwidth through the use of preemption priorities.

An example of the class-type can be a low-loss class-type that includes both AF1-based and AF2-based Ordering Aggregates. With such a class-type, one may implement some priority policy which assigns higher preemption priority to AF1-based traffic trunks over AF2-based ones, vice versa, or the same priority.

See [DIFF-TE] for detailed requirements on Diffserv-aware traffic engineering.

6.7 Network Controllability

Off-line (and on-line) traffic engineering considerations would be of limited utility if the network could not be controlled effectively to implement the results of TE decisions and to achieve desired network performance objectives. Capacity augmentation is a coarse grained solution to traffic engineering issues. However, it is simple and may be advantageous if bandwidth is abundant and cheap or if the current or expected network workload demands it. However, bandwidth is not always abundant and cheap, and the workload may not always demand additional capacity. Adjustments of administrative weights and other parameters associated with routing protocols provide finer grained control, but is difficult to use and imprecise because of the routing interactions that occur across the network. In certain network contexts, more flexible, finer grained approaches which provide more precise control over the mapping of traffic to routes and over the selection and placement of routes may be appropriate and useful.

Control mechanisms can be manual (e.g., administrative configuration), partially-automated (e.g., scripts) or fully-automated (e.g., policy based management systems). Automated mechanisms are particularly required in large scale networks. Multi-vendor interoperability can be facilitated by developing and deploying standardized management

systems (e.g., standard MIBs) and policies (PIBs) to support the control functions required to address traffic engineering objectives such as load distribution and protection/restoration.

Network control functions should be secure, reliable, and stable as these are often needed to operate correctly in times of network impairments (e.g., during network congestion or security attacks).

7.0 Inter-Domain Considerations

Inter-domain traffic engineering is concerned with the performance optimization for traffic that originates in one administrative domain and terminates in a different one.

Traffic exchange between autonomous systems in the Internet occurs through exterior gateway protocols. Currently, BGP [BGP4] is the standard exterior gateway protocol for the Internet. BGP provides a number of attributes and capabilities (e.g., route filtering) that can be used for inter-domain traffic engineering. More specifically, BGP permits the control of routing information and traffic exchange between Autonomous Systems (AS's) in the Internet. BGP incorporates a sequential decision process which calculates the degree of preference for various routes to a given destination network. There are two fundamental aspects to inter-domain traffic engineering using BGP:

- Route Redistribution: controlling the import and export of routes between AS's, and controlling the redistribution of routes between BGP and other protocols within an AS.

- Best path selection: selecting the best path when there are multiple candidate paths to a given destination network. Best path selection is performed by the BGP decision process based on a sequential procedure, taking a number of different considerations into account. Ultimately, best path selection under BGP boils down to selecting preferred exit points out of an AS towards specific destination networks. The BGP path selection process can be influenced by manipulating the attributes associated with the BGP decision process. These attributes include: NEXT-HOP, WEIGHT (Cisco proprietary which is also implemented by some other vendors), LOCAL-PREFERENCE, AS-PATH, ROUTE-ORIGIN, MULTI-EXIT-DESCRIMINATOR (MED), IGP METRIC, etc.

Route-maps provide the flexibility to implement complex BGP policies based on pre-configured logical conditions. In particular, Route-maps can be used to control import and export policies for incoming and outgoing routes, control the redistribution of routes between BGP and other protocols, and influence the selection of best paths by

manipulating the attributes associated with the BGP decision process. Very complex logical expressions that implement various types of policies can be implemented using a combination of Route-maps, BGP-attributes, Access-lists, and Community attributes.

When looking at possible strategies for inter-domain TE with BGP, it must be noted that the outbound traffic exit point is controllable, whereas the interconnection point where inbound traffic is received from an EBGP peer typically is not, unless a special arrangement is made with the peer sending the traffic. Therefore, it is up to each individual network to implement sound TE strategies that deal with the efficient delivery of outbound traffic from one's customers to one's peering points. The vast majority of TE policy is based upon a "closest exit" strategy, which offloads interdomain traffic at the nearest outbound peer point towards the destination autonomous system. Most methods of manipulating the point at which inbound traffic enters a network from an EBGP peer (inconsistent route announcements between peering points, AS pre-pending, and sending MEDs) are either ineffective, or not accepted in the peering community.

Inter-domain TE with BGP is generally effective, but it is usually applied in a trial-and-error fashion. A systematic approach for inter-domain traffic engineering is yet to be devised.

Inter-domain TE is inherently more difficult than intra-domain TE under the current Internet architecture. The reasons for this are both technical and administrative. Technically, while topology and link state information are helpful for mapping traffic more effectively, BGP does not propagate such information across domain boundaries for stability and scalability reasons. Administratively, there are differences in operating costs and network capacities between domains. Generally, what may be considered a good solution in one domain may not necessarily be a good solution in another domain. Moreover, it would generally be considered inadvisable for one domain to permit another domain to influence the routing and management of traffic in its network.

MPLS TE-tunnels (explicit LSPs) can potentially add a degree of flexibility in the selection of exit points for inter-domain routing. The concept of relative and absolute metrics can be applied to this purpose. The idea is that if BGP attributes are defined such that the BGP decision process depends on IGP metrics to select exit points for inter-domain traffic, then some inter-domain traffic destined to a given peer network can be made to prefer a specific exit point by establishing a TE-tunnel between the router making the selection to the peering point via a TE-tunnel and assigning the TE-tunnel a metric which is smaller than the IGP cost to all other peering

points. If a peer accepts and processes MEDs, then a similar MPLS TE-tunnel based scheme can be applied to cause certain entrance points to be preferred by setting MED to be an IGP cost, which has been modified by the tunnel metric.

Similar to intra-domain TE, inter-domain TE is best accomplished when a traffic matrix can be derived to depict the volume of traffic from one autonomous system to another.

Generally, redistribution of inter-domain traffic requires coordination between peering partners. An export policy in one domain that results in load redistribution across peer points with another domain can significantly affect the local traffic matrix inside the domain of the peering partner. This, in turn, will affect the intra-domain TE due to changes in the spatial distribution of traffic. Therefore, it is mutually beneficial for peering partners to coordinate with each other before attempting any policy changes that may result in significant shifts in inter-domain traffic. In certain contexts, this coordination can be quite challenging due to technical and non- technical reasons.

It is a matter of speculation as to whether MPLS, or similar technologies, can be extended to allow selection of constrained paths across domain boundaries.

8.0 Overview of Contemporary TE Practices in Operational IP Networks

This section provides an overview of some contemporary traffic engineering practices in IP networks. The focus is primarily on the aspects that pertain to the control of the routing function in operational contexts. The intent here is to provide an overview of the commonly used practices. The discussion is not intended to be exhaustive.

Currently, service providers apply many of the traffic engineering mechanisms discussed in this document to optimize the performance of their IP networks. These techniques include capacity planning for long time scales, routing control using IGP metrics and MPLS for medium time scales, the overlay model also for medium time scales, and traffic management mechanisms for short time scale.

When a service provider plans to build an IP network, or expand the capacity of an existing network, effective capacity planning should be an important component of the process. Such plans may take the following aspects into account: location of new nodes if any, existing and predicted traffic patterns, costs, link capacity, topology, routing design, and survivability.

Performance optimization of operational networks is usually an ongoing process in which traffic statistics, performance parameters, and fault indicators are continually collected from the network. This empirical data is then analyzed and used to trigger various traffic engineering mechanisms. Tools that perform what-if analysis can also be used to assist the TE process by allowing various scenarios to be reviewed before a new set of configurations are implemented in the operational network.

Traditionally, intra-domain real-time TE with IGP is done by increasing the OSPF or IS-IS metric of a congested link until enough traffic has been diverted from that link. This approach has some limitations as discussed in Section 6.2. Recently, some new intra-domain TE approaches/tools have been proposed [RR94][FT00][FT01][WANG]. Such approaches/tools take traffic matrix, network topology, and network performance objective(s) as input, and produce some link metrics and possibly some unequal load-sharing ratios to be set at the head-end routers of some ECMPs as output. These new progresses open new possibility for intra-domain TE with IGP to be done in a more systematic way.

The overlay model (IP over ATM or IP over Frame relay) is another approach which is commonly used in practice [AWD2]. The IP over ATM technique is no longer viewed favorably due to recent advances in MPLS and router hardware technology.

Deployment of MPLS for traffic engineering applications has commenced in some service provider networks. One operational scenario is to deploy MPLS in conjunction with an IGP (IS-IS-TE or OSPF-TE) that supports the traffic engineering extensions, in conjunction with constraint-based routing for explicit route computations, and a signaling protocol (e.g., RSVP-TE or CRLDP) for LSP instantiation.

In contemporary MPLS traffic engineering contexts, network administrators specify and configure link attributes and resource constraints such as maximum reservable bandwidth and resource class attributes for links (interfaces) within the MPLS domain. A link state protocol that supports TE extensions (IS-IS-TE or OSPF-TE) is used to propagate information about network topology and link attribute to all routers in the routing area. Network administrators also specify all the LSPs that are to originate each router. For each LSP, the network administrator specifies the destination node and the attributes of the LSP which indicate the requirements that to be satisfied during the path selection process. Each router then uses a local constraint-based routing process to compute explicit paths for all LSPs originating from it. Subsequently, a signaling

protocol is used to instantiate the LSPs. By assigning proper bandwidth values to links and LSPs, congestion caused by uneven traffic distribution can generally be avoided or mitigated.

The bandwidth attributes of LSPs used for traffic engineering can be updated periodically. The basic concept is that the bandwidth assigned to an LSP should relate in some manner to the bandwidth requirements of traffic that actually flows through the LSP. The traffic attribute of an LSP can be modified to accommodate traffic growth and persistent traffic shifts. If network congestion occurs due to some unexpected events, existing LSPs can be rerouted to alleviate the situation or network administrator can configure new LSPs to divert some traffic to alternative paths. The reservable bandwidth of the congested links can also be reduced to force some LSPs to be rerouted to other paths.

In an MPLS domain, a traffic matrix can also be estimated by monitoring the traffic on LSPs. Such traffic statistics can be used for a variety of purposes including network planning and network optimization. Current practice suggests that deploying an MPLS network consisting of hundreds of routers and thousands of LSPs is feasible. In summary, recent deployment experience suggests that MPLS approach is very effective for traffic engineering in IP networks [XIAO].

As mentioned previously in Section 7.0, one usually has no direct control over the distribution of inbound traffic. Therefore, the main goal of contemporary inter-domain TE is to optimize the distribution of outbound traffic between multiple inter-domain links. When operating a global network, maintaining the ability to operate the network in a regional fashion where desired, while continuing to take advantage of the benefits of a global network, also becomes an important objective.

Inter-domain TE with BGP usually begins with the placement of multiple peering interconnection points in locations that have high peer density, are in close proximity to originating/terminating traffic locations on one's own network, and are lowest in cost. There are generally several locations in each region of the world where the vast majority of major networks congregate and interconnect. Some location-decision problems that arise in association with inter-domain routing are discussed in [AWD5].

Once the locations of the interconnects are determined, and circuits are implemented, one decides how best to handle the routes heard from the peer, as well as how to propagate the peers' routes within one's own network. One way to engineer outbound traffic flows on a network with many EBGP peers is to create a hierarchy of peers. Generally,

the Local Preferences of all peers are set to the same value so that the shortest AS paths will be chosen to forward traffic. Then, by over-writing the inbound MED metric (Multi-exit-discriminator metric, also referred to as "BGP metric". Both terms are used interchangeably in this document) with BGP metrics to routes received at different peers, the hierarchy can be formed. For example, all Local Preferences can be set to 200, preferred private peers can be assigned a BGP metric of 50, the rest of the private peers can be assigned a BGP metric of 100, and public peers can be assigned a BGP metric of 600. "Preferred" peers might be defined as those peers with whom the most available capacity exists, whose customer base is larger in comparison to other peers, whose interconnection costs are the lowest, and with whom upgrading existing capacity is the easiest. In a network with low utilization at the edge, this works well. The same concept could be applied to a network with higher edge utilization by creating more levels of BGP metrics between peers, allowing for more granularity in selecting the exit points for traffic bound for a dual homed customer on a peer's network.

By only replacing inbound MED metrics with BGP metrics, only equal AS-Path length routes' exit points are being changed. (The BGP decision considers Local Preference first, then AS-Path length, and then BGP metric). For example, assume a network has two possible egress points, peer A and peer B. Each peer has 40% of the Internet's routes exclusively on its network, while the remaining 20% of the Internet's routes are from customers who dual home between A and B. Assume that both peers have a Local Preference of 200 and a BGP metric of 100. If the link to peer A is congested, increasing its BGP metric while leaving the Local Preference at 200 will ensure that the 20% of total routes belonging to dual homed customers will prefer peer B as the exit point. The previous example would be used in a situation where all exit points to a given peer were close to congestion levels, and traffic needed to be shifted away from that peer entirely.

When there are multiple exit points to a given peer, and only one of them is congested, it is not necessary to shift traffic away from the peer entirely, but only from the one congested circuit. This can be achieved by using passive IGP-metrics, AS-path filtering, or prefix filtering.

Occasionally, more drastic changes are needed, for example, in dealing with a "problem peer" who is difficult to work with on upgrades or is charging high prices for connectivity to their network. In that case, the Local Preference to that peer can be reduced below the level of other peers. This effectively reduces the amount of traffic sent to that peer to only originating traffic

(assuming no transit providers are involved). This type of change can affect a large amount of traffic, and is only used after other methods have failed to provide the desired results.

Although it is not much of an issue in regional networks, the propagation of a peer's routes back through the network must be considered when a network is peering on a global scale. Sometimes, business considerations can influence the choice of BGP policies in a given context. For example, it may be imprudent, from a business perspective, to operate a global network and provide full access to the global customer base to a small network in a particular country. However, for the purpose of providing one's own customers with quality service in a particular region, good connectivity to that in-country network may still be necessary. This can be achieved by assigning a set of communities at the edge of the network, which have a known behavior when routes tagged with those communities are propagating back through the core. Routes heard from local peers will be prevented from propagating back to the global network, whereas routes learned from larger peers may be allowed to propagate freely throughout the entire global network. By implementing a flexible community strategy, the benefits of using a single global AS Number (ASN) can be realized, while the benefits of operating regional networks can also be taken advantage of. An alternative to doing this is to use different ASNs in different regions, with the consequence that the AS path length for routes announced by that service provider will increase.

9.0 Conclusion

This document described principles for traffic engineering in the Internet. It presented an overview of some of the basic issues surrounding traffic engineering in IP networks. The context of TE was described, a TE process models and a taxonomy of TE styles were presented. A brief historical review of pertinent developments related to traffic engineering was provided. A survey of contemporary TE techniques in operational networks was presented. Additionally, the document specified a set of generic requirements, recommendations, and options for Internet traffic engineering.

10.0 Security Considerations

This document does not introduce new security issues.

11.0 Acknowledgments

The authors would like to thank Jim Boyle for inputs on the recommendations section, Francois Le Faucheur for inputs on Diffserv aspects, Blaine Christian for inputs on measurement, Gerald Ash for

inputs on routing in telephone networks and for text on event-dependent TE methods, Steven Wright for inputs on network controllability, and Jonathan Aufderheide for inputs on inter-domain TE with BGP. Special thanks to Randy Bush for proposing the TE taxonomy based on "tactical vs strategic" methods. The subsection describing an "Overview of ITU Activities Related to Traffic Engineering" was adapted from a contribution by Waisum Lai. Useful feedback and pointers to relevant materials were provided by J. Noel Chiappa. Additional comments were provided by Glenn Grotefeld during the working last call process. Finally, the authors would like to thank Ed Kern, the TEWG co-chair, for his comments and support.

12.0 References

[ASH2] J. Ash, Dynamic Routing in Telecommunications Networks, McGraw Hill, 1998.

[ASH3] Ash, J., "TE & QoS Methods for IP-, ATM-, & TDM-Based Networks", Work in Progress, March 2001.

[AWD1] D. Awduche and Y. Rekhter, "Multiprocotol Lambda Switching: Combining MPLS Traffic Engineering Control with Optical Crossconnects", IEEE Communications Magazine, March 2001.

[AWD2] D. Awduche, "MPLS and Traffic Engineering in IP Networks", IEEE Communications Magazine, Dec. 1999.

[AWD5] D. Awduche et al, "An Approach to Optimal Peering Between Autonomous Systems in the Internet", International Conference on Computer Communications and Networks (ICCCN'98), Oct. 1998.

[CRUZ] R. L. Cruz, "A Calculus for Network Delay, Part II: Network Analysis", IEEE Transactions on Information Theory, vol. 37, pp. 132-141, 1991.

[DIFF-TE] Le Faucheur, F., Nadeau, T., Tatham, M., Telkamp, T., Cooper, D., Boyle, J., Lai, W., Fang, L., Ash, J., Hicks, P., Chui, A., Townsend, W. and D. Skalecki, "Requirements for support of Diff-Serv-aware MPLS Traffic Engineering", Work in Progress, May 2001.

[ELW95] A. Elwalid, D. Mitra and R.H. Wentworth, "A New Approach for Allocating Buffers and Bandwidth to Heterogeneous, Regulated Traffic in an ATM Node", IEEE IEEE Journal on Selected Areas in Communications, 13:6, pp. 1115-1127, Aug. 1995.

[FGLR] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford, "NetScope: Traffic Engineering for IP Networks", IEEE Network Magazine, 2000.

[FLJA93] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, Vol. 1 Nov. 4., p. 387-413, Aug. 1993.

[FLOY94] S. Floyd, "TCP and Explicit Congestion Notification", ACM Computer Communication Review, V. 24, No. 5, p. 10-23, Oct. 1994.

[FT00] B. Fortz and M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights", IEEE INFOCOM 2000, Mar. 2000.

[FT01] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS Weights in a Changing World", www.research.att.com/~mthorup/PAPERS/papers.html.

[HUSS87] B.R. Hurley, C.J.R. Seidl and W.F. Sewel, "A Survey of Dynamic Routing Methods for Circuit-Switched Traffic", IEEE Communication Magazine, Sep. 1987.

[ITU-E600] ITU-T Recommendation E.600, "Terms and Definitions of Traffic Engineering", Mar. 1993.

[ITU-E701] ITU-T Recommendation E.701, "Reference Connections for Traffic Engineering", Oct. 1993.

[ITU-E801] ITU-T Recommendation E.801, "Framework for Service Quality Agreement", Oct. 1996.

[JAM] Jamoussi, B., Editior, Andersson, L., Collon, R. and R. Dantu, "Constraint-Based LSP Setup using LDP", RFC 3212, January 2002.

[KATZ] Katz, D., Yeung, D. and K. Kompella, "Traffic Engineering Extensions to OSPF", Work in Progress, February 2001.

[LNO96] T. Lakshman, A. Neidhardt, and T. Ott, "The Drop from Front Strategy in TCP over ATM and its Interworking with other Control Features", Proc. INFOCOM'96, p. 1242-1250, 1996.

[MA] Q. Ma, "Quality of Service Routing in Integrated Services Networks", PhD Dissertation, CMU-CS-98-138, CMU, 1998.

[MATE] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "MATE: MPLS Adaptive Traffic Engineering", Proc. INFOCOM'01, Apr. 2001.

[MCQ80] J.M. McQuillan, I. Richer, and E.C. Rosen, "The New Routing Algorithm for the ARPANET", IEEE. Trans. on Communications, vol. 28, no. 5, pp. 711-719, May 1980.

[MR99] D. Mitra and K.G. Ramakrishnan, "A Case Study of Multiservice, Multipriority Traffic Engineering Design for Data Networks", Proc. Globecom'99, Dec 1999.

[RFC-1458] Braudes, R. and S. Zabele, "Requirements for Multicast Protocols", RFC 1458, May 1993.

[RFC-1771] Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

[RFC-1812] Baker, F., "Requirements for IP Version 4 Routers", STD 4, RFC 1812, June 1995.

[RFC-1992] Castineyra, I., Chiappa, N. and M. Steenstrup, "The Nimrod Routing Architecture", RFC 1992, August 1996.

[RFC-1997] Chandra, R., Traina, P. and T. Li, "BGP Community Attributes", RFC 1997, August 1996.

[RFC-1998] Chen, E. and T. Bates, "An Application of the BGP Community Attribute in Multi-home Routing", RFC 1998, August 1996.

[RFC-2205] Braden, R., Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource Reservation Protocol (RSVP) - Version 1 Functional Specification", RFC 2205, September 1997.

[RFC-2211] Wroclawski, J., "Specification of the Controlled-Load Network Element Service", RFC 2211, September 1997.

[RFC-2212] Shenker, S., Partridge, C. and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, September 1997.

[RFC-2215] Shenker, S. and J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements", RFC 2215, September 1997.

[RFC-2216] Shenker, S. and J. Wroclawski, "Network Element Service Specification Template", RFC 2216, September 1997.

[RFC-2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, July 1997.

[RFC-2330] Paxson, V., Almes, G., Mahdavi, J. and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.

[RFC-2386] Crawley, E., Nair, R., Rajagopalan, B. and H. Sandick, "A Framework for QoS-based Routing in the Internet", RFC 2386, August 1998.

[RFC-2474] Nichols, K., Blake, S., Baker, F. and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC-2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[RFC-2597] Heinanen, J., Baker, F., Weiss, W. and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, June 1999.

[RFC-2678] Mahdavi, J. and V. Paxson, "IPPM Metrics for Measuring Connectivity", RFC 2678, September 1999.

[RFC-2679] Almes, G., Kalidindi, S. and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.

[RFC-2680] Almes, G., Kalidindi, S. and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.

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

[RFC-2722] Brownlee, N., Mills, C. and G. Ruth, "Traffic Flow Measurement: Architecture", RFC 2722, October 1999.

[RFC-2753] Yavatkar, R., Pendarakis, D. and R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, January 2000.

[RFC-2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F. and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, April 2000.

[RFC-2998] Bernet, Y., Ford, P., Yavatkar, R., Baker, F., Zhang, L., Speer, M., Braden, R., Davie, B., Wroclawski, J. and E. Felstaine, "A Framework for Integrated Services Operation over Diffserv Networks", RFC 2998, November 2000.

[RFC-3031] Rosen, E., Viswanathan, A. and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[RFC-3086] Nichols, K. and B. Carpenter, "Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification", RFC 3086, April 2001.

[RFC-3124] Balakrishnan, H. and S. Seshan, "The Congestion Manager", RFC 3124, June 2001.

[RFC-3209] 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.

[RFC-3210] Awduche, D., Hannan, A. and X. Xiao, "Applicability Statement for Extensions to RSVP for LSP-Tunnels", RFC 3210, December 2001.

[RFC-3213] Ash, J., Girish, M., Gray, E., Jamoussi, B. and G. Wright, "Applicability Statement for CR-LDP", RFC 3213, January 2002.

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

[RR94] M.A. Rodrigues and K.G. Ramakrishnan, "Optimal Routing in Shortest Path Networks", ITS'94, Rio de Janeiro, Brazil.

[SHAR] Sharma, V., Crane, B., Owens, K., Huang, C., Hellstrand, F., Weil, J., Anderson, L., Jamoussi, B., Cain, B., Civanlar, S. and A. Chui, "Framework for MPLS Based Recovery", Work in Progress.

[SLDC98] B. Suter, T. Lakshman, D. Stiliadis, and A. Choudhury, "Design Considerations for Supporting TCP with Per-flow Queueing", Proc. INFOCOM'98, p. 299-306, 1998.

[SMIT] Smit, H. and T. Li, "IS-IS extensions for Traffic Engineering", Work in Progress.

[WANG] Y. Wang, Z. Wang, L. Zhang, "Internet traffic engineering without full mesh overlaying", Proceedings of INFOCOM'2001, April 2001.

[XIAO] X. Xiao, A. Hannan, B. Bailey, L. Ni, "Traffic Engineering with MPLS in the Internet", IEEE Network magazine, Mar. 2000.

[YARE95] C. Yang and A. Reddy, "A Taxonomy for Congestion Control Algorithms in Packet Switching Networks", IEEE Network Magazine, p. 34-45, 1995.

13.0 Authors' Addresses

Daniel O. Awduche
Movaz Networks
7926 Jones Branch Drive, Suite 615
McLean, VA 22102

Phone: 703-298-5291
EMail: awduche@movaz.com

Angela Chiu
Celion Networks
1 Sheila Dr., Suite 2
Tinton Falls, NJ 07724

Phone: 732-747-9987
EMail: angela.chiu@celion.com

Anwar Elwalid
Lucent Technologies
Murray Hill, NJ 07974

Phone: 908 582-7589
EMail: anwar@lucent.com

Indra Widjaja
Bell Labs, Lucent Technologies
600 Mountain Avenue
Murray Hill, NJ 07974

Phone: 908 582-0435
EMail: iwidjaja@research.bell-labs.com

XiPeng Xiao
Redback Networks
300 Holger Way
San Jose, CA 95134

Phone: 408-750-5217
EMail: xipeng@redback.com

14.0 Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

Exhibit 3

Network Working Group
Request for Comments: 3386
Category: Informational

W. Lai, Ed.
AT&T
D. McDysan, Ed.
WorldCom

November 2002

Network Hierarchy and Multilayer Survivability

Status of this Memo

Copyright Notice

Abstract

This document presents a proposal of the near-term and practical requirements for network survivability and hierarchy in current service provider environments.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [2].

Table of Contents

1. Introduction

This document is the result of the Network Hierarchy and Survivability Techniques Design Team established within the Traffic Engineering Working Group. This team collected and documented current and near term requirements for survivability and hierarchy in service provider environments. For clarity, an expanded set of definitions is included. The team determined that there appears to be a need to define a small set of interoperable survivability approaches in packet and non-packet networks. Suggested approaches include path-based as well as one that repairs connections in

proximity to the network fault. They operate primarily at a single network layer. For hierarchy, there did not appear to be a driving near-term need for work on "vertical hierarchy," defined as communication between network layers such as Time Division Multiplexed (TDM)/optical and Multi-Protocol Label Switching (MPLS). In particular, instead of direct exchange of signaling and routing between vertical layers, some looser form of coordination and communication, such as the specification of hold-off timers, is a nearer term need. For "horizontal hierarchy" in data networks, there are several pressing needs. The requirement is to be able to set up many Label Switched Paths (LSPs) in a service provider network with hierarchical Interior Gateway Protocol (IGP). This is necessary to support layer 2 and layer 3 Virtual Private Network (VPN) services that require edge-to-edge signaling across a core network.

This document presents a proposal of the near-term and practical requirements for network survivability and hierarchy in current service provider environments. With feedback from the working group solicited, the objective is to help focus the work that is being addressed in the TEWG (Traffic Engineering Working Group), CCAMP (Common Control and Measurement Plane Working Group), and other working groups. A main goal of this work is to provide some expedience for required functionality in multi-vendor service provider networks. The initial focus is primarily on intra-domain operations. However, to maintain consistency in the provision of end-to-end service in a multi-provider environment, rules governing the operations of survivability mechanisms at domain boundaries must also be specified. While such issues are raised and discussed, where appropriate, they will not be treated in depth in the initial release of this document.

The document first develops a set of definitions to be used later in this document and potentially in other documents as well. It then addresses the requirements and issues associated with service restoration, hierarchy, and finally a short discussion of survivability in hierarchical context.

Here is a summary of the findings:

A. Survivability Requirements

- o need to define a small set of interoperable survivability approaches in packet and non-packet networks
- o suggested survivability mechanisms include
  - 1:1 path protection with pre-established backup capacity (non-shared)
  - 1:1 path protection with pre-planned backup capacity (shared)

  - local restoration with repairs in proximity to the network fault
  - path restoration through source-based rerouting
- timing bounds for service restoration to support voice call cutoff (140 msec to 2 sec), protocol timer requirements in premium data services, and mission critical applications
- use of restoration priority for service differentiation

B. Hierarchy Requirements

B.1. Horizontally Oriented Hierarchy (Intra-Domain)

- ability to set up many LSPs in a service provider network with hierarchical IGP, for the support of layer 2 and layer 3 VPN services
- requirements for multi-area traffic engineering need to be developed to provide guidance for any necessary protocol extensions

B.2. Vertically Oriented Hierarchy

The following functionality for survivability is common on most routing equipment today.

- near-term need is some loose form of coordination and communication based on the use of nested hold-off timers, instead of direct exchange of signaling and routing between vertical layers
- means for an upper layer to immediately begin recovery actions in the event that a lower layer is not configured to perform recovery

C. Survivability Requirements in Horizontal Hierarchy

- protection of end-to-end connection is based on a concatenated set of connections, each protected within their area
- mechanisms for connection routing may include (1) a network element that participates on both sides of a boundary (e.g., OSPF ABR) - note that this is a common point of failure; (2) a route server
- need for inter-area signaling of survivability information (1) to enable a "least common denominator" survivability mechanism at the boundary; (2) to convey the success or failure of the service restoration action; e.g., if a part of a "connection" is down on one side of a boundary, there is no need for the other side to recover from failures

2. Terminology and Concepts

2.1 Hierarchy

Hierarchy is a technique used to build scalable complex systems. It is based on an abstraction, at each level, of what is most significant from the details and internal structures of the levels further away. This approach makes use of a general property of all hierarchical systems composed of related subsystems that interactions between subsystems decrease as the level of communication between subsystems decreases.

Network hierarchy is an abstraction of part of a network's topology, routing and signaling mechanisms. Abstraction may be used as a mechanism to build large networks or as a technique for enforcing administrative, topological, or geographic boundaries. For example, network hierarchy might be used to separate the metropolitan and long-haul regions of a network, or to separate the regional and backbone sections of a network, or to interconnect service provider networks (with BGP which reduces a network to an Autonomous System).

In this document, network hierarchy is considered from two perspectives:

(1) Vertically oriented: between two network technology layers.
(2) Horizontally oriented: between two areas or administrative subdivisions within the same network technology layer.

2.1.1 Vertical Hierarchy

Vertical hierarchy is the abstraction, or reduction in information, which would be of benefit when communicating information across network technology layers, as in propagating information between optical and router networks.

In the vertical hierarchy, the total network functions are partitioned into a series of functional or technological layers with clear logical, and maybe even physical, separation between adjacent layers. Survivability mechanisms either currently exist or are being developed at multiple layers in networks [3]. The optical layer is now becoming capable of providing dynamic ring and mesh restoration functionality, in addition to traditional 1+1 or 1:1 protection. The Synchronous Digital Hierarchy (SDH)/Synchronous Optical NETwork (SONET) layer provides survivability capability with automatic protection switching (APS), as well as self-healing ring and mesh restoration architectures. Similar functionality has been defined in the Asynchronous Transfer Mode (ATM) Layer, with work ongoing to also provide such functionality using MPLS [4]. At the IP layer,

rerouting is used to restore service continuity following link and node outages. Rerouting at the IP layer, however, occurs after a period of routing convergence, which may require a few seconds to several minutes to complete [5].

2.1.2 Horizontal Hierarchy

Horizontal hierarchy is the abstraction that allows a network at one technology layer, for instance a packet network, to scale. Examples of horizontal hierarchy include BGP confederations, separate Autonomous Systems, and multi-area OSPF.

In the horizontal hierarchy, a large network is partitioned into multiple smaller, non-overlapping sub-networks. The partitioning criteria can be based on topology, network function, administrative policy, or service domain demarcation. Two networks at the *same* hierarchical level, e.g., two Autonomous Systems in BGP, may share a peer relation with each other through some loose form of coupling. On the other hand, for routing in large networks using multi-area OSPF, abstraction through the aggregation of routing information is achieved through a hierarchical partitioning of the network.

2.2 Survivability Terminology

In alphabetical order, the following terms are defined in this section:

backup entity, same as protection entity (section 2.2.2)
extra traffic (section 2.2.2)
non-revertive mode (section 2.2.2)
normalization (section 2.2.2)
preemptable traffic, same as extra traffic (section 2.2.2)
preemption priority (section 2.2.4)
protection (section 2.2.3)
protection entity (section 2.2.2)
protection switching (section 2.2.3)
protection switch time (section 2.2.4)
recovery (section 2.2.2)
recovery by rerouting, same as restoration (section 2.2.3)
recovery entity, same as protection entity (section 2.2.2)
restoration (section 2.2.3)
restoration priority (section 2.2.4)
restoration time (section 2.2.4)
revertive mode (section 2.2.2)
shared risk group (SRG) (section 2.2.2)
survivability (section 2.2.1)
working entity (section 2.2.2)

2.2.1 Survivability

Survivability is the capability of a network to maintain service continuity in the presence of faults within the network [6]. Survivability mechanisms such as protection and restoration are implemented either on a per-link basis, on a per-path basis, or throughout an entire network to alleviate service disruption at affordable costs. The degree of survivability is determined by the network's capability to survive single failures, multiple failures, and equipment failures.

2.2.2 Generic Operations

This document does not discuss the sequence of events of how network failures are monitored, detected, and mitigated. For more detail of this aspect, see [4]. Also, the repair process following a failure is out of the scope here.

A working entity is the entity that is used to carry traffic in normal operation mode. Depending upon the context, an entity can be a channel or a transmission link in the physical layer, an Label Switched Path (LSP) in MPLS, or a logical bundle of one or more LSPs.

A protection entity, also called backup entity or recovery entity, is the entity that is used to carry protected traffic in recovery operation mode, i.e., when the working entity is in error or has failed.

Extra traffic, also referred to as preemptable traffic, is the traffic carried over the protection entity while the working entity is active. Extra traffic is not protected, i.e., when the protection entity is required to protect the traffic that is being carried over the working entity, the extra traffic is preempted.

A shared risk group (SRG) is a set of network elements that are collectively impacted by a specific fault or fault type. For example, a shared risk link group (SRLG) is the union of all the links on those fibers that are routed in the same physical conduit in a fiber-span network. This concept includes, besides shared conduit, other types of compromise such as shared fiber cable, shared right of way, shared optical ring, shared office without power sharing, etc. The span of an SRG, such as the length of the sharing for compromised outside plant, needs to be considered on a per fault basis. The concept of SRG can be extended to represent a "risk domain" and its associated capabilities and summarization for traffic engineering purposes. See [7] for further discussion.

Normalization is the sequence of events and actions taken by a network that returns the network to the preferred state upon completing repair of a failure. This could include the switching or rerouting of affected traffic to the original repaired working entities or new routes. Revertive mode refers to the case where traffic is automatically returned to a repaired working entity (also called switch back).

Recovery is the sequence of events and actions taken by a network after the detection of a failure to maintain the required performance level for existing services (e.g., according to service level agreements) and to allow normalization of the network. The actions include notification of the failure followed by two parallel processes: (1) a repair process with fault isolation and repair of the failed components, and (2) a reconfiguration process using survivability mechanisms to maintain service continuity. In protection, reconfiguration involves switching the affected traffic from a working entity to a protection entity. In restoration, reconfiguration involves path selection and rerouting for the affected traffic.

Revertive mode is a procedure in which revertive action, i.e., switch back from the protection entity to the working entity, is taken once the failed working entity has been repaired. In non-revertive mode, such action is not taken. To minimize service interruption, switch-back in revertive mode should be performed at a time when there is the least impact on the traffic concerned, or by using the make-before-break concept.

Non-revertive mode is the case where there is no preferred path or it may be desirable to minimize further disruption of the service brought on by a revertive switching operation. A switch-back to the original working path is not desired or not possible since the original path may no longer exist after the occurrence of a fault on that path.

2.2.3 Survivability Techniques

Protection, also called protection switching, is a survivability technique based on predetermined failure recovery: as the working entity is established, a protection entity is also established. Protection techniques can be implemented by several architectures: 1+1, 1:1, 1:n, and m:n. In the context of SDH/SONET, they are referred to as Automatic Protection Switching (APS).

In the 1+1 protection architecture, a protection entity is dedicated to each working entity. The dual-feed mechanism is used whereby the working entity is permanently bridged onto the protection entity at

the source of the protected domain. In normal operation mode, identical traffic is transmitted simultaneously on both the working and protection entities. At the other end (sink) of the protected domain, both feeds are monitored for alarms and maintenance signals. A selection between the working and protection entity is made based on some predetermined criteria, such as the transmission performance requirements or defect indication.

In the 1:1 protection architecture, a protection entity is also dedicated to each working entity. The protected traffic is normally transmitted by the working entity. When the working entity fails, the protected traffic is switched to the protection entity. The two ends of the protected domain must signal detection of the fault and initiate the switchover.

In the 1:n protection architecture, a dedicated protection entity is shared by n working entities. In this case, not all of the affected traffic may be protected.

The m:n architecture is a generalization of the 1:n architecture. Typically m <= n, where m dedicated protection entities are shared by n working entities.

Restoration, also referred to as recovery by rerouting [4], is a survivability technique that establishes new paths or path segments on demand, for restoring affected traffic after the occurrence of a fault. The resources in these alternate paths are the currently unassigned (unreserved) resources in the same layer. Preemption of extra traffic may also be used if spare resources are not available to carry the higher-priority protected traffic. As initiated by detection of a fault on the working path, the selection of a recovery path may be based on preplanned configurations, network routing policies, or current network status such as network topology and fault information. Signaling is used for establishing the new paths to bypass the fault. Thus, restoration involves a path selection process followed by rerouting of the affected traffic from the working entity to the recovery entity.

2.2.4 Survivability Performance

Protection switch time is the time interval from the occurrence of a network fault until the completion of the protection-switching operations. It includes the detection time necessary to initiate the protection switch, any hold-off time to allow for the interworking of protection schemes, and the switch completion time.

Restoration time is the time interval from the occurrence of a network fault to the instant when the affected traffic is either completely restored, or until spare resources are exhausted, and/or no more extra traffic exists that can be preempted to make room.

Restoration priority is a method of giving preference to protect higher-priority traffic ahead of lower-priority traffic. Its use is to help determine the order of restoring traffic after a failure has occurred. The purpose is to differentiate service restoration time as well as to control access to available spare capacity for different classes of traffic.

Preemption priority is a method of determining which traffic can be disconnected in the event that not all traffic with a higher restoration priority is restored after the occurrence of a failure.

2.3 Survivability Mechanisms: Comparison

In a survivable network design, spare capacity and diversity must be built into the network from the beginning to support some degree of self-healing whenever failures occur. A common strategy is to associate each working entity with a protection entity having either dedicated resources or shared resources that are pre-reserved or reserved-on-demand. According to the methods of setting up a protection entity, different approaches to providing survivability can be classified. Generally, protection techniques are based on having a dedicated protection entity set up prior to failure. Such is not the case in restoration techniques, which mainly rely on the use of spare capacity in the network. Hence, in terms of trade-offs, protection techniques usually offer fast recovery from failure with enhanced availability, while restoration techniques usually achieve better resource utilization.

A 1+1 protection architecture is rather expensive since resource duplication is required for the working and protection entities. It is generally used for specific services that need a very high availability.

A 1:1 architecture is inherently slower in recovering from failure than a 1+1 architecture since communication between both ends of the protection domain is required to perform the switch-over operation. An advantage is that the protection entity can optionally be used to carry low-priority extra traffic in normal operation, if traffic preemption is allowed. Packet networks can pre-establish a protection path for later use with pre-planned but not pre-reserved capacity. That is, if no packets are sent onto a protection path,

then no bandwidth is consumed. This is not the case in transmission networks like optical or TDM where path establishment and resource reservation cannot be decoupled.

In the 1:n protection architecture, traffic is normally sent on the working entities. When multiple working entities have failed simultaneously, only one of them can be restored by the common protection entity. This contention could be resolved by assigning a different preemptive priority to each working entity. As in the 1:1 case, the protection entity can optionally be used to carry preemptable traffic in normal operation.

While the m:n architecture can improve system availability with small cost increases, it has rarely been implemented or standardized.

When compared with protection mechanisms, restoration mechanisms are generally more frugal as no resources are committed until after the fault occurs and the location of the fault is known. However, restoration mechanisms are inherently slower, since more must be done following the detection of a fault. Also, the time it takes for the dynamic selection and establishment of alternate paths may vary, depending on the amount of traffic and connections to be restored, and is influenced by the network topology, technology employed, and the type and severity of the fault. As a result, restoration time tends to be more variable than the protection switch time needed with pre-selected protection entities. Hence, in using restoration mechanisms, it is essential to use restoration priority to ensure that service objectives are met cost-effectively.

Once the network routing algorithms have converged after a fault, it may be preferable in some cases, to reoptimize the network by performing a reroute based on the current state of the network and network policies.

3. Survivability

3.1 Scope

Interoperable approaches to network survivability were determined to be an immediate requirement in packet networks as well as in SDH/SONET framed TDM networks. Not as pressing at this time were techniques that would cover all-optical networks (e.g., where framing is unknown), as the control of these networks in a multi-vendor environment appeared to have some other hurdles to first deal with. Also, not of immediate interest were approaches to coordinate or explicitly communicate survivability mechanisms across network layers (such as from a TDM or optical network to/from an IP network). However, a capability should be provided for a network operator to

perform fault notification and to control the operation of survivability mechanisms among different layers. This may require the development of corresponding OAM functionality. However, such issues and those related to OAM are currently outside the scope of this document. (For proposed MPLS OAM requirements, see [8, 9]).

The initial scope is to address only "backhoe failures" in the inter-office connections of a service provider network. A link connection in the router layer is typically comprised of multiple spans in the lower layers. Therefore, the types of network failures that cause a recovery to be performed include link/span failures. However, linecard and node failures may not need to be treated any differently than their respective link/span failures, as a router failure may be represented as a set of simultaneous link failures.

Depending on the actual network configuration, drop-side interface (e.g., between a customer and an access router, or between a router and an optical cross-connect) may be considered either inter-domain or inter-layer. Another inter-domain scenario is the use of intra-office links for interconnecting a metro network and a core network, with both networks being administered by the same service provider. Failures at such interfaces may be similarly protected by the mechanisms of this section.

Other more complex failure mechanisms such as systematic control-plane failure, configuration error, or breach of security are not within the scope of the survivability mechanisms discussed in this document. Network impairment such as congestion that results in lower throughput are also not covered.

3.2 Required initial set of survivability mechanisms

3.2.1 1:1 Path Protection with Pre-Established Capacity

In this protection mode, the head end of a working connection establishes a protection connection to the destination. There should be the ability to maintain relative restoration priorities between working and protection connections, as well as between different classes of protection connections.

In normal operation, traffic is only sent on the working connection, though the ability to signal that traffic will be sent on both connections (1+1 Path for signaling purposes) would be valuable in non-packet networks. Some distinction between working and protection connections is likely, either through explicit objects, or preferably through implicit methods such as general classes or priorities. Head ends need the ability to create connections that are as failure disjoint as possible from each other. This requires SRG information

that can be generally assigned to either nodes or links and propagated through the control or management plane. In this mechanism, capacity in the protection connection is pre-established, however it should be capable of carrying preemptable extra traffic in non-packet networks. When protection capacity is called into service during recovery, there should be the ability to promote the protection connection to working status (for non-revertive mode operation) with some form of make-before-break capability.

3.2.2 1:1 Path Protection with Pre-Planned Capacity

Similar to the above 1:1 protection with pre-established capacity, the protection connection in this case is also pre-signaled. The difference is in the way protection capacity is assigned. With pre-planned capacity, the mechanism supports the ability for the protection capacity to be shared, or "double-booked". Operators need the ability to provision different amounts of protection capacity according to expected failure modes and service level agreements. Thus, an operator may wish to provision sufficient restoration capacity to handle a single failure affecting all connections in an SRG, or may wish to provision less or more restoration capacity. Mechanisms should be provided to allow restoration capacity on each link to be shared by SRG-disjoint failures. In a sense, this is 1:1 from a path perspective; however, the protection capacity in the network (on a link by link basis) is shared in a 1:n fashion, e.g., see the proposals in [10, 11]. If capacity is planned but not allocated, some form of signaling could be required before traffic may be sent on protection connections, especially in TDM networks.

The use of this approach improves network resource utilization, but may require more careful planning. So, initial deployment might be based on 1:1 path protection with pre-established capacity and the local restoration mechanism to be described next.

3.2.3 Local Restoration

Due to the time impact of signal propagation, dynamic recovery of an entire path may not meet the service requirements of some networks. The solution to this is to restore connectivity of the link or span in immediate proximity to the fault, e.g., see the proposals in [12, 13]. At a minimum, this approach should be able to protect against connectivity-type SRGs, though protecting against node-based SRGs might be worthwhile. Also, this approach is applicable to support restoration on the inter-domain and inter-layer interconnection scenarios using intra-office links as described in the Scope Section.

Head end systems must have some control as to whether their connections are candidates for or excluded from local restoration. For example, best-effort and preemptable traffic may be excluded from local restoration; they only get restored if there is bandwidth available. This type of control may require the definition of an object in signaling.

Since local restoration may be suboptimal, a means for head end systems to later perform path-level re-grooming must be supported for this approach.

3.2.4 Path Restoration

In this approach, connections that are impacted by a fault are rerouted by the originating network element upon notification of connection failure. Such a source-based approach is efficient for network resources, but typically takes longer to accomplish restoration. It does not involve any new mechanisms. It merely is a mention of another common approach to protecting against faults in a network.

3.3 Applications Supported

With service continuity under failure as a goal, a network is "survivable" if, in the face of a network failure, connectivity is interrupted for a "brief" period and then recovered before the network failure ends. The length of this interrupted period is dependent upon the application supported. Here are some typical applications and considerations that drive the requirements for an acceptable protection switch time or restoration time:

- Best-effort data: recovery of network connectivity by rerouting at the IP layer would be sufficient
- Premium data service: need to meet TCP timeout or application protocol timer requirements
- Voice: call cutoff is in the range of 140 msec to 2 sec (the time that a person waits after interruption of the speech path before hanging up or the time that a telephone switch will disconnect a call)
- Other real-time service (e.g., streaming, fax) where an interruption would cause the session to terminate
- Mission-critical applications that cannot tolerate even brief interruptions, for example, real-time financial transactions

3.4 Timing Bounds for Survivability Mechanisms

The approach to picking the types of survivability mechanisms recommended was to consider a spectrum of mechanisms that can be used to protect traffic with varying characteristics of survivability and speed of protection/restoration, and then attempt to select a few general points that provide some coverage across that spectrum. The focus of this work is to provide requirements to which a small set of detailed proposals may be developed, allowing the operator some (limited) flexibility in approaches to meeting their design goals in engineering multi-vendor networks. Requirements of different applications as listed in the previous sub-section were discussed generally, however none on the team would likely attest to the scientific merit of the ability of the timing bounds below to meet any specific application's needs. A few assumptions include:

1. Approaches in which protection switch without propagation of information are likely to be faster than those that do require some form of fault notification to some or all elements in a network.

2. Approaches that require some form of signaling after a fault will also likely suffer some timing impact.

Proposed timing bounds for different survivability mechanisms are as follows (all bounds are exclusive of signal propagation):

1:1 path protection with pre-established capacity:	100-500 ms
1:1 path protection with pre-planned capacity:	100-750 ms
Local restoration:	50 ms
Path restoration:	1-5 seconds

To ensure that the service requirements for different applications can be met within the above timing bounds, restoration priority must be implemented to determine the order in which connections are restored (to minimize service restoration time as well as to gain access to available spare capacity on the best paths). For example, mission critical applications may require high restoration priority. At the fiber layer, instead of specific applications, it may be possible that priority be given to certain classifications of customers with their traffic types enclosed within the customer aggregate. Preemption priority should only be used in the event that not all connections can be restored, in which case connections with lower preemption priority should be released. Depending on a service provider's strategy in provisioning network resources for backup, preemption may or may not be needed in the network.

3.5 Coordination Among Layers

A common design goal for networks with multiple technological layers is to provide the desired level of service in the most cost-effective manner. Multilayer survivability may allow the optimization of spare resources through the improvement of resource utilization by sharing spare capacity across different layers, though further investigations are needed. Coordination during recovery among different network layers (e.g., IP, SDH/SONET, optical layer) might necessitate development of vertical hierarchy. The benefits of providing survivability mechanisms at multiple layers, and the optimization of the overall approach, must be weighed with the associated cost and service impacts.

A default coordination mechanism for inter-layer interaction could be the use of nested timers and current SDH/SONET fault monitoring, as has been done traditionally for backward compatibility. Thus, when lower-layer recovery happens in a longer time period than higher-layer recovery, a hold-off timer is utilized to avoid contention between the different single-layer survivability schemes. In other words, multilayer interaction is addressed by having successively higher multiplexing levels operate at a protection/restoration time scale greater than the next lowest layer. This can impact the overall time to recover service. For example, if SDH/SONET protection switching is used, MPLS recovery timers must wait until SDH/SONET has had time to switch. Setting such timers involves a tradeoff between rapid recovery and creation of a race condition where multiple layers are responding to the same fault, potentially allocating resources in an inefficient manner.

In other configurations where the lower layer does not have a restoration capability or is not expected to protect, say an unprotected SDH/SONET linear circuit, then there must be a mechanism for the lower layer to trigger the higher layer to take recovery actions immediately. This difference in network configuration means that implementations must allow for adjustment of hold-off timer values and/or a means for a lower layer to immediately indicate to a higher layer that a fault has occurred so that the higher layer can take restoration or protection actions.

Furthermore, faults at higher layers should not trigger restoration or protection actions at lower layers [3, 4].

It was felt that the current approach to coordination of survivability approaches currently did not have significant operational shortfalls. These approaches include protecting traffic solely at one layer (e.g., at the IP layer over linear WDM, or at the SDH/SONET layer). Where survivability mechanisms might be deployed

at several layers, such as when a routed network rides a SDH/SONET protected network, it was felt that current coordination approaches were sufficient in many cases. One exception is the hold-off of MPLS recovery until the completion of SDH/SONET protection switching as described above. This limits the recovery time of fast MPLS restoration. Also, by design, the operations and mechanisms within a given layer tend to be invisible to other layers.

3.6 Evolution Toward IP Over Optical

As more pressing requirements for survivability and horizontal hierarchy for edge-to-edge signaling are met with technical proposals, it is believed that the benefits of merging (in some manner) the control planes of multiple layers will be outlined. When these benefits are self-evident, it would then seem to be the right time to review whether vertical hierarchy mechanisms are needed, and what the requirements might be. For example, a future requirement might be to provide a better match between the recovery requirements of IP networks with the recovery capability of optical transport. One such proposal is described in [14].

4. Hierarchy Requirements

Efforts in the area of network hierarchy should focus on mechanisms that would allow more scalable edge-to-edge signaling, or signaling across networks with existing network hierarchy (such as multi-area OSPF). This appears to be a more urgent need than mechanisms that might be needed to interconnect networks at different layers.

4.1 Historical Context

One reason for horizontal hierarchy is functionality (e.g., metro versus backbone). Geographic "islands" or partitions reduce the need for interoperability and make administration and operations less complex. Using a simpler, more interoperable, survivability scheme at metro/backbone boundaries is natural for many provider network architectures. In transmission networks, creating geographic islands of different vendor equipment has been done for a long time because multi-vendor interoperability has been difficult to achieve. Traditionally, providers have to coordinate the equipment on either end of a "connection," and making this interoperable reduces complexity. A provider should be able to concatenate survivability mechanisms in order to provide a "protected link" to the next higher level. Think of SDH/SONET rings connecting to TDM DXCs with 1+1 line-layer protection between the ADM and the DXC port. The TDM connection, e.g., a DS3, is protected but usually all equipment on each SDH/SONET ring is from a single vendor. The DXC cross connections are controlled by the provider and the ports are

physically protected resulting in a highly available design. Thus, concatenation of survivability approaches can be used to cascade across a horizontal hierarchy. While not perfect, it is workable in the near to mid-term until multi-vendor interoperability is achieved.

While the problems associated with multi-vendor interoperability may necessitate horizontal hierarchy as a practical matter in the near to mid-term (at least this has been the case in TDM networks), there should not be a technical reason for it in the standards developed by the IETF for core networks, or even most access networks. Establishing interoperability of survivability mechanisms between multi-vendor equipment in core IP networks is urgently required to enable adoption of IP as a viable core transport technology and to facilitate the traffic engineering of future multi-service IP networks [3].

Some of the largest service provider networks currently run a single area/level IGP. Some service providers, as well as many large enterprise networks, run multi-area Open Shortest Path First (OSPF) to gain increases in scalability. Often, this was from an original design, so it is difficult to say if the network truly required the hierarchy to reach its current size.

Some proposals on improved mechanisms to address network hierarchy have been suggested [15, 16, 17, 18, 19]. This document aims to provide the concrete requirements so that these and other proposals can first aim to meet some limited objectives.

4.2 Applications for Horizontal Hierarchy

A primary driver for intra-domain horizontal hierarchy is signaling capabilities in the context of edge-to-edge VPNs, potentially across traffic-engineered data networks. There are a number of different approaches to layer 2 and layer 3 VPNs and they are currently being addressed by different emerging protocols in the provider-provisioned VPNs (e.g., virtual routers) and Pseudo Wire Edge-to-Edge Emulation (PWE3) efforts based on either MPLS and/or IP tunnels. These may or may not need explicit signaling from edge to edge, but it is a common perception that in order to meet SLAs, some form of edge-to-edge signaling may be required.

With a large number of edges (N), scalability is concerned with avoiding the O(N^2) properties of edge-to-edge signaling. However, the main issue here is not with the scalability of large amounts of signaling, such as in O(N^2) meshes with a "connection" between every edge-pair. This is because, even if establishing and maintaining connections is feasible in a large network, there might be an impact on core survivability mechanisms which would cause

protection/restoration times to grow with N^2, which would be undesirable. While some value of N may be inevitable, approaches to reduce N (e.g. to pull in from the edge to aggregation points) might be of value.

Thus, most service providers feel that O(N^2) meshes are not necessary for VPNs, and that the number of tunnels to support VPNs would be within the scalability bounds of current protocols and implementations. That may be the case, as there is currently a lack of ability to signal MPLS tunnels from edge to edge across IGP hierarchy, such as OSPF areas. This may require the development of signaling standards that support dynamic establishment and potentially the restoration of LSPs across a 2-level IGP hierarchy.

For routing scalability, especially in data applications, a major concern is the amount of processing/state that is required in the variety of network elements. If some nodes might not be able to communicate and process the state of every other node, it might be preferable to limit the information. There is one school of thought that says that the amount of information contained by a horizontal barrier should be significant, and that impacts this might have on optimality in route selection and ability to provide global survivability are accepted tradeoffs.

4.3 Horizontal Hierarchy Requirements

Mechanisms are required to allow for edge-to-edge signaling of connections through a network. One network scenario includes medium to large networks that currently have hierarchical interior routing such as multi-area OSPF or multi-level Intermediate System to Intermediate System (IS-IS). The primary context of this is edge-to-edge signaling, which is thought to be required to assure the SLAs for the layer 2 and layer 3 VPNs that are being carried across the network. Another possible context would be edge-to-edge signaling in TDM SDH/SONET networks with IP control, where metro and core networks again might be in a hierarchical interior routing domain.

To support edge-to-edge signaling in the above network scenarios within the framework of existing horizontal hierarchies, current traffic engineering (TE) methods [20, 6] may need to be extended. Requirements for multi-area TE need to be developed to provide guidance for any necessary protocol extensions.

5. Survivability and Hierarchy

When horizontal hierarchy exists in a network technology layer, a question arises as to how survivability can be provided along a connection that crosses hierarchical boundaries.

In designing protocols to meet the requirements of hierarchy, an approach to consider is that boundaries are either clean, or are of minimal value. However, the concept of network elements that participate on both sides of a boundary might be a consideration (e.g., OSPF ABRs). That would allow for devices on either side to take an intra-area approach within their region of knowledge, and for the ABR to do this in both areas, and splice the two protected connections together at a common point (granted it is a common point of failure now). If the limitations of this approach start to appear in operational settings, then perhaps it would be time to start thinking about route-servers and signaling propagated directives. However, one initial approach might be to signal through a common border router, and to consider the service as protected as it consists of a concatenated set of connections which are each protected within their area. Another approach might be to have a least common denominator mechanism at the boundary, e.g., 1+1 port protection. There should also be some standardized means for a survivability scheme on one side of such a boundary to communicate with the scheme on the other side regarding the success or failure of the recovery action. For example, if a part of a "connection" is down on one side of such a boundary, there is no need for the other side to recover from failures.

In summary, at this time, approaches as described above that allow concatenation of survivability schemes across hierarchical boundaries seem sufficient.

6. Security Considerations

The set of SRGs that are defined for a network under a common administrative control and the corresponding assignment of these SRGs to nodes and links within the administrative control is sensitive information and needs to be protected. An SRG is an acknowledgement that nodes and links that belong to an SRG are susceptible to a common threat. An adversary with access to information contained in an SRG could use that information to design an attack, determine the scope of damage caused by the attack and, therefore, be used to maximize the effect of an attack.

The label used to refer to a particular SRG must allow for an encoding such that sensitive information such as physical location, function, purpose, customer, fault type, etc. is not readily discernable by unauthorized users.

SRG information that is propagated through the control and management plane should allow for an encryption mechanism. An example of an approach would be to use IPSEC [21] on all packets carrying SRG information.

7. References

[1] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[3] K. Owens, V. Sharma, and M. Oommen, "Network Survivability Considerations for Traffic Engineered IP Networks", Work in Progress.

[4] V. Sharma, B. Crane, S. Makam, K. Owens, C. Huang, F. Hellstrand, J. Weil, L. Andersson, B. Jamoussi, B. Cain, S. Civanlar, and A. Chiu, "Framework for MPLS-based Recovery", Work in Progress.

[5] M. Thorup, "Fortifying OSPF/ISIS Against Link Failure", http://www.research.att.com/~mthorup/PAPERS/lf_ospf.ps

[6] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I. and X. Xiao, "Overview and Principles of Internet Traffic Engineering", RFC 3272, May 2002.

[7] S. Dharanikota, R. Jain, D. Papadimitriou, R. Hartani, G. Bernstein, V. Sharma, C. Brownmiller, Y. Xue, and J. Strand, "Inter-domain routing with Shared Risk Groups", Work in Progress.

[8] N. Harrison, P. Willis, S. Davari, E. Cuevas, B. Mack-Crane, E. Franze, H. Ohta, T. So, S. Goldfless, and F. Chen, "Requirements for OAM in MPLS Networks," Work in Progress.

[9] D. Allan and M. Azad, "A Framework for MPLS User Plane OAM," Work in Progress.

[10] S. Kini, M. Kodialam, T.V. Lakshman, S. Sengupta, and C. Villamizar, "Shared Backup Label Switched Path Restoration," Work in Progress.

[11] G. Li, C. Kalmanek, J. Yates, G. Bernstein, F. Liaw, and V. Sharma, "RSVP-TE Extensions For Shared-Mesh Restoration in Transport Networks", Work in Progress.

[12] P. Pan (Editor), D.H. Gan, G. Swallow, J. Vasseur, D. Cooper, A. Atlas, and M. Jork, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", Work in Progress.

[13] A. Atlas, C. Villamizar, and C. Litvanyi, "MPLS RSVP-TE Interoperability for Local Protection/Fast Reroute", Work in Progress.

[14] A. Chiu and J. Strand, "Joint IP/Optical Layer Restoration after a Router Failure", Proc. OFC'2001, Anaheim, CA, March 2001.

[15] K. Kompella and Y. Rekhter, "Multi-area MPLS Traffic Engineering", Work in Progress.

[16] G. Ash, et. al., "Requirements for Multi-Area TE", Work in Progress.

[17] A. Iwata, N. Fujita, G.R. Ash, and A. Farrel, "Crankback Routing Extensions for MPLS Signaling", Work in Progress.

[18] C-Y Lee, A. Celer, N. Gammage, S. Ghanti, G. Ash, "Distributed Route Exchangers", Work in Progress.

[19] C-Y Lee and S. Ghanti, "Path Request and Path Reply Message", Work in Progress.

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

[21] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

8. Acknowledgments

A lot of the direction taken in this document, and by the team in its initial effort was steered by the insightful questions provided by Bala Rajagoplan, Greg Bernstein, Yangguang Xu, and Avri Doria. The set of questions is attached as Appendix A in this document.

After the release of the first draft, a number of comments were received. Thanks to the inputs from Jerry Ash, Sudheer Dharanikota, Chuck Kalmanek, Dan Koller, Lyndon Ong, Steve Plote, and Yong Xue.

9. Contributing Authors

Jim Boyle (PDNets), Rob Coltun (Movaz), Tim Griffin (AT&T), Ed Kern, Tom Reddington (Lucent) and Malin Carlzon.

Appendix A: Questions used to help develop requirements

A. Definitions

1. In determining the specific requirements, the design team should precisely define the concepts "survivability", "restoration", "protection", "protection switching", "recovery", "re-routing" etc. and their relations. This would enable the requirements doc to describe precisely which of these will be addressed. In the following, the term "restoration" is used to indicate the broad set of policies and mechanisms used to ensure survivability.

B. Network types and protection modes

1. What is the scope of the requirements with regard to the types of networks covered? Specifically, are the following in scope:

   Restoration of connections in mesh optical networks (opaque or transparent)
   Restoration of connections in hybrid mesh-ring networks
   Restoration of LSPs in MPLS networks (composed of LSRs overlaid on a transport network, e.g., optical)
   Any other types of networks?
   Is commonality of approach, or optimization of approach more important?

2. What are the requirements with regard to the protection modes to be supported in each network type covered? (Examples of protection modes include 1+1, M:N, shared mesh, UPSR, BLSR, newly defined modes such as P-cycles, etc.)

3. What are the requirements on local span (i.e., link by link) protection and end-to-end protection, and the interaction between them? E.g.: what should be the granularity of connections for each type (single connection, bundle of connections, etc).

C. Hierarchy

1. Vertical (between two network layers):
   What are the requirements for the interaction between restoration procedures across two network layers, when these features are offered in both layers? (Example, MPLS network realized over pt-to-pt optical connections.) Under such a case,

   (a) Are there any criteria to choose which layer should provide protection?

(b) If both layers provide survivability features, what are the requirements to coordinate these mechanisms?

(c) How is lack of current functionality of cross-layer coordination currently hampering operations?

(d) Would the benefits be worth additional complexity associated with routing isolation (e.g. VPN, areas), security, address isolation and policy / authentication processes?

2. Horizontal (between two areas or administrative subdivisions within the same network layer):

(a) What are the criteria that trigger the creation of protocol or administrative boundaries pertaining to restoration? (e.g., scalability? multi-vendor interoperability? what are the practical issues?) multi-provider? Should multi-vendor necessitate hierarchical separation?

When such boundaries are defined:

(b) What are the requirements on how protection/restoration is performed end-to-end across such boundaries?

(c) If different restoration mechanisms are implemented on two sides of a boundary, what are the requirements on their interaction?

What is the primary driver of horizontal hierarchy? (select one)
- functionality (e.g. metro -v- backbone)
- routing scalability
- signaling scalability
- current network architecture, trying to layer on TE on top of an already hierarchical network architecture
- routing and signalling

For signalling scalability, is it
- manageability
- processing/state of network
- edge-to-edge N^2 type issue

For routing scalability, is it
- processing/state of network
- are you flat and want to go hierarchical
- or already hierarchical?
- data or TDM application?

D. Policy

1. What are the requirements for policy support during protection/restoration, e.g., restoration priority, preemption, etc.

E. Signaling Mechanisms

1. What are the requirements on the signaling transport mechanism (e.g., in-band over SDH/SONET overhead bytes, out-of-band over an IP network, etc.) used to communicate restoration protocol messages between network elements? What are the bandwidth and other requirements on the signaling channels?

2. What are the requirements on fault detection/localization mechanisms (which is the prelude to performing restoration procedures) in the case of opaque and transparent optical networks? What are the requirements in the case of MPLS restoration?

3. What are the requirements on signaling protocols to be used in restoration procedures (e.g., high priority processing, security, etc)?

4. Are there any requirements on the operation of restoration protocols?

F. Quantitative

1. What are the quantitative requirements (e.g., latency) for completing restoration under different protection modes (for both local and end-to-end protection)?

G. Management

1. What information should be measured/maintained by the control plane at each network element pertaining to restoration events?

2. What are the requirements for the correlation between control plane and data plane failures from the restoration point of view?

Editors' Addresses

Wai Sum Lai
AT&T
200 Laurel Avenue
Middletown, NJ 07748, USA

Phone: +1 732-420-3712
EMail: wlai@att.com

Dave McDysan
WorldCom
22001 Loudoun County Pkwy
Ashburn, VA 20147, USA

EMail: dave.mcdysan@wcom.com

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

Exhibit 4

Network Working Group
Request for Comments: 3469
Category: Informational

V. Sharma, Ed.
Metanoia, Inc.
F. Hellstrand, Ed.
Nortel Networks

February 2003

Framework for Multi-Protocol Label Switching (MPLS)-based Recovery

Status of this Memo

Copyright Notice

Abstract

Multi-protocol label switching (MPLS) integrates the label swapping forwarding paradigm with network layer routing. To deliver reliable service, MPLS requires a set of procedures to provide protection of the traffic carried on different paths. This requires that the label switching routers (LSRs) support fault detection, fault notification, and fault recovery mechanisms, and that MPLS signaling support the configuration of recovery. With these objectives in mind, this document specifies a framework for MPLS based recovery. Restart issues are not included in this framework.

Table of Contents

1. Introduction

This memo describes a framework for MPLS-based recovery. We provide a detailed taxonomy of recovery terminology, and discuss the motivation for, the objectives of, and the requirements for MPLS-based recovery. We outline principles for MPLS-based recovery, and also provide comparison criteria that may serve as a basis for comparing and evaluating different recovery schemes.

At points in the document, we provide some thoughts about the operation or viability of certain recovery objectives. These should be viewed as the opinions of the authors, and not the consolidated views of the IETF. The document is informational and it is expected that a standards track document will be developed in the future to describe a subset of this document as to meet the needs currently specified by the TE WG.

1.1. Background

Network routing deployed today is focused primarily on connectivity, and typically supports only one class of service, the best effort class. Multi-protocol label switching [RFC3031], on the other hand, by integrating forwarding based on label-swapping of a link local label with network layer routing allows flexibility in the delivery of new routing services. MPLS allows for using such media-specific forwarding mechanisms as label swapping. This enables some sophisticated features such as quality-of-service (QoS) and traffic engineering [RFC2702] to be implemented more effectively. An important component of providing QoS, however, is the ability to transport data reliably and efficiently. Although the current routing algorithms are robust and survivable, the amount of time they take to recover from a fault can be significant, in the order of several seconds (for interior gateway protocols (IGPs)) or minutes (for exterior gateway protocols, such as the Border Gateway Protocol (BGP)), causing disruption of service for some applications in the interim. This is unacceptable in situations where the aim is to provide a highly reliable service, with recovery times that are in the order of seconds down to 10's of milliseconds. IP routing may also not be able to provide bandwidth recovery, where the objective is to provide not only an alternative path, but also bandwidth equivalent to that available on the original path. (For some recent work on bandwidth recovery schemes, the reader is referred to [MPLS-BACKUP].) Examples of such applications are Virtual Leased Line services, Stock Exchange data services, voice traffic, video services etc, i.e., every application that gets a disruption in service long enough to not fulfill service agreements or the required level of quality.

MPLS recovery may be motivated by the notion that there are limitations to improving the recovery times of current routing algorithms. Additional improvement can be obtained by augmenting these algorithms with MPLS recovery mechanisms [MPLS-PATH]. Since MPLS is a possible technology of choice in future IP-based transport networks, it is useful that MPLS be able to provide protection and restoration of traffic. MPLS may facilitate the convergence of network functionality on a common control and management plane. Further, a protection priority could be used as a differentiating

mechanism for premium services that require high reliability, such as Virtual Leased Line services, and high priority voice and video traffic. The remainder of this document provides a framework for MPLS based recovery. It is focused at a conceptual level and is meant to address motivation, objectives and requirements. Issues of mechanism, policy, routing plans and characteristics of traffic carried by recovery paths are beyond the scope of this document.

1.2. Motivation for MPLS-Based Recovery

MPLS based protection of traffic (called MPLS-based Recovery) is useful for a number of reasons. The most important is its ability to increase network reliability by enabling a faster response to faults than is possible with traditional Layer 3 (or IP layer) approaches alone while still providing the visibility of the network afforded by Layer 3. Furthermore, a protection mechanism using MPLS could enable IP traffic to be put directly over WDM optical channels and provide a recovery option without an intervening SONET layer or optical protection. This would facilitate the construction of IP-over-WDM networks that request a fast recovery ability (Note that what is meant here is the transport of IP traffic over WDM links, not the Generalized MPLS, or GMPLS, control of a WDM link).

The need for MPLS-based recovery arises because of the following:

I. Layer 3 or IP rerouting may be too slow for a core MPLS network that needs to support recovery times that are smaller than the convergence times of IP routing protocols.

II. Layer 3 or IP rerouting does not provide the ability to provide bandwidth protection to specific flows (e.g., voice over IP, virtual leased line services).

III. Layer 0 (for example, optical layer) or Layer 1 (for example, SONET) mechanisms may be wasteful use of resources.

IV. The granularity at which the lower layers may be able to protect traffic may be too coarse for traffic that is switched using MPLS-based mechanisms.

V. Layer 0 or Layer 1 mechanisms may have no visibility into higher layer operations. Thus, while they may provide, for example, link protection, they cannot easily provide node protection or protection of traffic transported at layer 3. Further, this may prevent the lower layers from providing restoration based on the traffic's needs. For example, fast restoration for traffic that needs it, and slower restoration (with possibly more optimal use of resources) for traffic that does not require fast

restoration. In networks where the latter class of traffic is dominant, providing fast restoration to all classes of traffic may not be cost effective from a service provider's perspective.

VI. MPLS has desirable attributes when applied to the purpose of recovery for connectionless networks. Specifically that an LSP is source routed and a forwarding path for recovery can be "pinned" and is not affected by transient instability in SPF routing brought on by failure scenarios.

VII. Establishing interoperability of protection mechanisms between routers/LSRs from different vendors in IP or MPLS networks is desired to enable recovery mechanisms to work in a multivendor environment, and to enable the transition of certain protected services to an MPLS core.

1.3. Objectives/Goals

The following are some important goals for MPLS-based recovery.

I. MPLS-based recovery mechanisms may be subject to the traffic engineering goal of optimal use of resources.

II. MPLS based recovery mechanisms should aim to facilitate restoration times that are sufficiently fast for the end user application. That is, that better match the end-user's application requirements. In some cases, this may be as short as 10s of milliseconds.

We observe that I and II may be conflicting objectives, and a trade off may exist between them. The optimal choice depends on the end-user application's sensitivity to restoration time and the cost impact of introducing restoration in the network, as well as the end-user application's sensitivity to cost.

III. MPLS-based recovery should aim to maximize network reliability and availability. MPLS-based recovery of traffic should aim to minimize the number of single points of failure in the MPLS protected domain.

IV. MPLS-based recovery should aim to enhance the reliability of the protected traffic while minimally or predictably degrading the traffic carried by the diverted resources.

V. MPLS-based recovery techniques should aim to be applicable for protection of traffic at various granularities. For example, it should be possible to specify MPLS-based recovery for a portion of the traffic on an individual path, for all traffic

on an individual path, or for all traffic on a group of paths. Note that a path is used as a general term and includes the notion of a link, IP route or LSP.

VI. MPLS-based recovery techniques may be applicable for an entire end-to-end path or for segments of an end-to-end path.

VII. MPLS-based recovery mechanisms should aim to take into consideration the recovery actions of lower layers. MPLS-based mechanisms should not trigger lower layer protection switching nor should MPLS-based mechanisms be triggered when lower layer switching has or may imminently occur.

VIII. MPLS-based recovery mechanisms should aim to minimize the loss of data and packet reordering during recovery operations. (The current MPLS specification itself has no explicit requirement on reordering.)

IX. MPLS-based recovery mechanisms should aim to minimize the state overhead incurred for each recovery path maintained.

X. MPLS-based recovery mechanisms should aim to minimize the signaling overhead to setup and maintain recovery paths and to notify failures.

XI. MPLS-based recovery mechanisms should aim to preserve the constraints on traffic after switchover, if desired. That is, if desired, the recovery path should meet the resource requirements of, and achieve the same performance characteristics as, the working path.

We observe that some of the above are conflicting goals, and real deployment will often involve engineering compromises based on a variety of factors such as cost, end-user application requirements, network efficiency, complexity involved, and revenue considerations. Thus, these goals are subject to tradeoffs based on the above considerations.

2. Overview

There are several options for providing protection of traffic. The most generic requirement is the specification of whether recovery should be via Layer 3 (or IP) rerouting or via MPLS protection switching or rerouting actions.

Generally network operators aim to provide the fastest, most stable, and the best protection mechanism that can be provided at a reasonable cost. The higher the levels of protection, the more the

resources consumed. Therefore it is expected that network operators will offer a spectrum of service levels. MPLS-based recovery should give the flexibility to select the recovery mechanism, choose the granularity at which traffic is protected, and to also choose the specific types of traffic that are protected in order to give operators more control over that tradeoff. With MPLS-based recovery, it can be possible to provide different levels of protection for different classes of service, based on their service requirements. For example, using approaches outlined below, a Virtual Leased Line (VLL) service or real-time applications like Voice over IP (VoIP) may be supported using link/node protection together with pre-established, pre-reserved path protection. Best effort traffic, on the other hand, may use path protection that is established on demand or may simply rely on IP re-route or higher layer recovery mechanisms. As another example of their range of application, MPLS-based recovery strategies may be used to protect traffic not originally flowing on label switched paths, such as IP traffic that is normally routed hop-by-hop, as well as traffic forwarded on label switched paths.

2.1. Recovery Models

There are two basic models for path recovery: rerouting and protection switching.

Protection switching and rerouting, as defined below, may be used together. For example, protection switching to a recovery path may be used for rapid restoration of connectivity while rerouting determines a new optimal network configuration, rearranging paths, as needed, at a later time.

2.1.1 Rerouting

Recovery by rerouting is defined as establishing new paths or path segments on demand for restoring traffic after the occurrence of a fault. The new paths may be based upon fault information, network routing policies, pre-defined configurations and network topology information. Thus, upon detecting a fault, paths or path segments to bypass the fault are established using signaling.

Once the network routing algorithms have converged after a fault, it may be preferable, in some cases, to reoptimize the network by performing a reroute based on the current state of the network and network policies. This is discussed further in Section 3.8.

In terms of the principles defined in section 3, reroute recovery employs paths established-on-demand with resources reserved-on-demand.

2.1.2 Protection Switching

Protection switching recovery mechanisms pre-establish a recovery path or path segment, based upon network routing policies, the restoration requirements of the traffic on the working path, and administrative considerations. The recovery path may or may not be link and node disjoint with the working path. However if the recovery path shares sources of failure with the working path, the overall reliability of the construct is degraded. When a fault is detected, the protected traffic is switched over to the recovery path(s) and restored.

In terms of the principles in section 3, protection switching employs pre-established recovery paths, and, if resource reservation is required on the recovery path, pre-reserved resources. The various sub-types of protection switching are detailed in Section 4.4 of this document.

2.2. The Recovery Cycles

There are three defined recovery cycles: the MPLS Recovery Cycle, the MPLS Reversion Cycle and the Dynamic Re-routing Cycle. The first cycle detects a fault and restores traffic onto MPLS-based recovery paths. If the recovery path is non-optimal the cycle may be followed by any of the two latter cycles to achieve an optimized network again. The reversion cycle applies for explicitly routed traffic that does not rely on any dynamic routing protocols to converge. The dynamic re-routing cycle applies for traffic that is forwarded based on hop-by-hop routing.

2.2.1 MPLS Recovery Cycle Model

The MPLS recovery cycle model is illustrated in Figure 1. Definitions and a key to abbreviations follow.

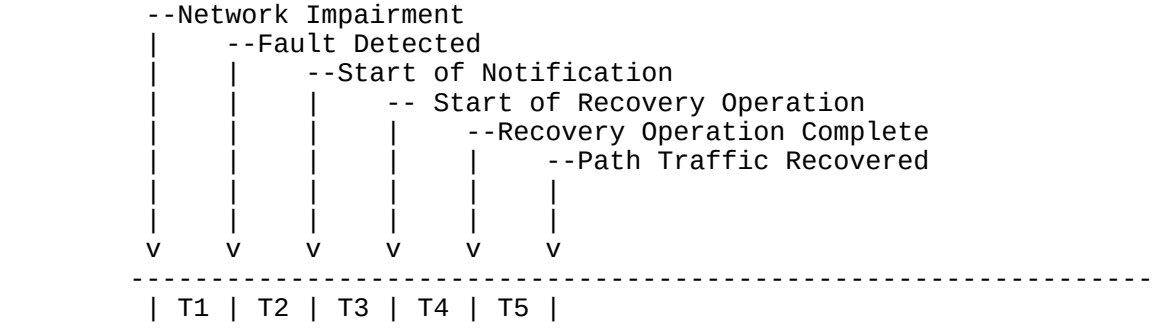


Figure 1. MPLS Recovery Cycle Model

The various timing measures used in the model are described below.

T1 Fault Detection Time
T2 Fault Hold-off Time
T3 Fault Notification Time
T4 Recovery Operation Time
T5 Traffic Recovery Time

Definitions of the recovery cycle times are as follows:

Fault Detection Time

The time between the occurrence of a network impairment and the moment the fault is detected by MPLS-based recovery mechanisms. This time may be highly dependent on lower layer protocols.

Fault Hold-Off Time

The configured waiting time between the detection of a fault and taking MPLS-based recovery action, to allow time for lower layer protection to take effect. The Fault Hold-off Time may be zero.

Note: The Fault Hold-Off Time may occur after the Fault Notification Time interval if the node responsible for the switchover, the Path Switch LSR (PSL), rather than the detecting LSR, is configured to wait.

Fault Notification Time

The time between initiation of a Fault Indication Signal (FIS) by the LSR detecting the fault and the time at which the Path Switch LSR (PSL) begins the recovery operation. This is zero if the PSL detects the fault itself or infers a fault from such events as an adjacency failure.

Note: If the PSL detects the fault itself, there still may be a Fault Hold-Off Time period between detection and the start of the recovery operation.

Recovery Operation Time

The time between the first and last recovery actions. This may include message exchanges between the PSL and PML (Path Merge LSR) to coordinate recovery actions.

Traffic Recovery Time

The time between the last recovery action and the time that the traffic (if present) is completely recovered. This interval is intended to account for the time required for traffic to once again arrive at the point in the network that experienced disrupted or degraded service due to the occurrence of the fault (e.g., the PML). This time may depend on the location of the fault, the recovery mechanism, and the propagation delay along the recovery path.

2.2.2 MPLS Reversion Cycle Model

Protection switching, revertive mode, requires the traffic to be switched back to a preferred path when the fault on that path is cleared. The MPLS reversion cycle model is illustrated in Figure 2. Note that the cycle shown below comes after the recovery cycle shown in Fig. 1.

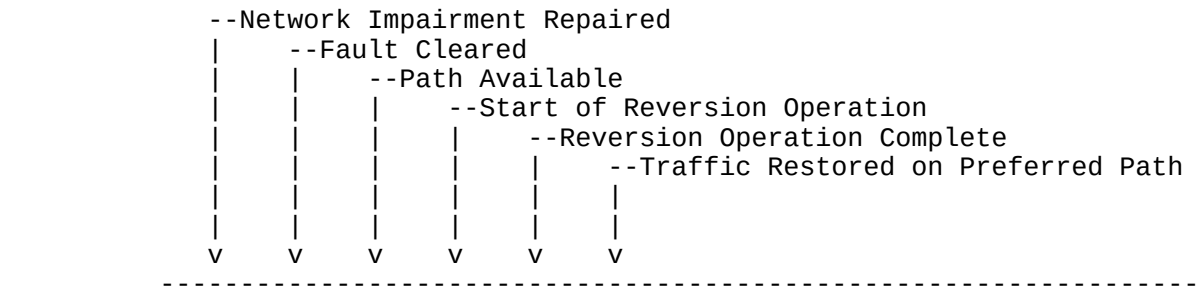


Figure 2. MPLS Reversion Cycle Model

The various timing measures used in the model are described below.

T7 Fault Clearing Time
T8 Clear Hold-Off Time
T9 Clear Notification Time
T10 Reversion Operation Time
T11 Traffic Reversion Time

Note that time T6 (not shown above) is the time for which the network impairment is not repaired and traffic is flowing on the recovery path.

Definitions of the reversion cycle times are as follows:

Fault Clearing Time

The time between the repair of a network impairment and the time that MPLS-based mechanisms learn that the fault has been cleared. This time may be highly dependent on lower layer protocols.

Clear Hold-Off Time

The configured waiting time between the clearing of a fault and MPLS-based recovery action(s). Waiting time may be needed to ensure that the path is stable and to avoid flapping in cases where a fault is intermittent. The Clear Hold-Off Time may be zero.

Note: The Clear Hold-Off Time may occur after the Clear Notification Time interval if the PSL is configured to wait.

Clear Notification Time

The time between initiation of a Fault Recovery Signal (FRS) by the LSR clearing the fault and the time at which the path switch LSR begins the reversion operation. This is zero if the PSL clears the fault itself.

Note: If the PSL clears the fault itself, there still may be a Clear Hold-off Time period between fault clearing and the start of the reversion operation.

Reversion Operation Time

The time between the first and last reversion actions. This may include message exchanges between the PSL and PML to coordinate reversion actions.

Traffic Reversion Time

The time between the last reversion action and the time that traffic (if present) is completely restored on the preferred path. This interval is expected to be quite small since both paths are working and care may be taken to limit the traffic disruption (e.g., using "make before break" techniques and synchronous switch-over).

In practice, the most interesting times in the reversion cycle are the Clear Hold-off Time and the Reversion Operation Time together with Traffic Reversion Time (or some other measure of traffic

disruption). The first interval is to ensure stability of the repaired path and the latter one is to minimize disruption time while the reversion action is in progress.

Given that both paths are available, it is better to wait to have a well-controlled switch-back with minimal disruption than have an immediate operation that may cause new faults to be introduced (except, perhaps, when the recovery path is unable to offer a quality of service comparable to the preferred path).

2.2.3 Dynamic Re-routing Cycle Model

Dynamic rerouting aims to bring the IP network to a stable state after a network impairment has occurred. A re-optimized network is achieved after the routing protocols have converged, and the traffic is moved from a recovery path to a (possibly) new working path. The steps involved in this mode are illustrated in Figure 3.

Note that the cycle shown below may be overlaid on the recovery cycle shown in Fig. 1 or the reversion cycle shown in Fig. 2, or both (in the event that both the recovery cycle and the reversion cycle take place before the routing protocols converge), and occurs if after the convergence of the routing protocols it is determined (based on on-line algorithms or off-line traffic engineering tools, network configuration, or a variety of other possible criteria) that there is a better route for the working path.

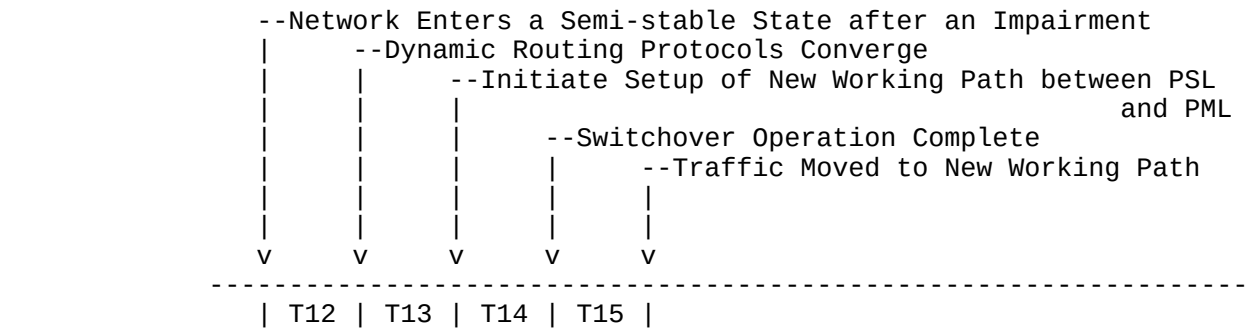


Figure 3. Dynamic Rerouting Cycle Model

The various timing measures used in the model are described below.

T12 Network Route Convergence Time
T13 Hold-down Time (optional)
T14 Switchover Operation Time
T15 Traffic Restoration Time

Network Route Convergence Time

We define the network route convergence time as the time taken for the network routing protocols to converge and for the network to reach a stable state.

Holddown Time

We define the holddown period as a bounded time for which a recovery path must be used. In some scenarios it may be difficult to determine if the working path is stable. In these cases a holddown time may be used to prevent excess flapping of traffic between a working and a recovery path.

Switchover Operation Time

The time between the first and last switchover actions. This may include message exchanges between the PSL and PML to coordinate the switchover actions.

Traffic Restoration Time

The time between the last restoration action and the time that traffic (if present) is completely restored on the new preferred path.

2.2.4 Example Recovery Cycle

As an example of the recovery cycle, we present a sequence of events that occur after a network impairment occurs and when a protection switch is followed by dynamic rerouting.

I. Link or path fault occurs
II. Signaling initiated (FIS) for the detected fault
III. FIS arrives at the PSL
IV. The PSL initiates a protection switch to a pre-configured recovery path
V. The PSL switches over the traffic from the working path to the recovery path
VI. The network enters a semi-stable state
VII. Dynamic routing protocols converge after the fault, and a new working path is calculated (based, for example, on some of the criteria mentioned in Section 2.1.1).
VIII. A new working path is established between the PSL and the PML (assumption is that PSL and PML have not changed)
IX. Traffic is switched over to the new working path.

2.3. Definitions and Terminology

This document assumes the terminology given in [RFC3031], and, in addition, introduces the following new terms.

2.3.1 General Recovery Terminology

Re-routing

A recovery mechanism in which the recovery path or path segments are created dynamically after the detection of a fault on the working path. In other words, a recovery mechanism in which the recovery path is not pre-established.

Protection Switching

A recovery mechanism in which the recovery path or path segments are created prior to the detection of a fault on the working path. In other words, a recovery mechanism in which the recovery path is pre-established.

Working Path

The protected path that carries traffic before the occurrence of a fault. The working path can be of different kinds; a hop-by-hop routed path, a trunk, a link, an LSP or part of a multipoint-to-point LSP.

Synonyms for a working path are primary path and active path.

Recovery Path

The path by which traffic is restored after the occurrence of a fault. In other words, the path on which the traffic is directed by the recovery mechanism. The recovery path is established by MPLS means. The recovery path can either be an equivalent recovery path and ensure no reduction in quality of service, or be a limited recovery path and thereby not guarantee the same quality of service (or some other criteria of performance) as the working path. A limited recovery path is not expected to be used for an extended period of time.

Synonyms for a recovery path are: back-up path, alternative path, and protection path.

Protection Counterpart

The "other" path when discussing pre-planned protection switching schemes. The protection counterpart for the working path is the recovery path and vice-versa.

Path Switch LSR (PSL)

An LSR that is responsible for switching or replicating the traffic between the working path and the recovery path.

Path Merge LSR (PML)

An LSR that is responsible for receiving the recovery path traffic, and either merging the traffic back onto the working path, or, if it is itself the destination, passing the traffic on to the higher layer protocols.

Point of Repair (POR)

An LSR that is setup for performing MPLS recovery. In other words, an LSR that is responsible for effecting the repair of an LSP. The POR, for example, can be a PSL or a PML, depending on the type of recovery scheme employed.

Intermediate LSR

An LSR on a working or recovery path that is neither a PSL nor a PML for that path.

Path Group (PG)

A logical bundling of multiple working paths, each of which is routed identically between a Path Switch LSR and a Path Merge LSR.

Protected Path Group (PPG)

A path group that requires protection.

Protected Traffic Portion (PTP)

The portion of the traffic on an individual path that requires protection. For example, code points in the EXP bits of the shim header may identify a protected portion.

Bypass Tunnel

A path that serves to back up a set of working paths using the label stacking approach [RFC3031]. The working paths and the bypass tunnel must all share the same path switch LSR (PSL) and the path merge LSR (PML).

Switch-Over

The process of switching the traffic from the path that the traffic is flowing on onto one or more alternate path(s). This may involve moving traffic from a working path onto one or more recovery paths, or may involve moving traffic from a recovery path(s) on to a more optimal working path(s).

Switch-Back

The process of returning the traffic from one or more recovery paths back to the working path(s).

Revertive Mode

A recovery mode in which traffic is automatically switched back from the recovery path to the original working path upon the restoration of the working path to a fault-free condition. This assumes a failed working path does not automatically surrender resources to the network.

Non-revertive Mode

A recovery mode in which traffic is not automatically switched back to the original working path after this path is restored to a fault-free condition. (Depending on the configuration, the original working path may, upon moving to a fault-free condition, become the recovery path, or it may be used for new working traffic, and be no longer associated with its original recovery path, i.e., is surrendered to the network.)

MPLS Protection Domain

The set of LSRs over which a working path and its corresponding recovery path are routed.

MPLS Protection Plan

The set of all LSP protection paths and the mapping from working to protection paths deployed in an MPLS protection domain at a given time.

Liveness Message

A message exchanged periodically between two adjacent LSRs that serves as a link probing mechanism. It provides an integrity check of the forward and the backward directions of the link between the two LSRs as well as a check of neighbor aliveness.

Path Continuity Test

A test that verifies the integrity and continuity of a path or path segment. The details of such a test are beyond the scope of this document. (This could be accomplished, for example, by transmitting a control message along the same links and nodes as the data traffic or similarly could be measured by the absence of traffic and by providing feedback.)

2.3.2 Failure Terminology

Path Failure (PF)

Path failure is a fault detected by MPLS-based recovery mechanisms, which is defined as the failure of the liveness message test or a path continuity test, which indicates that path connectivity is lost.

Path Degraded (PD)

Path degraded is a fault detected by MPLS-based recovery mechanisms that indicates that the quality of the path is unacceptable.

Link Failure (LF)

A lower layer fault indicating that link continuity is lost. This may be communicated to the MPLS-based recovery mechanisms by the lower layer.

Link Degraded (LD)

A lower layer indication to MPLS-based recovery mechanisms that the link is performing below an acceptable level.

Fault Indication Signal (FIS)

A signal that indicates that a fault along a path has occurred. It is relayed by each intermediate LSR to its upstream or downstream neighbor, until it reaches an LSR that is setup to perform MPLS recovery (the POR). The FIS is transmitted

periodically by the node/nodes closest to the point of failure, for some configurable length of time or until the transmitting node receives an acknowledgement from its neighbor.

Fault Recovery Signal (FRS)

A signal that indicates a fault along a working path has been repaired. Again, like the FIS, it is relayed by each intermediate LSR to its upstream or downstream neighbor, until is reaches the LSR that performs recovery of the original path. The FRS is transmitted periodically by the node/nodes closest to the point of failure, for some configurable length of time or until the transmitting node receives an acknowledgement from its neighbor.

2.4. Abbreviations

FIS: Fault Indication Signal.
FRS: Fault Recovery Signal.
LD: Link Degraded.
LF: Link Failure.
PD: Path Degraded.
PF: Path Failure.
PML: Path Merge LSR.
PG: Path Group.
POR: Point of Repair.
PPG: Protected Path Group.
PTP: Protected Traffic Portion.
PSL: Path Switch LSR.

3. MPLS-based Recovery Principles

MPLS-based recovery refers to the ability to effect quick and complete restoration of traffic affected by a fault in an MPLS-enabled network. The fault may be detected on the IP layer or in lower layers over which IP traffic is transported. Fastest MPLS recovery is assumed to be achieved with protection switching and may be viewed as the MPLS LSR switch completion time that is comparable to, or equivalent to, the 50 ms switch-over completion time of the SONET layer. Further, MPLS-based recovery may provide bandwidth protection for paths that require it. This section provides a discussion of the concepts and principles of MPLS-based recovery. The concepts are presented in terms of atomic or primitive terms that may be combined to specify recovery approaches. We do not make any assumptions about the underlying layer 1 or layer 2 transport mechanisms or their recovery mechanisms.

3.1. Configuration of Recovery

An LSR may support any or all of the following recovery options on a per-path basis:

Default-recovery (No MPLS-based recovery enabled): Traffic on the working path is recovered only via Layer 3 or IP rerouting or by some lower layer mechanism such as SONET APS. This is equivalent to having no MPLS-based recovery. This option may be used for low priority traffic or for traffic that is recovered in another way (for example load shared traffic on parallel working paths may be automatically recovered upon a fault along one of the working paths by distributing it among the remaining working paths).

Recoverable (MPLS-based recovery enabled): This working path is recovered using one or more recovery paths, either via rerouting or via protection switching.

3.2. Initiation of Path Setup

There are three options for the initiation of the recovery path setup. The active and recovery paths may be established by using either RSVP-TE [RFC2205][RFC3209] or CR-LDP [RFC3212], or by any other means including SNMP.

Pre-established:

This is the same as the protection switching option. Here a recovery path(s) is established prior to any failure on the working path. The path selection can either be determined by an administrative centralized tool, or chosen based on some algorithm implemented at the PSL and possibly intermediate nodes. To guard against the situation when the pre-established recovery path fails before or at the same time as the working path, the recovery path should have secondary configuration options as explained in Section 3.3 below.

Pre-Qualified:

A pre-established path need not be created, it may be pre-qualified. A pre-qualified recovery path is not created expressly for protecting the working path, but instead is a path created for other purposes that is designated as a recovery path after determining that it is an acceptable alternative for carrying the working path traffic. Variants include the case where an optical path or trail is configured, but no switches are set.

Established-on-Demand:

This is the same as the rerouting option. Here, a recovery path is established after a failure on its working path has been detected and notified to the PSL. The recovery path may be pre-computed or computed on demand, which influences recovery times.

3.3. Initiation of Resource Allocation

A recovery path may support the same traffic contract as the working path, or it may not. We will distinguish these two situations by using different additive terms. If the recovery path is capable of replacing the working path without degrading service, it will be called an equivalent recovery path. If the recovery path lacks the resources (or resource reservations) to replace the working path without degrading service, it will be called a limited recovery path. Based on this, there are two options for the initiation of resource allocation:

Pre-reserved:

This option applies only to protection switching. Here a pre-established recovery path reserves required resources on all hops along its route during its establishment. Although the reserved resources (e.g., bandwidth and/or buffers) at each node cannot be used to admit more working paths, they are available to be used by all traffic that is present at the node before a failure occurs. The resources held by a set of recovery paths may be shared if they protect resources that are not simultaneously subject to failure.

Reserved-on-Demand:

This option may apply either to rerouting or to protection switching. Here a recovery path reserves the required resources after a failure on the working path has been detected and notified to the PSL and before the traffic on the working path is switched over to the recovery path.

Note that under both the options above, depending on the amount of resources reserved on the recovery path, it could either be an equivalent recovery path or a limited recovery path.

3.3.1 Subtypes of Protection Switching

The resources (bandwidth, buffers, processing) on the recovery path may be used to carry either a copy of the working path traffic or extra traffic that is displaced when a protection switch occurs. This leads to two subtypes of protection switching.

In 1+1 ("one plus one") protection, the resources (bandwidth, buffers, processing capacity) on the recovery path are fully reserved, and carry the same traffic as the working path. Selection between the traffic on the working and recovery paths is made at the path merge LSR (PML). In effect the PSL function is deprecated to establishment of the working and recovery paths and a simple replication function. The recovery intelligence is delegated to the PML.

In 1:1 ("one for one") protection, the resources (if any) allocated on the recovery path are fully available to preemptible low priority traffic except when the recovery path is in use due to a fault on the working path. In other words, in 1:1 protection, the protected traffic normally travels only on the working path, and is switched to the recovery path only when the working path has a fault. Once the protection switch is initiated, the low priority traffic being carried on the recovery path may be displaced by the protected traffic. This method affords a way to make efficient use of the recovery path resources.

This concept can be extended to 1:n (one for n) and m:n (m for n) protection.

3.4. Scope of Recovery

3.4.1 Topology

3.4.1.1 Local Repair

The intent of local repair is to protect against a link or neighbor node fault and to minimize the amount of time required for failure propagation. In local repair (also known as local recovery), the node immediately upstream of the fault is the one to initiate recovery (either rerouting or protection switching). Local repair can be of two types:

Link Recovery/Restoration

In this case, the recovery path may be configured to route around a certain link deemed to be unreliable. If protection switching is used, several recovery paths may be configured for one working path, depending on the specific faulty link that each protects against.

Alternatively, if rerouting is used, upon the occurrence of a fault on the specified link, each path is rebuilt such that it detours around the faulty link.

In this case, the recovery path need only be disjoint from its working path at a particular link on the working path, and may have overlapping segments with the working path. Traffic on the working path is switched over to an alternate path at the upstream LSR that connects to the failed link. Link recovery is potentially the fastest to perform the switchover, and can be effective in situations where certain path components are much more unreliable than others.

Node Recovery/Restoration

In this case, the recovery path may be configured to route around a neighbor node deemed to be unreliable. Thus the recovery path is disjoint from the working path only at a particular node and at links associated with the working path at that node. Once again, the traffic on the primary path is switched over to the recovery path at the upstream LSR that directly connects to the failed node, and the recovery path shares overlapping portions with the working path.

3.4.1.2 Global Repair

The intent of global repair is to protect against any link or node fault on a path or on a segment of a path, with the obvious exception of the faults occurring at the ingress node of the protected path segment. In global repair, the POR is usually distant from the failure and needs to be notified by a FIS.

In global repair also, end-to-end path recovery/restoration applies. In many cases, the recovery path can be made completely link and node disjoint with its working path. This has the advantage of protecting against all link and node fault(s) on the working path (end-to-end path or path segment).

However, it may, in some cases, be slower than local repair since the fault notification message must now travel to the POR to trigger the recovery action.

3.4.1.3 Alternate Egress Repair

It is possible to restore service without specifically recovering the faulted path.

For example, for best effort IP service it is possible to select a recovery path that has a different egress point from the working path (i.e., there is no PML). The recovery path egress must simply be a router that is acceptable for forwarding the FEC carried by the working path (without creating looping). In an engineering context, specific alternative FEC/LSP mappings with alternate egresses can be formed.

This may simplify enhancing the reliability of implicitly constructed MPLS topologies. A PSL may qualify LSP/FEC bindings as candidate recovery paths as simply link and node disjoint with the immediate downstream LSR of the working path.

3.4.1.4 Multi-Layer Repair

Multi-layer repair broadens the network designer’s tool set for those cases where multiple network layers can be managed together to achieve overall network goals. Specific criteria for determining when multi-layer repair is appropriate are beyond the scope of this document.

3.4.1.5 Concatenated Protection Domains

A given service may cross multiple networks and these may employ different recovery mechanisms. It is possible to concatenate protection domains so that service recovery can be provided end-to-end. It is considered that the recovery mechanisms in different domains may operate autonomously, and that multiple points of attachment may be used between domains (to ensure there is no single point of failure). Alternate egress repair requires management of concatenated domains in that an explicit MPLS point of failure (the PML) is by definition excluded. Details of concatenated protection domains are beyond the scope of this document.

3.4.2 Path Mapping

Path mapping refers to the methods of mapping traffic from a faulty working path on to the recovery path. There are several options for this, as described below. Note that the options below should be viewed as atomic terms that only describe how the working and protection paths are mapped to each other. The issues of resource reservation along these paths, and how switchover is actually performed lead to the more commonly used composite terms, such as 1+1 and 1:1 protection, which were described in Section 4.3.1..

1-to-1 Protection

In 1-to-1 protection the working path has a designated recovery path that is only to be used to recover that specific working path.

n-to-1 Protection

In n-to-1 protection, up to n working paths are protected using only one recovery path. If the intent is to protect against any single fault on any of the working paths, the n working paths should be diversely routed between the same PSL and PML. In some cases, handshaking between PSL and PML may be required to complete the recovery, the details of which are beyond the scope of this document.

n-to-m Protection

In n-to-m protection, up to n working paths are protected using m recovery paths. Once again, if the intent is to protect against any single fault on any of the n working paths, the n working paths and the m recovery paths should be diversely routed between the same PSL and PML. In some cases, handshaking between PSL and PML may be required to complete the recovery, the details of which are beyond the scope of this document. n-to-m protection is for further study.

Split Path Protection

In split path protection, multiple recovery paths are allowed to carry the traffic of a working path based on a certain configurable load splitting ratio. This is especially useful when no single recovery path can be found that can carry the entire traffic of the working path in case of a fault. Split path protection may require handshaking between the PSL and the PML(s), and may require the PML(s) to correlate the traffic arriving on

multiple recovery paths with the working path. Although this is an attractive option, the details of split path protection are beyond the scope of this document.

3.4.3 Bypass Tunnels

It may be convenient, in some cases, to create a "bypass tunnel" for a PPG between a PSL and PML, thereby allowing multiple recovery paths to be transparent to intervening LSRs [RFC2702]. In this case, one LSP (the tunnel) is established between the PSL and PML following an acceptable route and a number of recovery paths can be supported through the tunnel via label stacking. It is not necessary to apply label stacking when using a bypass tunnel. A bypass tunnel can be used with any of the path mapping options discussed in the previous section.

As with recovery paths, the bypass tunnel may or may not have resource reservations sufficient to provide recovery without service degradation. It is possible that the bypass tunnel may have sufficient resources to recover some number of working paths, but not all at the same time. If the number of recovery paths carrying traffic in the tunnel at any given time is restricted, this is similar to the n-to-1 or n-to-m protection cases mentioned in Section 3.4.2.

3.4.4 Recovery Granularity

Another dimension of recovery considers the amount of traffic requiring protection. This may range from a fraction of a path to a bundle of paths.

3.4.4.1 Selective Traffic Recovery

This option allows for the protection of a fraction of traffic within the same path. The portion of the traffic on an individual path that requires protection is called a protected traffic portion (PTP). A single path may carry different classes of traffic, with different protection requirements. The protected portion of this traffic may be identified by its class, as for example, via the EXP bits in the MPLS shim header or via the priority bit in the ATM header.

3.4.4.2 Bundling

Bundling is a technique used to group multiple working paths together in order to recover them simultaneously. The logical bundling of multiple working paths requiring protection, each of which is routed identically between a PSL and a PML, is called a protected path group

(PPG). When a fault occurs on the working path carrying the PPG, the PPG as a whole can be protected either by being switched to a bypass tunnel or by being switched to a recovery path.

3.4.5 Recovery Path Resource Use

In the case of pre-reserved recovery paths, there is the question of what use these resources may be put to when the recovery path is not in use. There are two options:

Dedicated-resource: If the recovery path resources are dedicated, they may not be used for anything except carrying the working traffic. For example, in the case of 1+1 protection, the working traffic is always carried on the recovery path. Even if the recovery path is not always carrying the working traffic, it may not be possible or desirable to allow other traffic to use these resources.

Extra-traffic-allowed: If the recovery path only carries the working traffic when the working path fails, then it is possible to allow extra traffic to use the reserved resources at other times. Extra traffic is, by definition, traffic that can be displaced (without violating service agreements) whenever the recovery path resources are needed for carrying the working path traffic.

Shared-resource: A shared recovery resource is dedicated for use by multiple primary resources that (according to SRLGs) are not expected to fail simultaneously.

3.5. Fault Detection

MPLS recovery is initiated after the detection of either a lower layer fault or a fault at the IP layer or in the operation of MPLS-based mechanisms. We consider four classes of impairments: Path Failure, Path Degraded, Link Failure, and Link Degraded.

Path Failure (PF) is a fault that indicates to an MPLS-based recovery scheme that the connectivity of the path is lost. This may be detected by a path continuity test between the PSL and PML. Some, and perhaps the most common, path failures may be detected using a link probing mechanism between neighbor LSRs. An example of a probing mechanism is a liveness message that is exchanged periodically along the working path between peer LSRs [MPLS-PATH]. For either a link probing mechanism or path continuity test to be effective, the test message must be guaranteed to follow the same route as the working or recovery path, over the segment being tested. In addition, the path continuity test must take the path merge points

into consideration. In the case of a bi-directional link implemented as two unidirectional links, path failure could mean that either one or both unidirectional links are damaged.

Path Degraded (PD) is a fault that indicates to MPLS-based recovery schemes/mechanisms that the path has connectivity, but that the quality of the connection is unacceptable. This may be detected by a path performance monitoring mechanism, or some other mechanism for determining the error rate on the path or some portion of the path. This is local to the LSR and consists of excessive discarding of packets at an interface, either due to label mismatch or due to TTL errors, for example.

Link Failure (LF) is an indication from a lower layer that the link over which the path is carried has failed. If the lower layer supports detection and reporting of this fault (that is, any fault that indicates link failure e.g., SONET LOS (Loss of Signal)), this may be used by the MPLS recovery mechanism. In some cases, using LF indications may provide faster fault detection than using only MPLS-based fault detection mechanisms.

Link Degraded (LD) is an indication from a lower layer that the link over which the path is carried is performing below an acceptable level. If the lower layer supports detection and reporting of this fault, it may be used by the MPLS recovery mechanism. In some cases, using LD indications may provide faster fault detection than using only MPLS-based fault detection mechanisms.

3.6. Fault Notification

MPLS-based recovery relies on rapid and reliable notification of faults. Once a fault is detected, the node that detected the fault must determine if the fault is severe enough to require path recovery. If the node is not capable of initiating direct action (e.g., as a point of repair, POR) the node should send out a notification of the fault by transmitting a FIS to the POR. This can take several forms:

(i) control plane messaging: relayed hop-by-hop along the path upstream of the failed LSP until a POR is reached.
(ii) user plane messaging: sent downstream to the PML, which may take corrective action (as a POR for 1+1) or communicate with a POR upstream (for 1:n) by any of several means:
- control plane messaging
- user plane return path (either through a bi-directional LSP or via other means)

Since the FIS is a control message, it should be transmitted with high priority to ensure that it propagates rapidly towards the affected POR(s). Depending on how fault notification is configured in the LSRs of an MPLS domain, the FIS could be sent either as a Layer 2 or Layer 3 packet [MPLS-PATH]. The use of a Layer 2-based notification requires a Layer 2 path direct to the POR. An example of a FIS could be the liveness message sent by a downstream LSR to its upstream neighbor, with an optional fault notification field set or it can be implicitly denoted by a teardown message. Alternatively, it could be a separate fault notification packet. The intermediate LSR should identify which of its incoming links to propagate the FIS on.

3.7. Switch-Over Operation

3.7.1 Recovery Trigger

The activation of an MPLS protection switch following the detection or notification of a fault requires a trigger mechanism at the PSL. MPLS protection switching may be initiated due to automatic inputs or external commands. The automatic activation of an MPLS protection switch results from a response to a defect or fault conditions detected at the PSL or to fault notifications received at the PSL. It is possible that the fault detection and trigger mechanisms may be combined, as is the case when a PF, PD, LF, or LD is detected at a PSL and triggers a protection switch to the recovery path. In most cases, however, the detection and trigger mechanisms are distinct, involving the detection of fault at some intermediate LSR followed by the propagation of a fault notification to the POR via the FIS, which serves as the protection switch trigger at the POR. MPLS protection switching in response to external commands results when the operator initiates a protection switch by a command to a POR (or alternatively by a configuration command to an intermediate LSR, which transmits the FIS towards the POR).

Note that the PF fault applies to hard failures (fiber cuts, transmitter failures, or LSR fabric failures), as does the LF fault, with the difference that the LF is a lower layer impairment that may be communicated to MPLS-based recovery mechanisms. The PD (or LD) fault, on the other hand, applies to soft defects (excessive errors due to noise on the link, for instance). The PD (or LD) results in a fault declaration only when the percentage of lost packets exceeds a given threshold, which is provisioned and may be set based on the service level agreement(s) in effect between a service provider and a customer.

3.7.2 Recovery Action

After a fault is detected or FIS is received by the POR, the recovery action involves either a rerouting or protection switching operation. In both scenarios, the next hop label forwarding entry for a recovery path is bound to the working path.

3.8. Post Recovery Operation

When traffic is flowing on the recovery path, decisions can be made as to whether to let the traffic remain on the recovery path and consider it as a new working path or to do a switch back to the old or to a new working path. This post recovery operation has two styles, one where the protection counterparts, i.e., the working and recovery path, are fixed or "pinned" to their routes, and one in which the PSL or other network entity with real-time knowledge of failure dynamically performs re-establishment or controlled rearrangement of the paths comprising the protected service.

3.8.1 Fixed Protection Counterparts

For fixed protection counterparts the PSL will be pre-configured with the appropriate behavior to take when the original fixed path is restored to service. The choices are revertive and non-revertive mode. The choice will typically be dependent on relative costs of the working and protection paths, and the tolerance of the service to the effects of switching paths yet again. These protection modes indicate whether or not there is a preferred path for the protected traffic.

3.8.1.1 Revertive Mode

If the working path always is the preferred path, this path will be used whenever it is available. Thus, in the event of a fault on this path, its unused resources will not be reclaimed by the network on failure. Resources here may include assigned labels, links, bandwidth etc. If the working path has a fault, traffic is switched to the recovery path. In the revertive mode of operation, when the preferred path is restored the traffic is automatically switched back to it.

There are a number of implications to pinned working and recovery paths:

- upon failure and after traffic has been moved to the recovery path, the traffic is unprotected until such time as the path defect in the original working path is repaired and that path restored to service.

- upon failure and after traffic has been moved to the recovery path, the resources associated with the original path remain reserved.

3.8.1.2 Non-revertive Mode

In the non-revertive mode of operation, there is no preferred path or it may be desirable to minimize further disruption of the service brought on by a revertive switching operation. A switch-back to the original working path is not desired or not possible since the original path may no longer exist after the occurrence of a fault on that path. If there is a fault on the working path, traffic is switched to the recovery path. When or if the faulty path (the originally working path) is restored, it may become the recovery path (either by configuration, or, if desired, by management actions).

In the non-revertive mode of operation, the working traffic may or may not be restored to a new optimal working path or to the original working path anyway. This is because it might be useful, in some cases, to either: (a) administratively perform a protection switch back to the original working path after gaining further assurances about the integrity of the path, or (b) it may be acceptable to continue operation on the recovery path, or (c) it may be desirable to move the traffic to a new optimal working path that is calculated based on network topology and network policies. Once a new working path has been defined, an associated recovery path may be setup.

3.8.2 Dynamic Protection Counterparts

For dynamic protection counterparts when the traffic is switched over to a recovery path, the association between the original working path and the recovery path may no longer exist, since the original path itself may no longer exist after the fault. Instead, when the network reaches a stable state following routing convergence, the recovery path may be switched over to a different preferred path either optimization based on the new network topology and associated information or based on pre-configured information.

Dynamic protection counterparts assume that upon failure, the PSL or other network entity will establish new working paths if another switch-over will be performed.

3.8.3 Restoration and Notification

MPLS restoration deals with returning the working traffic from the recovery path to the original or a new working path. Restoration is performed by the PSL either upon receiving notification, via FRS, that the working path is repaired, or upon receiving notification that a new working path is established.

For fixed counterparts in revertive mode, an LSR that detected the fault on the working path also detects the restoration of the working path. If the working path had experienced a LF defect, the LSR detects a return to normal operation via the receipt of a liveness message from its peer. If the working path had experienced a LD defect at an LSR interface, the LSR could detect a return to normal operation via the resumption of error-free packet reception on that interface. Alternatively, a lower layer that no longer detects a LF defect may inform the MPLS-based recovery mechanisms at the LSR that the link to its peer LSR is operational. The LSR then transmits FRS to its upstream LSR(s) that were transmitting traffic on the working path. At the point the PSL receives the FRS, it switches the working traffic back to the original working path.

A similar scheme is used for dynamic counterparts where e.g., an update of topology and/or network convergence may trigger installation or setup of new working paths and may send notification to the PSL to perform a switch over.

We note that if there is a way to transmit fault information back along a recovery path towards a PSL and if the recovery path is an equivalent working path, it is possible for the working path and its recovery path to exchange roles once the original working path is repaired following a fault. This is because, in that case, the recovery path effectively becomes the working path, and the restored working path functions as a recovery path for the original recovery path. This is important, since it affords the benefits of non-revertive switch operation outlined in Section 4.8.1, without leaving the recovery path unprotected.

3.8.4 Reverting to Preferred Path (or Controlled Rearrangement)

In the revertive mode, "make before break" restoration switching can be used, which is less disruptive than performing protection switching upon the occurrence of network impairments. This will minimize both packet loss and packet reordering. The controlled rearrangement of paths can also be used to satisfy traffic engineering requirements for load balancing across an MPLS domain.

3.9. Performance

Resource/performance requirements for recovery paths should be specified in terms of the following attributes:

I. Resource Class Attribute:
Equivalent Recovery Class: The recovery path has the same performance guarantees as the working path. In other words, the recovery path meets the same SLAs as the working path.

Limited Recovery Class: The recovery path does not have the same performance guarantees as the working path.

A. Lower Class:
The recovery path has lower resource requirements or less stringent performance requirements than the working path.

B. Best Effort Class:
The recovery path is best effort.

II. Priority Attribute:
The recovery path has a priority attribute just like the working path (i.e., the priority attribute of the associated traffic trunks). It can have the same priority as the working path or lower priority.

III. Preemption Attribute:
The recovery path can have the same preemption attribute as the working path or a lower one.

4. MPLS Recovery Features

The following features are desirable from an operational point of view:

I. It is desirable that MPLS recovery provides an option to identify protection groups (PPGs) and protection portions (PTPs).

II. Each PSL should be capable of performing MPLS recovery upon the detection of the impairments or upon receipt of notifications of impairments.

III. A MPLS recovery method should not preclude manual protection switching commands. This implies that it would be possible under administrative commands to transfer traffic from a working path to a recovery path, or to transfer traffic from a recovery

path to a working path, once the working path becomes operational following a fault.

IV. A PSL may be capable of performing either a switch back to the original working path after the fault is corrected or a switchover to a new working path, upon the discovery or establishment of a more optimal working path.

V. The recovery model should take into consideration path merging at intermediate LSRs. If a fault affects the merged segment, all the paths sharing that merged segment should be able to recover. Similarly, if a fault affects a non-merged segment, only the path that is affected by the fault should be recovered.

5. Comparison Criteria

Possible criteria to use for comparison of MPLS-based recovery schemes are as follows:

Recovery Time

We define recovery time as the time required for a recovery path to be activated (and traffic flowing) after a fault. Recovery Time is the sum of the Fault Detection Time, Hold-off Time, Notification Time, Recovery Operation Time, and the Traffic Restoration Time. In other words, it is the time between a failure of a node or link in the network and the time before a recovery path is installed and the traffic starts flowing on it.

Full Restoration Time

We define full restoration time as the time required for a permanent restoration. This is the time required for traffic to be routed onto links, which are capable of or have been engineered sufficiently to handle traffic in recovery scenarios. Note that this time may or may not be different from the "Recovery Time" depending on whether equivalent or limited recovery paths are used.

Setup vulnerability

The amount of time that a working path or a set of working paths is left unprotected during such tasks as recovery path computation and recovery path setup may be used to compare schemes. The nature of this vulnerability should be taken into account, e.g., End to End schemes correlate the vulnerability with working paths,

Local Repair schemes have a topological correlation that cuts across working paths and Network Plan approaches have a correlation that impacts the entire network.

Backup Capacity

Recovery schemes may require differing amounts of "backup capacity" in the event of a fault. This capacity will be dependent on the traffic characteristics of the network. However, it may also be dependent on the particular protection plan selection algorithms as well as the signaling and re-routing methods.

Additive Latency

Recovery schemes may introduce additive latency for traffic. For example, a recovery path may take many more hops than the working path. This may be dependent on the recovery path selection algorithms.

Quality of Protection

Recovery schemes can be considered to encompass a spectrum of "packet survivability" which may range from "relative" to "absolute". Relative survivability may mean that the packet is on an equal footing with other traffic of, as an example, the same diff-serv code point (DSCP) in contending for the resources of the portion of the network that survives the failure. Absolute survivability may mean that the survivability of the protected traffic has explicit guarantees.

Re-ordering

Recovery schemes may introduce re-ordering of packets. Also the action of putting traffic back on preferred paths might cause packet re-ordering.

State Overhead

As the number of recovery paths in a protection plan grows, the state required to maintain them also grows. Schemes may require differing numbers of paths to maintain certain levels of coverage, etc. The state required may also depend on the particular scheme used for recovery. The state overhead may be a function of several parameters. For example, the number of recovery paths and the number of the protected facilities (links, nodes, or shared link risk groups (SRLGs)).

Loss

Recovery schemes may introduce a certain amount of packet loss during switchover to a recovery path. Schemes that introduce loss during recovery can measure this loss by evaluating recovery times in proportion to the link speed.

In case of link or node failure a certain packet loss is inevitable.

Coverage

Recovery schemes may offer various types of failover coverage. The total coverage may be defined in terms of several metrics:

I. Fault Types: Recovery schemes may account for only link faults or both node and link faults or also degraded service. For example, a scheme may require more recovery paths to take node faults into account.

II. Number of concurrent faults: dependent on the layout of recovery paths in the protection plan, multiple fault scenarios may be able to be restored.

III. Number of recovery paths: for a given fault, there may be one or more recovery paths.

IV. Percentage of coverage: dependent on a scheme and its implementation, a certain percentage of faults may be covered. This may be subdivided into percentage of link faults and percentage of node faults.

V. The number of protected paths may effect how fast the total set of paths affected by a fault could be recovered. The ratio of protection is n/N, where n is the number of protected paths and N is the total number of paths.

6. Security Considerations

The MPLS recovery that is specified herein does not raise any security issues that are not already present in the MPLS architecture.

Confidentiality or encryption of information on the recovery path is outside the scope of this document, but any method designed to do this in other contexts may be used with the methods described in this document.

7. Intellectual Property Considerations

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

8. Acknowledgements

We would like to thank members of the MPLS WG mailing list for their suggestions on the earlier versions of this document. In particular, Bora Akyol, Dave Allan, Dave Danenberg, Sharam Davari, and Neil Harrison whose suggestions and comments were very helpful in revising the document.

The editors would like to give very special thanks to Curtis Villamizar for his careful and extremely thorough reading of the document and for taking the time to provide numerous suggestions, which were very helpful in the last couple of revisions of the document. Thanks are also due to Adrian Farrel for a through reading of the last version of the document, and to Jean-Phillipe Vasseur and Anna Charny for several useful editorial comments and suggestions, and for input on bandwidth recovery.

9. References

9.1 Normative

[RFC3031] Rosen, E., Viswanathan, A. and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M. and J. McManus, "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.

[RFC3212] Jamoussi, B. (Ed.), Andersson, L., Callon, R., Dantu, R., Wu, L., Doolan, P., Worster, T., Feldman, N., Fredette, A., Girish, M., Gray, E., Heinanen, J., Kilty, T. and A. Malis, "Constraint-Based LSP Setup using LDP", RFC 3212, January 2002.

9.2 Informative

[MPLS-BACKUP] Vasseur, J. P., Charny, A., LeFaucheur, F., and Achirica, "MPLS Traffic Engineering Fast reroute: backup tunnel path computation for bandwidth protection", Work in Progress.

[MPLS-PATH] Haung, C., Sharma, V., Owens, K., Makam, V. "Building Reliable MPLS Networks Using a Path Protection Mechanism", IEEE Commun. Mag., Vol. 40, Issue 3, March 2002, pp. 156-162.

[RFC2205] Braden, R., Zhang, L., Berson, S., Herzog, S., "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

10. Contributing Authors

This document was the collective work of several individuals over a period of three years. The text and content of this document was contributed by the editors and the co-authors listed below. (The contact information for the editors appears in Section 11, and is not repeated below.)

Ben Mack-Crane
Tellabs Operations, Inc.
1415 West Diehl Road
Naperville, IL 60563

Phone: (630) 798-6197
EMail: Ben.Mack-Crane@tellabs.com

Srinivas Makam
Eshernet, Inc.
1712 Ada Ct.
Naperville, IL 60540

Phone: (630) 308-3213
EMail: Smakam60540@yahoo.com

Ken Owens
Edward Jones Investments
201 Progress Parkway
St. Louis, MO 63146

Phone: (314) 515-3431
EMail: ken.owens@edwardjones.com

Changcheng Huang
Carleton University
Minto Center, Rm. 3082
1125 Colonial By Drive
Ottawa, Ont. K1S 5B6 Canada

Phone: (613) 520-2600 x2477
EMail: Changcheng.Huang@sce.carleton.ca

Jon Weil

Brad Cain
Storigen Systems
650 Suffolk Street
Lowell, MA 01854

Phone: (978) 323-4454
EMail: bcain@storigen.com

Loa Andersson

EMail: loa@pi.se

Bilel Jamoussi
Nortel Networks
3 Federal Street, BL3-03
Billerica, MA 01821, USA

Phone:(978) 288-4506
EMail: jamoussi@nortelnetworks.com

Angela Chiu
AT&T Labs-Research
200 Laurel Ave. Rm A5-1F13
Middletown , NJ 07748

Phone: (732) 420-9061
EMail: chiu@research.att.com

Seyhan Civanlar
Lemur Networks, Inc.
135 West 20th Street, 5th Floor
New York, NY 10011

Phone: (212) 367-7676
EMail: scivanlar@lemurnetworks.com

11. Editors' Addresses

Vishal Sharma (Editor)
Metanoia, Inc.
1600 Villa Street, Unit 352
Mountain View, CA 94041-1174

Phone: (650) 386-6723
EMail: v.sharma@ieee.org

Fiffi Hellstrand (Editor)
Nortel Networks
St Eriksgatan 115
PO Box 6701
113 85 Stockholm, Sweden

Phone: +46 8 5088 3687
EMail: fiffi@nortelnetworks.com

12. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

Exhibit 5

Network Working Group
Request for Comments: 3916
Category: Informational

X. Xiao, Ed.
Riverstone Networks
D. McPherson, Ed.
Arbor Networks
P. Pate, Ed.
Overture Networks

September 2004

Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)

Status of this Memo

Copyright Notice

Abstract

This document describes base requirements for the Pseudo-Wire Emulation Edge to Edge Working Group (PWE3 WG). It provides guidelines for other working group documents that will define mechanisms for providing pseudo-wire emulation of Ethernet, ATM, and Frame Relay. Requirements for pseudo-wire emulation of TDM (i.e., "synchronous bit streams at rates defined by ITU G.702") are defined in another document. It should be noted that the PWE3 WG standardizes mechanisms that can be used to provide PWE3 services, but not the services themselves.

Table of Contents

1. Introduction

1.1. What Are Pseudo Wires?

Pseudo Wire Emulation Edge-to-Edge (PWE3) is a mechanism that emulates the essential attributes of a service such as ATM, Frame Relay or Ethernet over a Packet Switched Network (PSN). The required functions of PWs include encapsulating service-specific PDUs arriving at an ingress port, and carrying them across a path or tunnel, managing their timing and order, and any other operations required to emulate the behavior and characteristics of the service as faithfully as possible.

From the customer perspective, the PW is perceived as an unshared link or circuit of the chosen service. However, there may be deficiencies that impede some applications from being carried on a PW. These limitations should be fully described in the appropriate service-specific documents and Applicability Statements.

1.2. Current Network Architecture

The following sections give some background on where networks are today and why they are changing. It also talks about the motivation to provide converged networks while continuing to support existing services. Finally, it discusses how PWs can be a solution for this dilemma.

1.2.1. Multiple Networks

For any given service provider delivering multiple services, the current infrastructure usually consists of parallel or "overlay" networks. Each of these networks implements a specific service, such as Frame Relay, Internet access, etc. This is expensive, both in terms of capital expense and operational costs. Furthermore, the presence of multiple networks complicates planning. Service providers wind up asking themselves these questions:

- Which of my networks do I build out?
- How many fibers do I need for each network?
- How do I efficiently manage multiple networks?

A converged network helps service providers answer these questions in a consistent and economical fashion.

1.2.2. Transition to a Packet-Optimized Converged Network

In order to maximize return on their assets and minimize their operating costs, service providers often look to consolidate the delivery of multiple service types onto a single networking technology.

As packet traffic takes up a larger and larger portion of the available network bandwidth, it becomes increasingly useful to optimize public networks for the Internet Protocol. However, many service providers are confronting several obstacles in engineering packet-optimized networks. Although Internet traffic is the fastest growing traffic segment, it does not generate the highest revenue per bit. For example, Frame Relay traffic currently generates higher revenue per bit than native IP services do. Private line TDM services still generate even more revenue per bit than does Frame Relay. In addition, there is a tremendous amount of legacy equipment deployed within public networks that does not communicate using the Internet Protocol. Service providers continue to utilize non-IP equipment to deploy a variety of services, and see a need to interconnect this legacy equipment over their IP-optimized core networks.

1.3. PWE3 as a Path to Convergence

How do service providers realize the capital and operational benefits of a new packet-based infrastructure, while leveraging the existing equipment and also protecting the large revenue stream associated with this equipment? How do they move from mature Frame Relay or ATM networks, while still being able to provide these lucrative services?

One possibility is the emulation of circuits or services via PWs. Circuit emulation over ATM and interworking of Frame Relay and ATM have already been standardized. Emulation allows existing services to be carried across the new infrastructure, and thus enables the interworking of disparate networks.

Implemented correctly, PWE3 can provide a means for supporting today's services over a new network.

1.4. Suitable Applications for PWE3

What makes an application suitable (or not) for PWE3 emulation? When considering PWs as a means of providing an application, the following questions must be considered:

- Is the application sufficiently deployed to warrant emulation?
- Is there interest on the part of service providers in providing an emulation for the given application?
- Is there interest on the part of equipment manufacturers in providing products for the emulation of a given application?
- Are the complexities and limitations of providing an emulation worth the savings in capital and operational expenses?

If the answer to all four questions is "yes", then the application is likely to be a good candidate for PWE3. Otherwise, there may not be sufficient overlap between the customers, service providers, equipment manufacturers and technology to warrant providing such an emulation.

1.5. Summary

To maximize the return on their assets and minimize their operational costs, many service providers are looking to consolidate the delivery of multiple service offerings and traffic types onto a single IP-optimized network.

In order to create this next-generation converged network, standard methods must be developed to emulate existing telecommunications

formats such as Ethernet, Frame Relay, and ATM over IP-optimized core networks. This document describes requirements for accomplishing this goal.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALLNOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

Some terms used throughout this document are listed below.

Attachment Circuit (AC)
: The physical or virtual circuit attaching a CE to a PE. An AC can be a Frame Relay DLCI, an ATM VPI/VCI, an Ethernet port, a VLAN, a HDLC link, a PPP connection on a physical interface, a PPP session from an L2TP tunnel, an MPLS LSP, etc.

Customer Edge (CE)
: A device where one end of a service originates and/or terminates. The CE is not aware that it is using an emulated service rather than a native service.

Packet Switched Network (PSN)
: Within the context of PWE3, this is a network using IP or MPLS as the mechanism for packet forwarding.

Provider Edge (PE)
: A device that provides PWE3 to a CE.

Pseudo Wire (PW)
: A mechanism that carries the essential elements of an emulated circuit from one PE to another PE over a PSN.

Pseudo Wire Emulation Edge to Edge (PWE3)
: A mechanism that emulates the essential attributes of a service (such as a T1 leased line or Frame Relay) over a PSN.

Pseudo Wire PDU
: A Protocol Data Unit (PDU) sent on the PW that contains all of the data and control information necessary to emulate the desired service.

PSN Tunnel
: A tunnel across a PSN inside which one or more PWs can be carried.

3. Reference Model of PWE3

A pseudo-wire (PW) is a connection between two provider edge (PE) devices which connects two attachment circuits (ACs). An AC can be a Frame Relay DLCI, an ATM VPI/VCI, an Ethernet port, a VLAN, a HDLC link, a PPP connection on a physical interface, a PPP session from an L2TP tunnel, an MPLS LSP, etc.

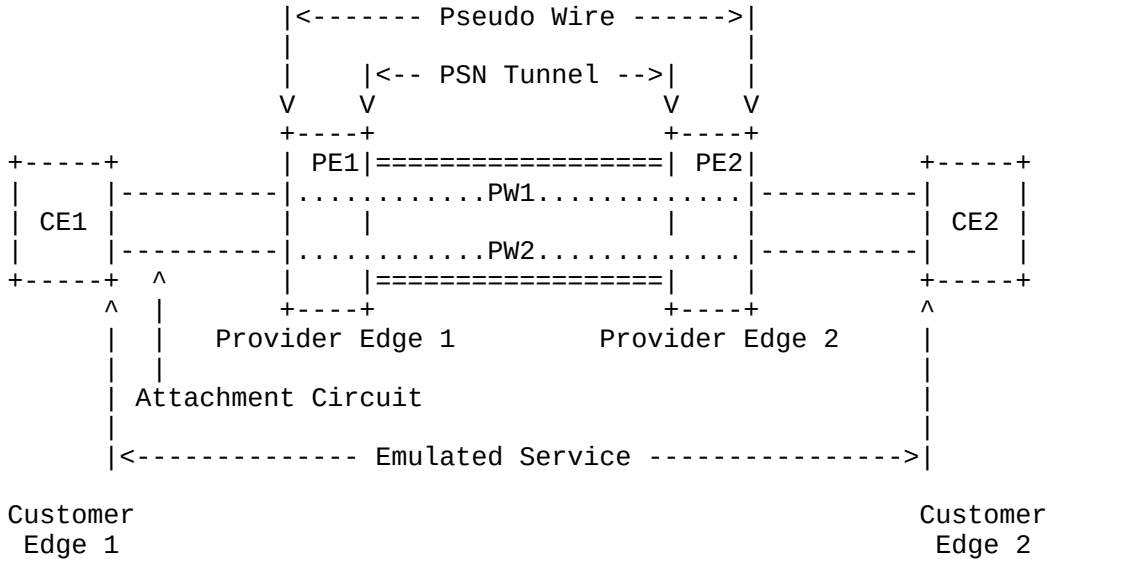


Figure 1: PWE3 Reference Model

During the setup of a PW, the two PEs will be configured or will automatically exchange information about the service to be emulated so that later they know how to process packets coming from the other end. After a PW is set up between two PEs, frames received by one PE from an AC are encapsulated and sent over the PW to the remote PE, where native frames are re-constructed and forwarded to the other CE. For a detailed PWE3 architecture overview, readers should refer to the PWE3 architecture document [PWE3_ARCH].

This document does not assume that a particular type of PWs (e.g., [L2TPv3] sessions or [MPLS] LSPs) or PSNs (e.g., IP or MPLS) is used. Instead, it describes generic requirements that apply to all PWs and PSNs, for all services including Ethernet, ATM, and Frame Relay, etc.

4. Packet Processing

This section describes data plane requirements for PWE3.

4.1. Encapsulation

Every PE MUST provide an encapsulation mechanism for PDUs from an AC. It should be noted that the PDUs to be encapsulated may or may not contain L2 header information. This is service specific. Every PWE3 service MUST specify what the PDU is.

A PW header consists of all the header fields in a PW PDU that are used by the PW egress to determine how to process the PDU. The PSN tunnel header is not considered as part of the PW header.

Specific requirements on PDU encapsulation are listed below.

4.1.1. Conveyance of Necessary L2 Header Information

The egress of a PW needs some information, e.g., which native service the PW PDUs belong to, and possibly some L2 header information, in order to know how to process the PDUs received. A PWE3 encapsulation approach MUST provide some mechanism for conveying such information from the PW ingress to the egress. It should be noted that not all such information must be carried in the PW header of the PW PDUs. Some information (e.g., service type of a PW) can be stored as state information at the egress during PW setup.

4.1.2. Support of Variable Length PDUs

A PWE3 approach MUST accommodate variable length PDUs, if variable length PDUs are allowed by the native service. For example, a PWE3 approach for Frame Relay MUST accommodate variable length frames.

4.1.3. Support of Multiplexing and Demultiplexing

If a service in its native form is capable of grouping multiple circuits into a "trunk", e.g., multiple ATM VCCs in a VPC or multiple Ethernet 802.1Q interfaces in a port, some mechanism SHOULD be provided so that a single PW can be used to connect two end-trunks. From encapsulation perspective, sufficient information MUST be carried so that the egress of the PW can demultiplex individual circuits from the PW.

4.1.4. Validation of PW-PDU

Most L2 frames have a checksum field to assure frame integrity. Every PWE3 service MUST specify whether the frame's checksum should be preserved across the PW, or should be removed at the ingress PE and then be re-calculated and inserted at the egress PE. For protocols such as ATM and FR, the checksum covers link-local information such as the circuit identifiers (e.g., FR DLCI or ATM VPI/VCI). Therefore, such checksum MUST be removed at the ingress PE and recalculated at the egress PE.

4.1.5. Conveyance of Payload Type Information

Under some circumstances, it is desirable to be able to distinguish PW traffic from other types of traffic such as IPv4 or IPv6 or OAM. For example, if Equal Cost Multi-Path (ECMP) is employed in a PSN, this additional distinguishability can be used to reduce the chance that PW packets get misordered by the load balancing mechanism. Some mechanism SHOULD provide this distinguishability if needed. Such mechanism MAY be defined in the PWE3 WG or other WGs.

4.2. Frame Ordering

When packets carrying the PW PDUs traverse a PW, they may arrive at the egress out of order. For some services, the frames (either control frames only or both control and data frames) must be delivered in order. For such services, some mechanism MUST be provided for ensuring in-order delivery. Providing a sequence number in the PW header for each packet is one possible approach to detect out-of-order frames. Mechanisms for re-ordering frames may be provided by Native Service Processing (NSP) [PWE3_ARCH] but are out of scope of PWE3.

4.3. Frame Duplication

In rare cases, packets traversing a PW may be duplicated. For some services, frame duplication is not allowed. For such services some mechanism MUST be provided to ensure that duplicated frames will not be delivered. The mechanism may or may not be the same as the mechanism used to ensure in-order frame delivery.

4.4. Fragmentation

If the combined size of the L2 payload and its associated PWE3 and PSN headers exceeds the PSN path MTU, the L2 payload may need to be fragmented (Alternatively the L2 frame may be dropped). For certain native service, fragmentation may also be needed to maintain a control frame's relative position to the data frames (e.g., an ATM PM

cell's relative position). In general, fragmentation has a performance impact. It is therefore desirable to avoid fragmentation if possible. However, for different services, the need for fragmentation can be different. When there is potential need for fragmentation, each service-specific PWE3 document MUST specify whether to fragment the frame in question or to drop it. If an emulated service chooses to drop the frame, the consequence MUST be specified in its applicability statement.

4.5. Consideration of Per-PSN Packet Overhead

When the L2 PDU size is small, in order to reduce PSN tunnel header overhead, multiple PDUs MAY be concatenated before a PSN tunnel header is added. Each encapsulated PDU still carries its own PW header so that the egress PE knows how to process it. However, the benefit of concatenating multiple PDUs for header efficiency should be weighed against the resulting increase in delay, jitter and the larger penalty incurred by packet loss.

5. Maintenance of Emulated Services

This section describes maintenance requirements for PWE3.

5.1. Setup and Teardown of Pseudo-Wires

A PW must be set up before an emulated circuit can be established, and must be torn down when an emulated circuit is no longer needed. Setup and teardown of a PW can be triggered by a command from the management plane of a PE, or by Setup/Teardown of an AC (e.g., an ATM SVC), or by an auto-discovery mechanism.

Every PWE3 approach MUST define some setup mechanism for establishing the PWs. During the setup process, the PEs need to exchange some information (e.g., to learn each other's capability). The setup mechanism MUST enable the PEs to exchange all necessary information. For example, both endpoints must agree on methods for encapsulating PDUs and handling frame ordering. Which signaling protocol to use and what information to exchange are service specific. Every PWE3 approach MUST specify them. Manual configuration of PWs can be considered as a special kind of signaling and is allowed.

If a native circuit is bi-directional, the corresponding emulated circuit can be signaled "Up" only when the associated PW and PSN tunnels in both directions are functional.

5.2. Handling Maintenance Message of the Native Services

Some native services have mechanisms for maintenance purpose, e.g., ATM OAM and FR LMI. Such maintenance messages can be in-band (i.e., mixed with data messages in the same AC) or out-of-band (i.e., sent in a dedicated control circuit). For such services, all in-band maintenance messages related to a circuit SHOULD be transported in-band just like data messages through the corresponding PW to the remote CE. In other words, no translation is needed at the PEs for in-band maintenance messages. In addition, it MAY be desirable to provide higher reliability for maintenance messages. The mechanisms for providing high reliability do not have to be defined in the PWE3 WG.

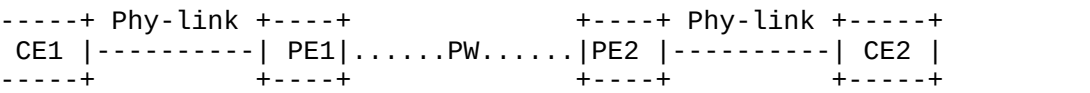
Out-of-band maintenance messages between a CE and a PE may relate to multiple ACs between the CE and the PE. They need to be processed at the local PE and possibly at the remote PE as well. If a native service has some out-of-band maintenance messages, the corresponding emulated service MUST specify how to process such messages at the PEs. In general, an out-of-band maintenance message is either translated into an in-band maintenance message of the native service or a PWE-specific maintenance message for every AC related to that out-of-band message. As an example, assume the ACs between a CE and a PE are some ATM VCCs inside a VPC. When a F4 AIS [UNI3.0] from the CE is received by the PE, the PE should translate that F4 AIS into a F5 AIS and send it to the remote CE for every VCC. Alternatively, the PE should generate a PWE-specific maintenance message (e.g., label withdrawal) to the remote PE for every VCC. When the remote PE receives such a PWE-specific maintenance message, it may need to generate a maintenance message of the native service and send it to the attached CE.

5.3. PE-initiated Maintenance Messages

A PE needs to initiate some maintenance messages under some circumstances without being triggered by any native maintenance messages from the CE. These circumstances are usually caused by fault, e.g., a PW failure in the PSN or a link failure between the CE and the PE.

The reason the PEs need to initiate some maintenance messages under a fault condition is because the existence of a PW between two CEs would otherwise reduce the CEs' maintenance capability. This is illustrated in the following example. If two CEs are directly connected by a physical wire, a native service (e.g., ATM) can use notifications from the lower layer (e.g., the physical link layer) to

assist its maintenance. For example, an ATM PVC can be signaled "Down" if the physical wire fails. However, consider the following scenario.



If the PW between PE1 and PE2 fails, CE1 and CE2 will not receive physical link failure notification. As a result, they cannot declare failure of the emulated circuit in a timely fashion, which will in turn affect higher layer applications. Therefore, when the PW fails, PE1 and PE2 need to initiate some maintenance messages to notify the client layer on CE1 and CE2 that use the PW as a server layer. (In this case, the client layer is the emulated service). Similarly, if the physical link between PE1-CE1 fails, PE1 needs to initiate some maintenance message(s) so that the client layer at CE2 will be notified. PE2 may need to be involved in this process.

In the rare case when a physical wire between two CEs incurs many bit errors, the physical link can be declared "Down" and the client layer at the CEs be notified. Similarly, a PW can incur packet loss, corruption, and out-of-order delivery. These can be considered as "generalized bit error". Upon detection of excessive "generalized bit error", a PW can be declared "Down" and the detecting PE needs to initiate a maintenance message so that the client layer at the CE is notified.

In general, every emulated service MUST specify:

* Under what circumstances PE-initiated maintenance messages are needed,
* Format of the maintenance messages, and
* How to process the maintenance messages at the remote PE.

Some monitoring mechanisms are needed for detecting such circumstances, e.g., a PW failure. Such mechanisms can be defined in the PWE3 WG or elsewhere.

Status of a group of emulated circuits may be affected identically by a single network incidence. For example, when the physical link between a CE and a PE fails, all the emulated circuits that go through that link will fail. It is desirable that a single maintenance message be used to notify failure of the whole group of emulated circuits connected to the same remote PE. A PWE3 approach MAY provide some mechanism for notifying status changes of a group of emulated circuits. One possible approach is to associate each

emulated circuit with a group ID while setting up the PW for that emulated circuit. In a maintenance message, that group ID can be used to refer to all the emulated circuits in that group.

If a PE needs to generate and send a maintenance message to a CE, the PE MUST use a maintenance message of the native service. This is essential in keeping the emulated service transparent to the CEs.

The requirements stated in this section are aligned with the ITU-T maintenance philosophy for telecommunications networks [G805] (i.e., client layer/server layer concept).

6. Management of Emulated Services

Each PWE3 approach SHOULD provide some mechanisms for network operators to manage the emulated service. These mechanisms can be in the forms described below.

6.1. MIBs

SNMP MIBs [SMIV2] MUST be provided for managing each emulated circuit as well as pseudo-wire in general. These MIBs SHOULD be created with the following requirements.

6.2. General MIB Requirements

New MIBs MUST augment or extend where appropriate, existing tables as defined in other existing service-specific MIBs for existing services such as MPLS or L2TP. For example, the ifTable as defined in the Interface MIB [IFMIB] MUST be augmented to provide counts of out-of-order packets. A second example is the extension of the MPLS-TE-MIB [TEMIB] when emulating circuit services over MPLS. Rather than redefining the tunnelTable so that PWE can utilize MPLS tunnels, for example, entries in this table MUST instead be extended to add additional PWE-specific objects. A final example might be to extend the IP Tunnel MIB [IPTUNMIB] in such a way as to provide PWE3-specific semantics when tunnels other than MPLS are used as PSN transport. Doing so facilitates a natural extension of those objects defined in the existing MIBs in terms of management, as well as leveraging existing agent implementations.

An AC MUST appear as an interface in the ifTable.

6.3. Configuration and Provisioning

MIB Tables MUST be designed to facilitate configuration and provisioning of the AC.

The MIB(s) MUST facilitate intra-PSN configuration and monitoring of ACs.

6.4. Performance Monitoring

MIBs MUST collect statistics for performance and fault management.

MIBs MUST provide a description of how existing counters are used for PW emulation and SHOULD not replicate existing MIB counters.

6.5. Fault Management and Notifications

Notifications SHOULD be defined where appropriate to notify the network operators of any interesting situations, including faults detected in the AC.

Objects defined to augment existing protocol-specific notifications in order to add PWE functionality MUST explain how these notifications are to be emitted.

6.6. Pseudo-Wire Connection Verification and Traceroute

For network management purpose, a connection verification mechanism SHOULD be supported by PWs. Connection verification as well as other alarming mechanisms can alert network operators that a PW has lost its remote connection. It is sometimes desirable to know the exact functional path of a PW for troubleshooting purpose, thus a traceroute function capable of reporting the path taken by data packets over the PW SHOULD be provided.

7. Faithfulness of Emulated Services

An emulated service SHOULD be as similar to the native service as possible, but NOT REQUIRED to be identical. The applicability statement of a PWE3 service MUST report limitations of the emulated service.

Some basic requirements on faithfulness of an emulated service are described below.

7.1. Characteristics of an Emulated Service

From the perspective of a CE, an emulated circuit is characterized as an unshared link or circuit of the chosen service, although service quality of the emulated service may be different from that of a native one. Specifically, the following requirements MUST be met:

1) It MUST be possible to define type (e.g., Ethernet, which is inherited from the native service), speed (e.g., 100Mbps), and MTU size for an emulated circuit, if it is possible to do so for a native circuit.

2) If the two endpoints CE1 and CE2 of emulated circuit #1 are connected to PE1 and PE2, respectively, and CE3 and CE4 of emulated circuit #2 are also connected to PE1 and PE2, then the PWs of these two emulated circuits may share the same physical paths between PE1 and PE2. But from each CE's perspective, its emulated circuit MUST appear as unshared. For example, CE1/CE2 MUST NOT be aware of existence of emulated circuit #2 or CE3/CE4.

3) If an emulated circuit fails (either at one of the ACs or in the middle of the PW), both CEs MUST be notified in a timely manner, if they will be notified in the native service (see Section 5.3 for more information). The definition of "timeliness" is service-dependent.

4) If a routing protocol (e.g., IGP) adjacency can be established over a native circuit, it MUST be possible to be established over an emulated circuit as well.

7.2. Service Quality of Emulated Services

It is NOT REQUIRED that an emulated service provide the same service quality as the native service. The PWE3 WG only defines mechanisms for providing PW emulation, not the services themselves. What quality to provide for a specific emulated service is a matter between a service provider (SP) and its customers, and is outside scope of the PWE3 WG.

8. Non-Requirements

Some non-requirements are mentioned in various sections of this document. Those work items are outside scope of the PWE3 WG. They are summarized below:

- Service interworking;

  In Service Interworking, the IWF (Interworking Function) between two dissimilar protocols (e.g., ATM & MPLS, Frame Relay & ATM, ATM & IP, ATM & L2TP, etc.) terminates the protocol used in one network and translates (i.e., maps) its Protocol Control Information (PCI) to the PCI of the protocol used in other network for User, Control and Management Plane functions to the extent possible.

- Selection of a particular type of PWs;

- To make the emulated services perfectly match their native services;

- Defining mechanisms for signaling the PSN tunnels;

- Defining how to perform traffic management on packets that carry PW PDUs;

- Providing any multicast service that is not native to the emulated medium.

  To illustrate this point, Ethernet transmission to a multicast IEEE-48 address is considered in scope, while multicast services like [MARS] that are implemented on top of the medium are out of scope;

9. Quality of Service (QoS) Considerations

Some native services such as ATM can offer higher service quality than best effort Internet service. QoS is therefore essential for ensuring that emulated services are compatible (but not necessarily identical) to their native forms. It is up to network operators to decide how to provide QoS - They can choose to rely on over-provisioning and/or deploy some QoS mechanisms.

In order to take advantage of QoS mechanisms defined in other working groups, e.g., the traffic management schemes defined in DiffServ WG, it is desirable that some mechanisms exists for differentiating the packets resulted from PDU encapsulation. These mechanisms do not have to be defined in the PWE3 approaches themselves. For example, if the resulted packets are MPLS or IP packets, their EXP or DSCP field can be used for marking and differentiating. A PWE3 approach MAY provide guidelines for marking and differentiating.

The applicability of PWE3 to a particular service depends on the sensitivity of that service (or the CE implementation) to delay/jitter etc and the ability of the application layer to mask them. PWE3 may not be applicable to services that have severe constraints in this respect.

10. Inter-domain Issues

PWE is a matter between the PW end-points and is transparent to the network devices between the PW end-points. Therefore, inter-domain PWE is fundamentally similar to intra-domain PWE. As long as PW end-points use the same PWE approach, they can communicate effectively, regardless of whether they are in the same domain. Security may become more important in the inter-domain case and some security measure such as end-point authentication MAY be applied. QoS may become more difficult to deliver too, as one service provider has no control over another service provider's provisioning and traffic management policy. To solve the inter-domain QoS problem, service providers have to cooperate. Once they agree at a contractual level to provider high quality of service to certain traffic (e.g., PWE traffic), the mechanisms defined in other working groups, e.g., Diffserv WG, can be used.

Inter-domain PSN tunnels are generally more difficult to set up, tear down and maintain than intra-domain ones. But that is an issue for PSN tunneling protocols such as MPLS and L2TPv3 and is outside the scope of PWE3.

11. Security Considerations

The PW end-point, PW demultiplexing mechanism, and the payloads of the native service can all be vulnerable to attack. PWE3 should leverage security mechanisms provided by the PW Demultiplexer or PSN Layers. Such mechanisms SHOULD protect PW end-point and PW Demultiplexer mechanism from denial-of-service (DoS) attacks and spoofing of the native data units. Preventing unauthorized access to PW end-points and other network devices is generally effective against DoS attacks and spoofing, and can be part of protection mechanism. Protection mechanisms SHOULD also address the spoofing of tunneled PW data. The validation of traffic addressed to the PW Demultiplexer end-point is paramount in ensuring integrity of PW encapsulation. Security protocols such as IPsec [RFC2401] can be used.

12. Acknowledgments

The authors would like to acknowledge input from M. Aissaoui, M. Bocci, S. Bryant, R. Cohen, N. Harrison, G. Heron, T. Johnson, A. Malis, L. Martini, E. Rosen, J. Rutemiller, T. So, Y. Stein, and S. Vainshtein.

13. References

13.1. Normative References

[IFMIB] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[SMIV2] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

13.2. Informative References

[G805] "Generic Functional Architecture of Transport Networks", ITU-T Recommendation G.805, 2000.

[IPTUNMIB] Thaler, D., "IP Tunnel MIB", RFC 2667, August 1999.

[L2TPv3] Lau, J., Townsley, M., and I. Goyret, et al., "Layer Two Tunneling Protocol (Version 3)", Work in Progress, June 2004.

[MARS] Armitage, G., "Support for Multicast over UNI 3.0/3.1 based ATM Networks", RFC 2022, November 1996.

[MPLS] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[PWE3_ARCH] S. Bryant and P. Pate, et. al., "PWE3 Architecture", Work in Progress, March 2004.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[TEMIB] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)", RFC 3812, June 2004.

[UNI3.0] ATM Forum, "ATM User-Network Interface Specification Version 3.0", Sept. 1993.

14. Authors' Addresses

XiPeng Xiao (Editor)
Riverstone Networks
5200 Great America Parkway
Santa Clara, CA 95054

EMail: xxiao@riverstonenet.com

Danny McPherson (Editor)
Arbor Networks

EMail: danny@arbor.net

Prayson Pate (Editor)
Overture Networks
507 Airport Boulevard, Suite 111
Morrisville, NC, USA 27560

EMail: prayson.pate@overturenetworks.com

Vijay Gill
AOL Time Warner

EMail: vijaygill9@aol.com

Kireeti Kompella
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, CA 94089

EMail: kireeti@juniper.net

Thomas D. Nadeau
Cisco Systems, Inc.
300 Beaver Brook Drive
Boxborough, MA 01719
EMail: tnadeau@cisco.com

Craig White
Level 3 Communications, LLC.
1025 Eldorado Blvd.
Broomfield, CO, 80021

EMail: Craig.White@Level3.com

15. Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

Exhibit 6

Network Working Group
Request for Comments: 3985
Category: Informational

S. Bryant, Ed.
Cisco Systems
P. Pate, Ed.
Overture Networks, Inc.

March 2005

Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture

Status of This Memo

Copyright Notice

Abstract

This document describes an architecture for Pseudo Wire Emulation Edge-to-Edge (PWE3). It discusses the emulation of services such as Frame Relay, ATM, Ethernet, TDM, and SONET/SDH over packet switched networks (PSNs) using IP or MPLS. It presents the architectural framework for pseudo wires (PWs), defines terminology, and specifies the various protocol elements and their functions.

Table of Contents

1. Introduction

This document describes an architecture for Pseudo Wire Emulation Edge-to-Edge (PWE3) in support of [RFC3916]. It discusses the emulation of services such as Frame Relay, ATM, Ethernet, TDM, and SONET/SDH over packet switched networks (PSNs) using IP or MPLS. It presents the architectural framework for pseudo wires (PWs), defines terminology, and specifies the various protocol elements and their functions.

1.1. Pseudo Wire Definition

PWE3 is a mechanism that emulates the essential attributes of a telecommunications service (such as a T1 leased line or Frame Relay) over a PSN. PWE3 is intended to provide only the minimum necessary functionality to emulate the wire with the required degree of faithfulness for the given service definition. Any required switching functionality is the responsibility of a forwarder function

(FWRD). Any translation or other operation needing knowledge of the payload semantics is carried out by native service processing (NSP) elements. The functional definition of any FWRD or NSP elements is outside the scope of PWE3.

The required functions of PWs include encapsulating service-specific bit streams, cells, or PDUs arriving at an ingress port and carrying them across an IP path or MPLS tunnel. In some cases it is necessary to perform other operations such as managing their timing and order, to emulate the behavior and characteristics of the service to the required degree of faithfulness.

From the perspective of Customer Edge Equipment (CE), the PW is characterized as an unshared link or circuit of the chosen service. In some cases, there may be deficiencies in the PW emulation that impact the traffic carried over a PW and therefore limit the applicability of this technology. These limitations must be fully described in the appropriate service-specific documentation.

For each service type, there will be one default mode of operation that all PEs offering that service type must support. However, optional modes may be defined to improve the faithfulness of the emulated service, if it can be clearly demonstrated that the additional complexity associated with the optional mode is offset by the value it offers to PW users.

1.2. PW Service Functionality

PWs provide the following functions in order to emulate the behavior and characteristics of the native service.

- Encapsulation of service-specific PDUs or circuit data arriving at the PE-bound port (logical or physical).
- Carriage of the encapsulated data across a PSN tunnel.
- Establishment of the PW, including the exchange and/or distribution of the PW identifiers used by the PSN tunnel endpoints.
- Managing the signaling, timing, order, or other aspects of the service at the boundaries of the PW.
- Service-specific status and alarm management.

1.3. Non-Goals of This Document

The following are non-goals for this document:

- The on-the-wire specification of PW encapsulations.
- The detailed definition of the protocols involved in PW setup and maintenance.

The following are outside the scope of PWE3:

- Any multicast service not native to the emulated medium. Thus, Ethernet transmission to a "multicast" IEEE-48 address is in scope, but multicast services such as MARS [RFC2022] that are implemented on top of the medium are not.
- Methods to signal or control the underlying PSN.

1.4. Terminology

This document uses the following definitions of terms. These terms are illustrated in context in Figure 2.

Attachment Circuit (AC)	The physical or virtual circuit attaching a CE to a PE. An attachment Circuit may be, for example, a Frame Relay DLCI, an ATM VPI/VCI, an Ethernet port, a VLAN, a PPP connection on a physical interface, a PPP session from an L2TP tunnel, or an MPLS LSP. If both physical and virtual ACs are of the same technology (e.g., both ATM, both Ethernet, both Frame Relay), the PW is said to provide "homogeneous transport"; otherwise, it is said to provide "heterogeneous transport".
CE-bound	The traffic direction in which PW-PDUs are received on a PW via the PSN, processed, and then sent to the destination CE.
CE Signaling	Messages sent and received by the CE’s control plane. It may be desirable or even necessary for the PE to participate in or to monitor this signaling in order to emulate the service effectively.
Control Word (CW)	A four-octet header used in some encapsulations to carry per-packet information when the PSN is MPLS.

Customer Edge (CE)	A device where one end of a service originates and/or terminates. The CE is not aware that it is using an emulated service rather than a native service.
Forwarder (FWRD)	A PE subsystem that selects the PW to use in order to transmit a payload received on an AC.
Fragmentation	The action of dividing a single PDU into multiple PDUs before transmission with the intent of the original PDU being reassembled elsewhere in the network. Packets may undergo fragmentation if they are larger than the MTU of the network they will traverse.
Maximum Transmission unit (MTU)	The packet size (excluding data link header) that an interface can transmit without needing to fragment.
Native Service Processing (NSP)	Processing of the data received by the PE from the CE before presentation to the PW for transmission across the core, or processing of the data received from a PW by a PE before it is output on the AC. NSP functionality is defined by standards bodies other than the IETF, such as ITU-T,ANSI, or ATMF.)
Packet Switched Network (PSN)	Within the context of PWE3, this is a network using IP or MPLS as the mechanism for packet forwarding.
PE-Bound	The traffic direction in which information from a CE is adapted to a PW, and PW-PDUs are sent into the PSN.
PE/PW Maintenance	Used by the PEs to set up, maintain, and tear down the PW. It may be coupled with CE Signaling in order to manage the PW effectively.
Protocol Data Unit (PDU)	The unit of data output to, or received from, the network by a protocol layer.
Provider Edge (PE)	A device that provides PWE3 to a CE.
Pseudo Wire (PW)	A mechanism that carries the essential elements of an emulated service from one PE to one or more other PEs over a PSN.

Pseudo Wire Emulation Edge to Edge (PWE3)	A mechanism that emulates the essential attributes of service (such as a T1 leased line or Frame Relay) over a PSN.
Pseudo Wire PDU (PW-PDU)	A PDU sent on the PW that contains all of the data and control information necessary to emulate the desired service.
PSN Tunnel	A tunnel across a PSN, inside which one or more PWs can be carried.
PSN Tunnel Signaling	Used to set up, maintain, and tear down the underlying PSN tunnel.
PW Demultiplexer	Data-plane method of identifying a PW terminating at a PE.
Time Domain Multiplexing (TDM)	Time Division Multiplexing. Frequently used to refer to the synchronous bit streams at rates defined by G.702.
Tunnel	A method of transparently carrying information over a network.

2. PWE3 Applicability

The PSN carrying a PW will subject payload packets to loss, delay, delay variation, and re-ordering. During a network transient there may be a sustained period of impaired service. The applicability of PWE3 to a particular service depends on the sensitivity of that service (or the CE implementation) to these effects, and on the ability of the adaptation layer to mask them. Some services, such as IP over FR over PWE3, may prove quite resilient to IP and MPLS PSN characteristics. Other services, such as the interconnection of PBX systems via PWE3, will require more careful consideration of the PSN and adaptation layer characteristics. In some instances, traffic engineering of the underlying PSN will be required, and in some cases the constraints may make the required service guarantees impossible to provide.

3. Protocol Layering Model

The PWE3 protocol-layering model is intended to minimize the differences between PWs operating over different PSN types. The design of the protocol-layering model has the goals of making each PW definition independent of the underlying PSN, and of maximizing the reuse of IETF protocol definitions and their implementations.

3.1. Protocol Layers

The logical protocol-layering model required to support a PW is shown in Figure 1.

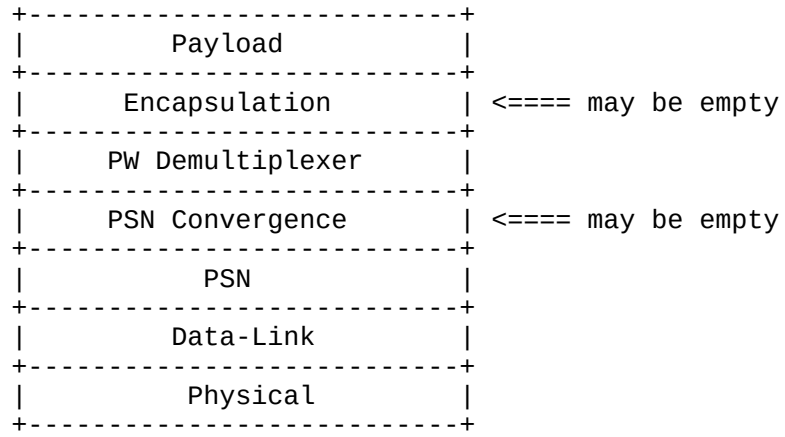


Figure 1. Logical Protocol Layering Model

The payload is transported over the Encapsulation Layer. The Encapsulation Layer carries any information, not already present within the payload itself, that is needed by the PW CE-bound PE interface to send the payload to the CE via the physical interface. If no further information is needed in the payload itself, this layer is empty.

The Encapsulation Layer also provides support for real-time processing, and if needed for sequencing.

The PW Demultiplexer layer provides the ability to deliver multiple PWs over a single PSN tunnel. The PW demultiplexer value used to identify the PW in the data plane may be unique per PE, but this is not a PWE3 requirement. It must, however, be unique per tunnel endpoint. If it is necessary to identify a particular tunnel, then that is the responsibility of the PSN layer.

The PSN Convergence layer provides the enhancements needed to make the PSN conform to the assumed PSN service requirement. Therefore, this layer provides a consistent interface to the PW, making the PW independent of the PSN type. If the PSN already meets the service requirements, this layer is empty.

The PSN header, MAC/Data-Link, and Physical Layer definitions are outside the scope of this document. The PSN can be IPv4, IPv6, or MPLS.

3.2. Domain of PWE3

PWE3 defines the Encapsulation Layer, the method of carrying various payload types, and the interface to the PW Demultiplexer Layer. It is expected that the other layers will be provided by tunneling methods such as L2TP or MPLS over the PSN.

3.3. Payload Types

The payload is classified into the following generic types of native data units:

- Packet
- Cell
- Bit stream
- Structured bit stream

Within these generic types there are specific service types:

Generic Payload Type	PW Service
Packet	Ethernet (all types), HDLC framing, Frame Relay, ATM AAL5 PDU.
Cell	ATM.
Bit stream	Unstructured E1, T1, E3, T3.
Structured bit stream	SONET/SDH (e.g., SPE, VT, NxDS0).

3.3.1. Packet Payload

A packet payload is a variable-size data unit delivered to the PE via the AC. A packet payload may be large compared to the PSN MTU. The delineation of the packet boundaries is encapsulation specific. HDLC or Ethernet PDUs can be considered examples of packet payloads. Typically, a packet will be stripped of transmission overhead such as HDLC flags and stuffing bits before transmission over the PW.

A packet payload would normally be relayed across the PW as a single unit. However, there will be cases where the combined size of the packet payload and its associated PWE3 and PSN headers exceeds the PSN path MTU. In these cases, some fragmentation methodology has to be applied. This may, for example, be the case when a user provides

the service and attaches to the service provider via Ethernet, or when nested pseudo-wires are involved. Fragmentation is discussed in more detail in section 5.3.

A packet payload may need sequencing and real-time support.

In some situations, the packet payload may be selected from the packets presented on the emulated wire on the basis of some sub-multiplexing technique. For example, one or more Frame Relay PDUs may be selected for transport over a particular pseudo wire based on the Frame Relay Data-Link Connection Identifier (DLCI), or, in the case of Ethernet payloads, by using a suitable MAC bridge filter. This is a forwarder function, and this selection would therefore be made before the packet was presented to the PW Encapsulation Layer.

3.3.2. Cell Payload

A cell payload is created by capturing, transporting, and replaying groups of octets presented on the wire in a fixed-size format. The delineation of the group of bits that comprise the cell is specific to the encapsulation type. Two common examples of cell payloads are ATM 53-octet cells, and the larger 188-octet MPEG Transport Stream packets [DVB].

To reduce per-PSN packet overhead, multiple cells may be concatenated into a single payload. The Encapsulation Layer may consider the payload complete on the expiry of a timer, after a fixed number of cells have been received or when a significant cell (e.g., an ATM OAM cell) has been received. The benefit of concatenating multiple PDUs should be weighed against a possible increase in packet delay variation and the larger penalty incurred by packet loss. In some cases, it may be appropriate for the Encapsulation Layer to perform some type of compression, such as silence suppression or voice compression.

The generic cell payload service will normally need sequence number support and may also need real-time support. The generic cell payload service would not normally require fragmentation.

The Encapsulation Layer may apply some form of compression to some of these sub-types (e.g., idle cells may be suppressed).

In some instances, the cells to be incorporated in the payload may be selected by filtering them from the stream of cells presented on the wire. For example, an ATM PWE3 service may select cells based on their VCI or VPI fields. This is a forwarder function, and the selection would therefore be made before the packet was presented to the PW Encapsulation Layer.

3.3.3. Bit Stream

A bit stream payload is created by capturing, transporting, and replaying the bit pattern on the emulated wire, without taking advantage of any structure that, on inspection, may be visible within the relayed traffic (i.e., the internal structure has no effect on the fragmentation into packets).

In some instances it is possible to apply suppression to bit streams. For example, E1 and T1 send "all-ones" to indicate failure. This condition can be detected without any knowledge of the structure of the bit stream, and transmission of packetized can be data suppressed.

This service will require sequencing and real-time support.

3.3.4. Structured Bit Stream

A structured bit stream payload is created by using some knowledge of the underlying structure of the bit stream to capture, transport, and replay the bit pattern on the emulated wire.

Two important points distinguish structured and unstructured bit streams:

- Some parts of the original bit stream may be stripped in the PSN-bound direction by an NSP block. For example, in Structured SONET the section and line overhead (and possibly more) may be stripped. A framer is required to enable such stripping. It is also required for frame/payload alignment for fractional T1/E1 applications.

- The PW must preserve the structure across the PSN so that the CE-bound NSP block can insert it correctly into the reconstructed unstructured bit stream. The stripped information (such as SONET pointer justifications) may appear in the encapsulation layer to facilitate this reconstitution.

As an option, the Encapsulation Layer may also perform silence/idle suppression or similar compression on a structured bit stream.

Structured bit streams are distinguished from cells in that the structures may be too long to be carried in a single packet. Note that "short" structures are indistinguishable from cells and may benefit from the use of methods described in section 3.3.2.

This service requires sequencing and real-time support.

3.3.5. Principle of Minimum Intervention

To minimize the scope of information, and to improve the efficiency of data flow through the Encapsulation Layer, the payload should be transported as received, with as few modifications as possible [RFC1958].

This minimum intervention approach decouples payload development from PW development and requires fewer translations at the NSP in a system with similar CE interfaces at each end. It also prevents unwanted side effects due to subtle misrepresentation of the payload in the intermediate format.

An approach that does intervene can be more wire efficient in some cases and may result in fewer translations at the NSP whereby the CE interfaces are of different types. Any intermediate format effectively becomes a new framing type, requiring documentation and assured interoperability. This increases the amount of work for handling the protocol that the intermediate format carries and is undesirable.

4. Architecture of Pseudo Wires

This section describes the PWE3 architectural model.

4.1. Network Reference Model

Figure 2 illustrates the network reference model for point-to-point PWs.

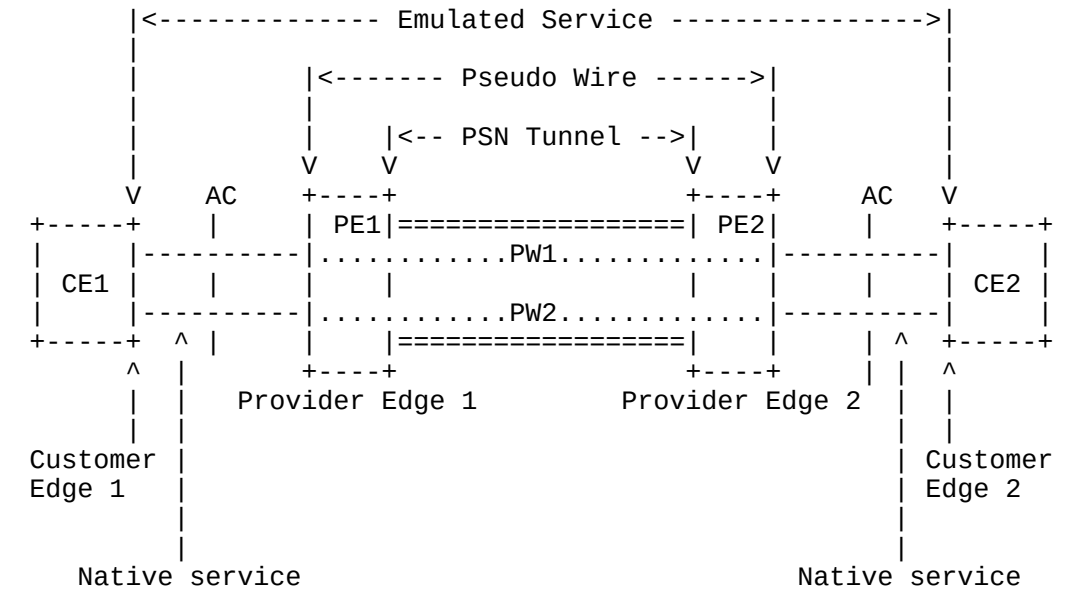


Figure 2. PWE3 Network Reference Model

The two PEs (PE1 and PE2) have to provide one or more PWs on behalf of their client CEs (CE1 and CE2) to enable the client CEs to communicate over the PSN. A PSN tunnel is established to provide a data path for the PW. The PW traffic is invisible to the core network, and the core network is transparent to the CEs. Native data units (bits, cells, or packets) arrive via the AC, are encapsulated in a PW-PDU, and are carried across the underlying network via the PSN tunnel. The PEs perform the necessary encapsulation and decapsulation of PW-PDUs and handle any other functions required by the PW service, such as sequencing or timing.

4.2. PWE3 Pre-processing

Some applications have to perform operations on the native data units received from the CE (including both payload and signaling traffic) before they are transmitted across the PW by the PE. Examples include Ethernet bridging, SONET cross-connect, translation of locally-significant identifiers such as VCI/VPI, or translation to another service type. These operations could be carried out in external equipment, and the processed data could be sent to the PE

over one or more physical interfaces. In most cases, could be in undertaking these operations within the PE provides cost and operational benefits. Processed data is then presented to the PW via a virtual interface within the PE. These pre-processing operations are included in the PWE3 reference model to provide a common reference point, but the detailed description of these operations is outside the scope of the PW definition given here.

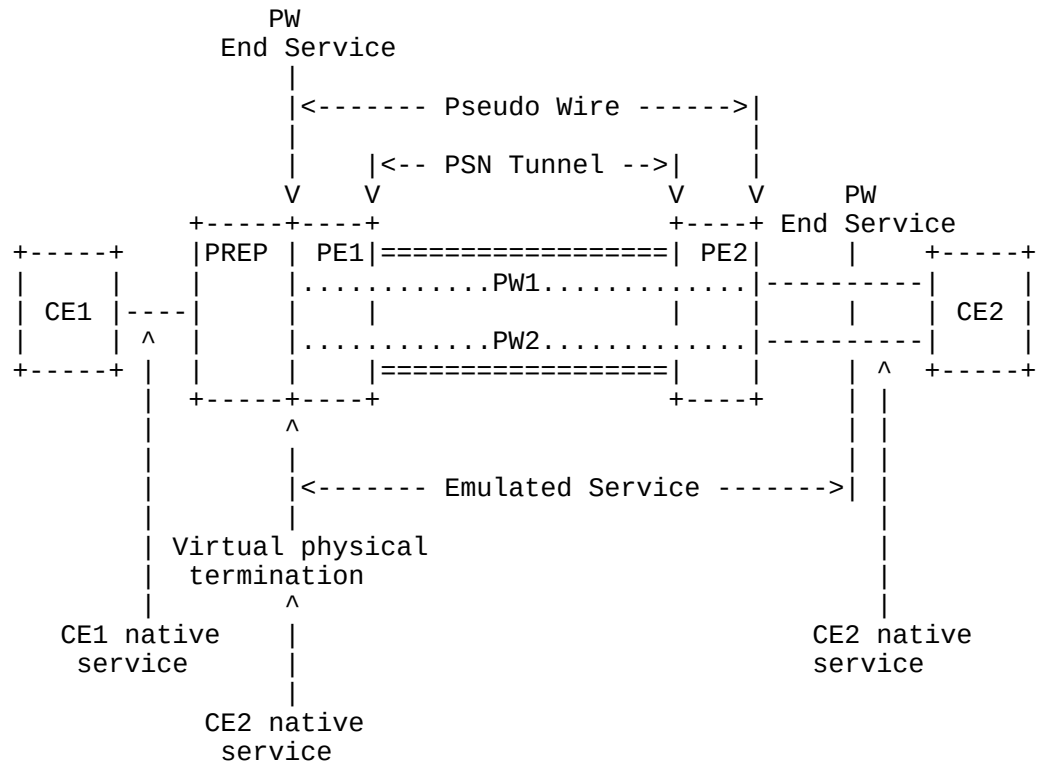


Figure 3. Pre-processing within the PWE3 Network Reference Model

Figure 3 shows the interworking of one PE with pre-processing (PREP), and a second without this functionality. This reference point emphasizes that the functional interface between PREP and the PW is that represented by a physical interface carrying the service. This effectively defines the necessary inter-working specification.

The operation of a system in which both PEs include PREP functionality is also supported.

The required pre-processing can be divided into two components:

- o Forwarder (FWRD)
- o Native Service Processing (NSP)

4.2.1. Forwarders

Some applications have to forward payload elements selectively from one or more ACs to one or more PWs. In such cases, there will also be a need to perform the inverse function on PWE3-PDUs received by a PE from the PSN. This is the function of the forwarder.

The forwarder selects the PW based on, for example, the incoming AC, the contents of the payload, or some statically and/or dynamically configured forwarding information.

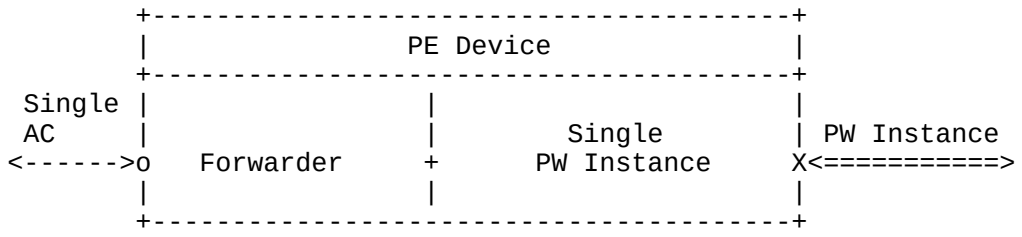


Figure 4a. Simple Point-to-Point Service

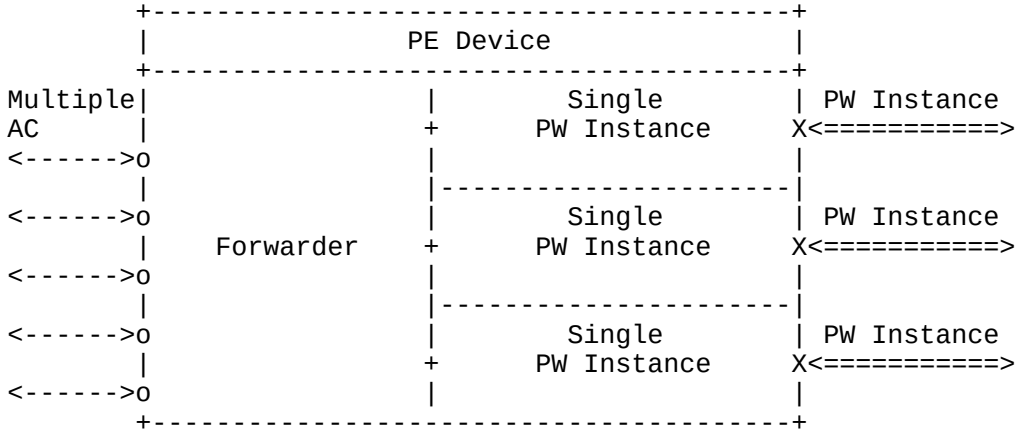


Figure 4b. Multiple AC to Multiple PW Forwarding

Figure 4a shows a simple forwarder that performs some type of filtering operation. Because the forwarder has a single input and a single output interface, filtering is the only type of forwarding

operation that applies. Figure 4b shows a more general forwarding situation where payloads are extracted from one or more ACs and directed to one or more PWs. In this case filtering, direction, and combination operations may be performed on the payloads. For example, if the AC were Frame Relay, the forwarder might perform Frame Relay switching and the PW instances might be the inter-switch links.

4.2.2. Native Service Processing

Some applications required some form of data or address translation, or some other operation requiring knowledge of the semantics of the payload. This is the function of the Native Service Processor (NSP).

The use of the NSP approach simplifies the design of the PW by restricting a PW to homogeneous operation. NSP is included in the reference model to provide a defined interface to this functionality. The specification of the various types of NSP is outside the scope of PWE3.

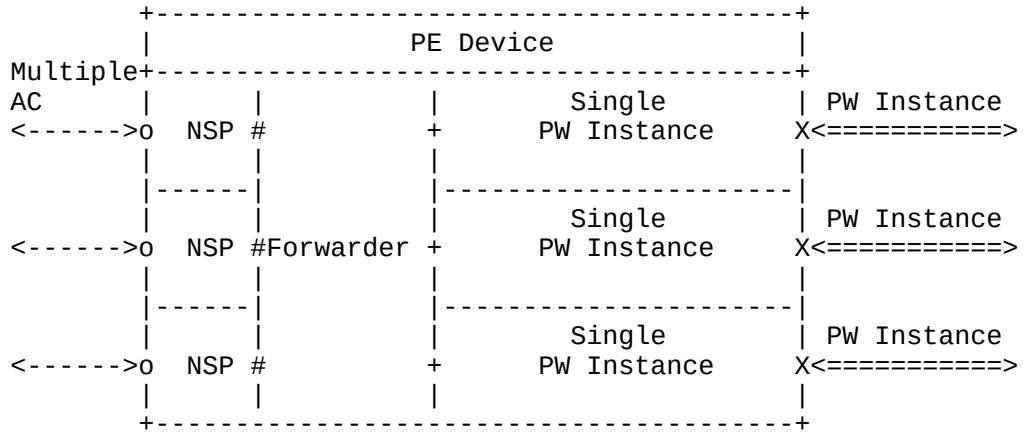


Figure 5. NSP in a Multiple AC to Multiple PW Forwarding PE

Figure 5 illustrates the relationship between NSP, forwarder, and PWs in a PE. The NSP function may apply any transformation operation (modification, injection, etc.) on the payloads as they pass between the physical interface to the CE and the virtual interface to the forwarder. These transformation operations will, of course, be limited to those that have been implemented in the data path, and that are enabled by the PE configuration. A PE device may contain more than one forwarder.

This model also supports the operation of a system in which the NSP functionality includes terminating the data-link, and the application of Network Layer processing to the payload.

4.3. Maintenance Reference Model

Figure 6 illustrates the maintenance reference model for PWs.

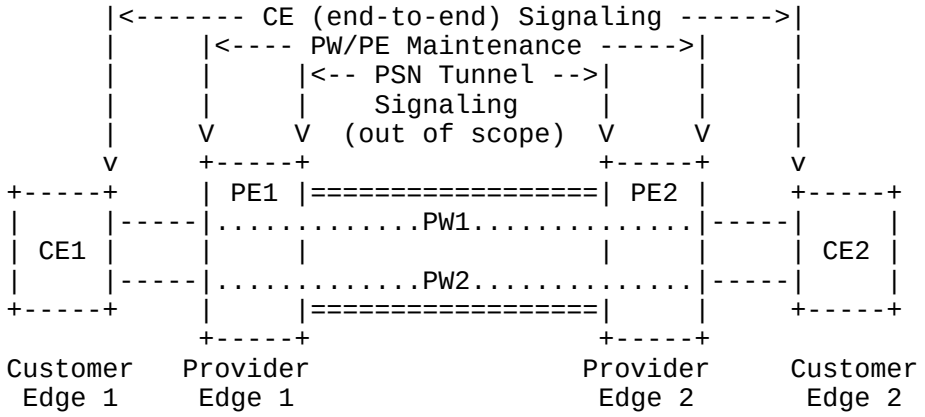


Figure 6. PWE3 Maintenance Reference Model

The following signaling mechanisms are required:

- The CE (end-to-end) signaling is between the CEs. This signaling could be Frame Relay PVC status signaling, ATM SVC signaling, TDM CAS signaling, etc.

- The PW/PE Maintenance is used between the PEs (or NSPs) to set up, maintain, and tear down PWs, including any required coordination of parameters.

- The PSN Tunnel signaling controls the PW multiplexing and some elements of the underlying PSN. Examples are L2TP control protocol, MPLS LDP, and RSVP-TE. The definition of the information that PWE3 needs signaled is within the scope of PWE3, but the signaling protocol itself is not.

4.4. Protocol Stack Reference Model

Figure 7 illustrates the protocol stack reference model for PWs.

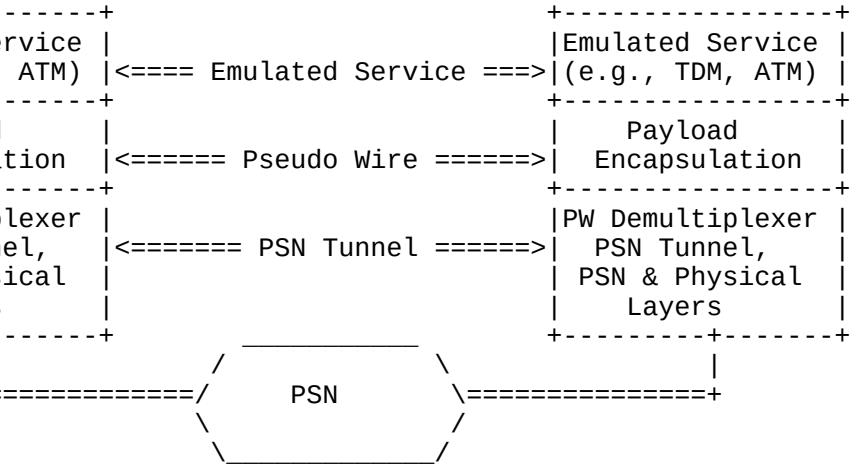


Figure 7. PWE3 Protocol Stack Reference Model

The PW provides the CE with an emulated physical or virtual connection to its peer at the far end. Native service PDUs from the CE are passed through an Encapsulation Layer at the sending PE and then sent over the PSN. The receiving PE removes the encapsulation and restores the payload to its native format for transmission to the destination CE.

4.5. Pre-processing Extension to Protocol Stack Reference Model

Figure 8 illustrates how the protocol stack reference model is extended to include the provision of pre-processing (forwarding and NSP). This shows the placement of the physical interface relative to the CE.

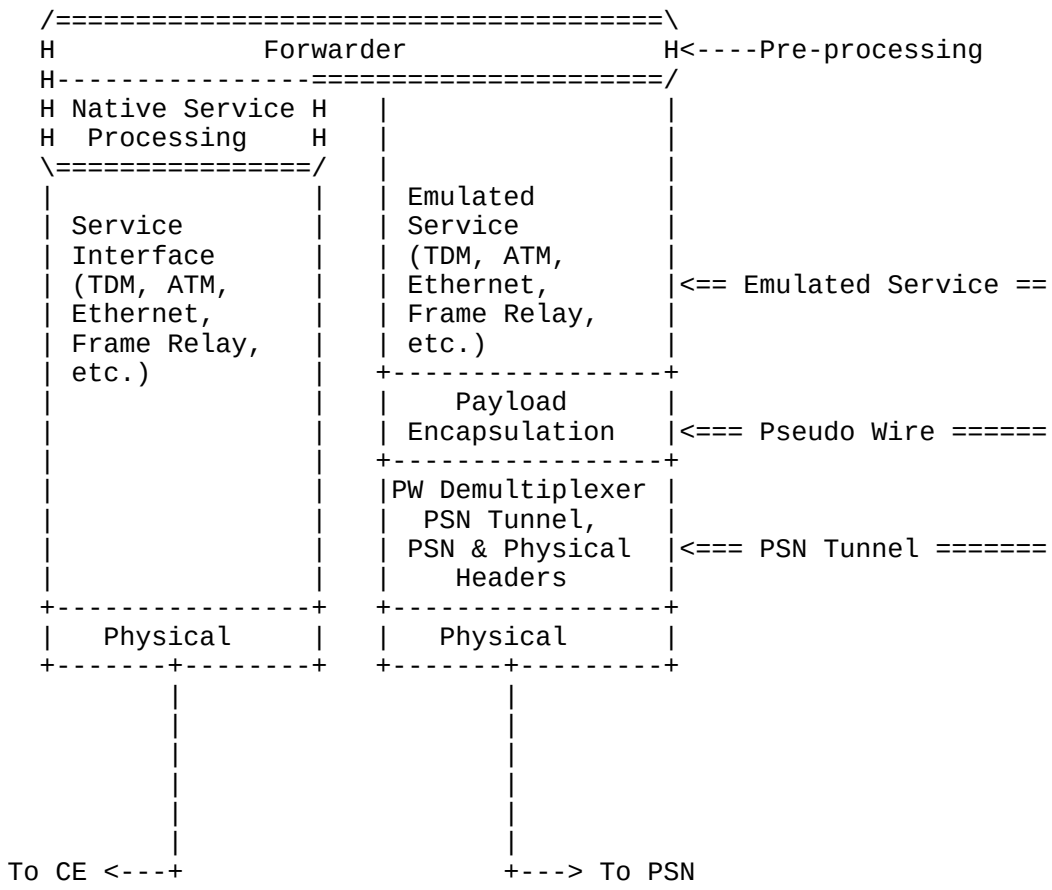


Figure 8. Protocol Stack Reference Model with Pre-processing

5. PW Encapsulation

The PW Encapsulation Layer provides the necessary infrastructure to adapt the specific payload type being transported over the PW to the PW Demultiplexer Layer used to carry the PW over the PSN.

The PW Encapsulation Layer consists of three sub-layers:

- o Payload Convergence
- o Timing
- o Sequencing

The PW Encapsulation sub-layering and its context with the protocol stack are shown in Figure 9.

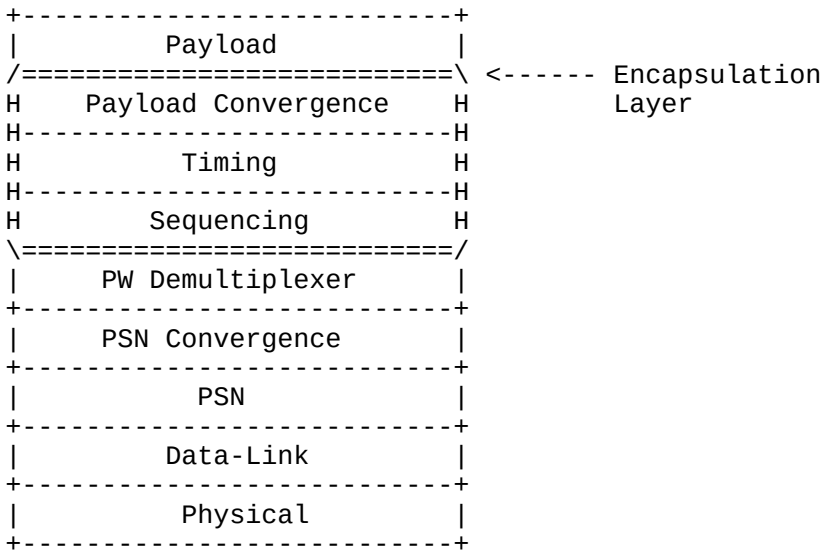


Figure 9. PWE3 Encapsulation Layer in Context

The Payload Convergence sub-layer is highly tailored to the specific payload type. However grouping a number of target payload types into a generic class, and then providing a single convergence sub-layer type common to the group, reduces the number of payload convergence sub-layer types. This decreases implementation complexity. The provision of per-packet signaling and other out-of-band information (other than sequencing or timing) is undertaken by this layer.

The Timing and Sequencing Layers provide generic services to the Payload Convergence Layer for all payload types that require them.

5.1. Payload Convergence Layer

5.1.1. Encapsulation

The primary task of the Payload Convergence Layer is the encapsulation of the payload in PW-PDUs. The native data units to be encapsulated may contain an L2 header or L1 overhead. This is service specific. The Payload Convergence header carries the additional information needed to replay the native data units at the CE-bound physical interface. The PW Demultiplexer header is not considered part of the PW header.

Not all the additional information needed to replay the native data units have to be carried in the PW header of the PW PDUs. Some information (e.g., service type of a PW) may be stored as state information at the destination PE during PW set up.

5.1.2. PWE3 Channel Types

The PW Encapsulation Layer and its associated signaling require one or more of the following types of channels from its underlying PW Demultiplexer and PSN Layers (channel type 1 plus one or more of channel types 2 through 4):

1. A reliable control channel for signaling line events, status indications, and, in exceptional cases, CE-CE events that must be translated and sent reliably between PEs. PWE3 may need this type of control channel to provide faithful emulation of complex data-link protocols.

2. A high-priority, unreliable, sequenced channel. A typical use is for CE-to-CE signaling. "High priority" may simply be indicated via the DSCP bits for IP or the EXP bits for MPLS, giving the packet priority during transit. This channel type could also use a bit in the tunnel header itself to indicate that packets received at the PE should be processed with higher priority [RFC2474].

3. A sequenced channel for data traffic that is sensitive to packet reordering (one classification for use could be for any non-IP traffic).

4. An unsequenced channel for data traffic insensitive to packet order.

The data channels (2, 3, and 4 above) should be carried "in band" with one another to as much of a degree as is reasonably possible on a PSN.

Where end-to-end connectivity may be disrupted by address translation [RFC3022], access-control lists, firewalls, etc., the control channel may be able to pass traffic and setup the PW, while the PW data traffic is blocked by one or more of these mechanisms. In these cases unless the control channel is also carried "in band", the signaling to set up the PW will not confirm the existence of an end-to-end data path. In some cases there is a need to synchronize CE events with the data carried over a PW. This is especially the case

with TDM circuits (e.g., the on-hook/off-hook events in PSTN switches might be carried over a reliable control channel whereas the associated bit stream is carried over a sequenced data channel).

PWE3 channel types that are not needed by the supported PWs need not be included in such an implementation.

5.1.3. Quality of Service Considerations

Where possible, it is desirable to employ mechanisms to provide PW Quality of Service (QoS) support over PSNs.

5.2. Payload-Independent PW Encapsulation Layers

Two PWE3 Encapsulation sub-layers provide common services to all payload types: Sequencing and Timing. These services are optional and are only used if a particular PW instance needs them. If the service is not needed, the associated header may be omitted in order to conserve processing and network resources.

Sometimes a specific payload type will require transport with or without sequence and/or real-time support. For example, an invariant of Frame Relay transport is the preservation of packet order. Some Frame Relay applications expect delivery in order and may not cope with reordering of the frames. However, where the Frame Relay service is itself only being used to carry IP, it may be desirable to relax this constraint to reduce per-packet processing cost.

The guiding principle is that, when possible, an existing IETF protocol should be used to provide these services. When a suitable protocol is not available, the existing protocol should be extended or modified to meet the PWE3 requirements, thereby making that protocol available for other IETF uses. In the particular case of timing, more than one general method may be necessary to provide for the full scope of payload timing requirements.

5.2.1. Sequencing

The sequencing function provides three services: frame ordering, frame duplication detection, and frame loss detection. These services allow the emulation of the invariant properties of a physical wire. Support for sequencing depends on the payload type and may be omitted if it is not needed.

The size of the sequence-number space depends on the speed of the emulated service, and on the maximum time of the transient conditions in the PSN. A sequence number space greater than 2^16 may therefore be needed to prevent the sequence number space from wrapping during the transient.

5.2.1.1. Frame Ordering

When packets carrying the PW-PDUs traverse a PSN, they may arrive out of order at the destination PE. For some services, the frames (control frames, data frames, or both) must be delivered in order. For these services, some mechanism must be provided for ensuring in-order delivery. Providing a sequence number in the sequence sub-layer header for each packet is one possible approach. Alternatively, it can be noted that sequencing is a subset of the problem of delivering timed packets, and that a single combined mechanism such as [RFC3550] may be employed.

There are two possible misordering strategies:

- Drop misordered PW PDUs.
- Try to sort PW PDUs into the correct order.

The choice of strategy will depend on

- how critical the loss of packets is to the operation of the PW (e.g., the acceptable bit error rate),
- the speeds of the PW and PSN,
- the acceptable delay (as delay must be introduced to reorder), and
- the expected incidence of misordering.

5.2.1.2. Frame Duplication Detection

In rare cases, packets traversing a PW may be duplicated by the underlying PSN. For some services, frame duplication is not acceptable. For these services, some mechanism must be provided to ensure that duplicated frames will not be delivered to the destination CE. The mechanism may be the same as that used to ensure in-order frame delivery.

5.2.1.3. Frame Loss Detection

A destination PE can determine whether a frame has been lost by tracking the sequence numbers of the PW PDUs received.

In some instances, if a PW PDU fails to arrive within a certain time, a destination PE will have to presume that it is lost. If a PW-PDU that has been processed as lost subsequently arrives, the destination PE must discard it.

5.2.2. Timing

A number of native services have timing expectations based on the characteristics of the networks they were designed to travel over. The emulated service may have to duplicate these network characteristics as closely as possible: e.g., in delivering native traffic with bitrate, jitter, wander, and delay characteristics similar to those received at the sending PE.

In such cases, the receiving PE has to play out the native traffic as it was received at the sending PE. This relies on timing information either sent between the two PEs, or in some cases received from an external reference.

Therefore, Timing Sub-layer must support two timing functions: clock recovery and timed payload delivery. A particular payload type may require either or both of these services.

5.2.2.1. Clock Recovery

Clock recovery is the extraction of output transmission bit timing information from the delivered packet stream, and it requires a suitable mechanism. A physical wire carries the timing information natively, but extracting timing from a highly jittered source, such as packet stream, is a relatively complex task. Therefore, it is desirable that an existing real-time protocol such as [RFC3550] be used for this purpose, unless it can be shown that this is unsuitable or unnecessary for a particular payload type.

5.2.2.2. Timed Delivery

Timed delivery is the delivery of non-contiguous PW PDUs to the PW output interface with a constant phase relative to the input interface. The timing of the delivery may be relative to a clock derived from the packet stream received over the PSN clock recovery, or to an external clock.

5.3. Fragmentation

Ideally, a payload would be relayed across the PW as a single unit. However, there will be cases where the combined size of the payload and its associated PWE3 and PSN headers will exceed the PSN path MTU. When a packet size exceeds the MTU of a given network, fragmentation and reassembly have to be performed for the packet to be delivered. Since fragmentation and reassembly generally consume considerable network resources, as compared to simply switching a packet in its entirety, the need for fragmentation and reassembly throughout a network should be reduced or eliminated to the extent possible. Of particular concern for fragmentation and reassembly are aggregation points where large numbers of PWs are processed (e.g., at the PE).

Ideally, the equipment originating the traffic sent over the PW will have adaptive measures in place (e.g., [RFC1191], [RFC1981]) that ensure that packets needing to be fragmented are not sent. When this fails, the point closest to the sending host with fragmentation and reassembly capabilities should attempt to reduce the size of packets to satisfy the PSN MTU. Thus, in the reference model for PWE3 (Figure 3), fragmentation should first be performed at the CE if possible. Only if the CE cannot adhere to an acceptable MTU size for the PW should the PE attempt its own fragmentation method.

In cases where MTU management fails to limit the payload to a size suitable for transmission of the PW, the PE may fall back to either a generic PW fragmentation method or, if available, the fragmentation service of the underlying PSN.

It is acceptable for a PE implementation not to support fragmentation. A PE that does not will drop packets that exceed the PSN MTU, and the management plane of the encapsulating PE may be notified.

If the length of a L2/L1 frame, restored from a PW PDU, exceeds the MTU of the destination AC, it must be dropped. In this case, the management plane of the destination PE may be notified.

5.4. Instantiation of the Protocol Layers

This document does not address the detailed mapping of the Protocol Layering model to existing or future IETF standards. The instantiation of the logical Protocol Layering model is shown in Figure 9.

5.4.1. PWE3 over an IP PSN

The protocol definition of PWE3 over an IP PSN should employ existing IETF protocols where possible.

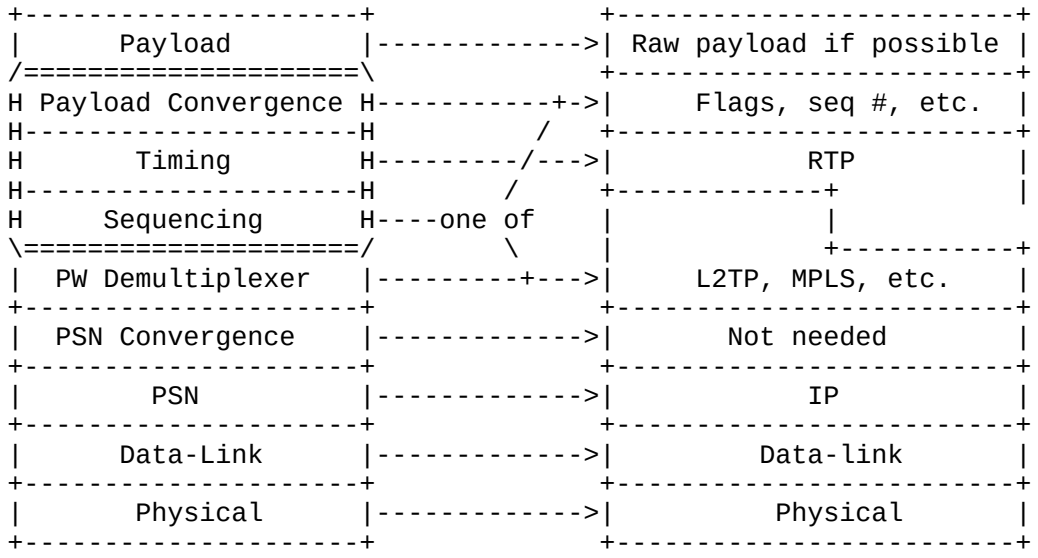


Figure 10. PWE3 over an IP PSN

Figure 10 shows the protocol layering for PWE3 over an IP PSN. As a rule, the payload should be carried as received from the NSP, with the Payload Convergence Layer provided when needed. However, in certain circumstances it may be justifiable to transmit the payload in some processed form. The reasons for this must be documented in the Encapsulation Layer definition for that payload type.

Where appropriate, explicit timing is provided by RTP [RFC3550], which, when used, also provides a sequencing service. When the PSN is UDP/IP, the RTP header follows the UDP header and precedes the PW control field. For all other cases the RTP header follows the PW control header.

The encapsulation layer may additionally carry a sequence number. Sequencing is to be provided either by RTP or by the PW encapsulation layer, but not by both.

PW Demultiplexing is provided by the PW label, which may take the form specified in a number of IETF protocols; e.g., an MPLS label [MPLSIP], an L2TP session ID [RFC3931], or a UDP port number [RFC768]. When PWs are carried over IP, the PSN Convergence Layer will not be needed.

As a special case, if the PW Demultiplexer is an MPLS label, the protocol architecture of section 5.4.2 can be used instead of the protocol architecture of this section.

5.4.2. PWE3 over an MPLS PSN

The MPLS ethos places importance on wire efficiency. By using a control word, some components of the PWE3 protocol layers can be compressed to increase this efficiency.

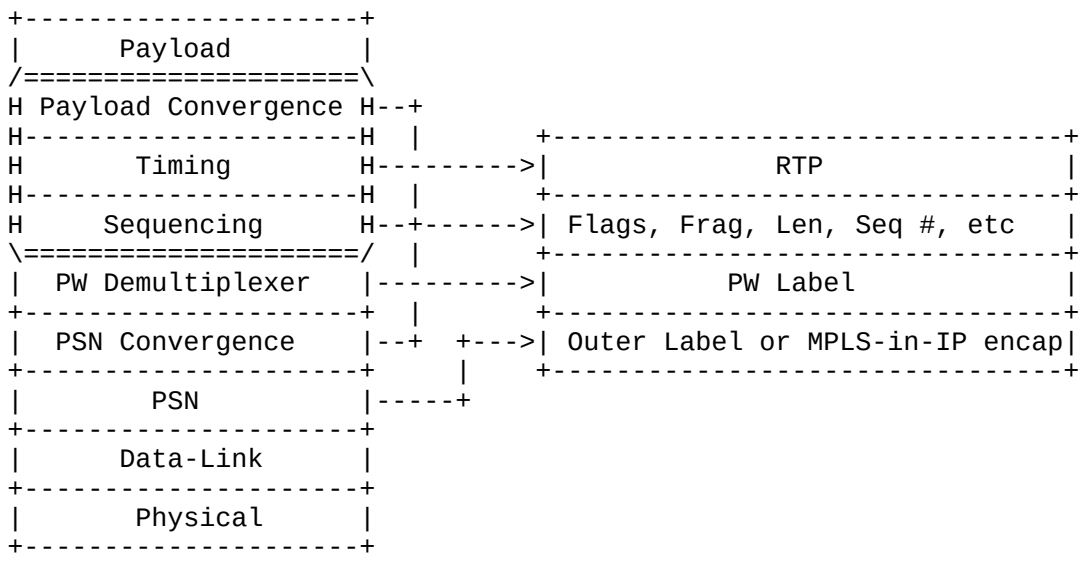


Figure 11. PWE3 over an MPLS PSN Using a Control Word

Figure 11 shows the protocol layering for PWE3 over an MPLS PSN. An inner MPLS label is used to provide the PW demultiplexing function. A control word is used to carry most of the information needed by the PWE3 Encapsulation Layer and the PSN Convergence Layer in a compact format. The flags in the control word provide the necessary payload convergence. A sequence field provides support for both in-order payload delivery and a PSN fragmentation service within the PSN Convergence Layer (supported by a fragmentation control method). Ethernet pads all frames to a minimum size of 64 bytes. The MPLS header does not include a length indicator. Therefore, to allow PWE3

to be carried in MPLS to pass correctly over an Ethernet data-link, a length correction field is needed in the control word. As with an IP PSN, where appropriate, timing is provided by RTP [RFC3550].

In some networks, it may be necessary to carry PWE3 over MPLS over IP. In these circumstances, the PW is encapsulated for carriage over MPLS as described in this section, and then a method of carrying MPLS over an IP PSN (such as GRE [RFC2784], [RFC2890]) is applied to the resultant PW-PDU.

5.4.3. PW-IP Packet Discrimination

For MPLS PSNs, there is an additional constraint on the PW packet format. Some label switched routers detect IP packets based on the initial four bits of the packet content. To facilitate proper functioning, these bits in PW packets must not be the same as an IP version number in current use.

6. PW Demultiplexer Layer and PSN Requirements

PWE3 places three service requirements on the protocol layers used to carry it across the PSN:

- Multiplexing
- Fragmentation
- Length and Delivery

6.1. Multiplexing

The purpose of the PW Demultiplexer Layer is to allow multiple PWs to be carried in a single tunnel. This minimizes complexity and conserves resources.

Some types of native service are capable of grouping multiple circuits into a "trunk"; e.g., multiple ATM VCs in a VP, multiple Ethernet VLANs on a physical media, or multiple DS0 services within a T1 or E1. A PW may interconnect two end-trunks. That trunk would have a single multiplexing identifier.

When a MPLS label is used as a PW Demultiplexer, setting of the TTL value [RFC3032] in the PW label is application specific.

6.2. Fragmentation

If the PSN provides a fragmentation and reassembly service of adequate performance, it may be used to obtain an effective MTU that is large enough to transport the PW PDUs. See section 5.3 for a full discussion of the PW fragmentation issues.

6.3. Length and Delivery

PDU delivery to the egress PE is the function of the PSN Layer.

If the underlying PSN does not provide all the information necessary to determine the length of a PW-PDU, the Encapsulation Layer must provide it.

6.4. PW-PDU Validation

It is a common practice to use an error detection mechanism such as a CRC or similar mechanism to ensure end-to-end integrity of frames. The PW service-specific mechanisms must define whether the packet's checksum shall be preserved across the PW or be removed from PE-bound PDUs and then be recalculated for insertion in CE-bound data.

The former approach saves work, whereas the latter saves bandwidth. For a given implementation, the choice may be dictated by hardware restrictions, which may not allow the preservation of the checksum.

For protocols such as ATM and FR, the scope of the checksum is restricted to a single link. This is because the circuit identifiers (e.g., FR DLCI or ATM VPI/VCI) only have local significance and are changed on each hop or span. If the circuit identifier (and thus checksum) were going to change as part of the PW emulation, it would be more efficient to strip and recalculate the checksum.

The service-specific document for each protocol must describe the validation scheme to be used.

6.5. Congestion Considerations

The PSN carrying the PW may be subject to congestion. The congestion characteristics will vary with the PSN type, the network architecture and configuration, and the loading of the PSN.

If the traffic carried over the PW is known to be TCP friendly (by, for example, packet inspection), packet discard in the PSN will trigger the necessary reduction in offered load, and no additional congestion avoidance action is necessary.

If the PW is operating over a PSN that provides enhanced delivery, the PEs should monitor packet loss to ensure that the requested service is actually being delivered. If it is not, then the PE should assume that the PSN is providing a best-effort service and should use the best-effort service congestion avoidance measures described below.

If best-effort service is being used and the traffic is not known to be TCP friendly, the PEs should monitor packet loss to ensure that the loss rate is within acceptable parameters. Packet loss is considered acceptable if a TCP flow across the same network path and experiencing the same network conditions would achieve an average throughput, measured on a reasonable timescale, not less than that which the PW flow is achieving. This condition can be satisfied by implementing a rate-limiting measure in the NSP, or by shutting down one or more PWs. The choice of which approach to use depends upon the type of traffic being carried. Where congestion is avoided by shutting down a PW, a suitable mechanism must be provided to prevent it from immediately returning to service and causing a series of congestion pulses.

The comparison to TCP cannot be specified exactly but is intended as an "order-of-magnitude" comparison in timescale and throughput. The timescale on which TCP throughput is measured is the round-trip time of the connection. In essence, this requirement states that it is not acceptable to deploy an application (using PWE3 or any other transport protocol) on the best-effort Internet, which consumes bandwidth arbitrarily and does not compete fairly with TCP within an order of magnitude. One method of determining an acceptable PW bandwidth is described in [RFC3448].

7. Control Plane

This section describes PWE3 control plane services.

7.1. Setup or Teardown of Pseudo Wires

A PW must be set up before an emulated service can be established and must be torn down when an emulated service is no longer needed.

Setup or teardown of a PW can be triggered by an operator command, from the management plane of a PE, by signaling set-up or teardown of an AC (e.g., an ATM SVC), or by an auto-discovery mechanism.

During the setup process, the PEs have to exchange information (e.g., learn each other’s capabilities). The tunnel signaling protocol may be extended to provide mechanisms that enable the PEs to exchange all necessary information on behalf of the PW.

Manual configuration of PWs can be considered a special kind of signaling and is allowed.

7.2. Status Monitoring

Some native services have mechanisms for status monitoring. For example, ATM supports OAM for this purpose. For these services, the corresponding emulated services must specify how to perform status monitoring.

7.3. Notification of Pseudo Wire Status Changes

7.3.1. Pseudo Wire Up/Down Notification

If a native service requires bi-directional connectivity, the corresponding emulated service can only be signaled as being up when the PW and PSN tunnels (if used), are functional in both directions.

Because the two CEs of an emulated service are not adjacent, a failure may occur at a place so that one or both physical links between the CEs and PEs remain up. For example, in Figure 2, if the physical link between CE1 and PE1 fails, the physical link between CE2 and PE2 will not be affected and will remain up. Unless CE2 is notified about the remote failure, it will continue to send traffic over the emulated service to CE1. Such traffic will be discarded at PE1. Some native services have failure notification so that when the services fail, both CEs will be notified. For these native services, the corresponding PWE3 service must provide a failure notification mechanism.

Similarly, if a native service has notification mechanisms so that all the affected services will change status from "Down" to "Up" when a network failure is fixed, the corresponding emulated service must provide a similar mechanism for doing so.

These mechanisms may already be built into the tunneling protocol. For example, the L2TP control protocol [RFC2661] [RFC3931] has this capability, and LDP has the ability to withdraw the corresponding MPLS label.

7.3.2. Misconnection and Payload Type Mismatch

With PWE3, misconnection and payload type mismatch can occur. Misconnection can breach the integrity of the system. Payload mismatch can disrupt the customer network. In both instances, there are security and operational concerns.

The services of the underlying tunneling mechanism and its associated control protocol can be used to mitigate this. As part of the PW setup, a PW-TYPE identifier is exchanged. This is then used by the forwarder and the NSP to verify the compatibility of the ACs.

7.3.3. Packet Loss, Corruption, and Out-of-Order Delivery

A PW can incur packet loss, corruption, and out-of-order delivery on the PSN path between the PEs. This can affect the working condition of an emulated service. For some payload types, packet loss, corruption, and out-of-order delivery can be mapped either to a bit error burst, or to loss of carrier on the PW. If a native service has some mechanism to deal with bit error, the corresponding PWE3 service should provide a similar mechanism.

7.3.4. Other Status Notification

A PWE3 approach may provide a mechanism for other status notifications, if any are needed.

7.3.5. Collective Status Notification

The status of a group of emulated services may be affected identically by a single network incident. For example, when the physical link (or sub-network) between a CE and a PE fails, all the emulated services that go through that link (or sub-network) will fail. It is likely that a group of emulated services all terminate at a remote CE. There may also be multiple such CEs affected by the failure. Therefore, it is desirable that a single notification message be used to notify failure of the whole group of emulated services.

A PWE3 approach may provide a mechanism for notifying status changes of a group of emulated circuits. One possible method is to associate each emulated service with a group ID when the PW for that emulated service is set up. Multiple emulated services can then be grouped by associating them with the same group ID. In status notification, this group ID can be used to refer all the emulated services in that group. The group ID mechanism should be a mechanism provided by the underlying tunnel signaling protocol.

7.4. Keep-Alive

If a native service has a keep-alive mechanism, the corresponding emulated service must provide a mechanism to propagate it across the PW. Transparently transporting keep-alive messages over the PW would follow the principle of minimum intervention. However, to reproduce

the semantics of the native mechanism accurately, some PWs may require an alternative approach, such as piggy-backing on the PW signaling mechanism.

7.5. Handling Control Messages of the Native Services

Some native services use control messages for circuit maintenance. These control messages may be in-band (e.g., Ethernet flow control, ATM performance management, or TDM tone signaling) or out-of-band, (e.g., the signaling VC of an ATM VP, or TDM CCS signaling).

Given the principle of minimum intervention, it is desirable that the PEs participate as little as possible in the signaling and maintenance of the native services. This principle should not, however, override the need to emulate the native service satisfactorily.

If control messages are passed through, it may be desirable to send them by using either a higher priority or a reliable channel provided by the PW Demultiplexer layer. See Section 5.1.2, PWE3 Channel Types.

8. Management and Monitoring

This section describes the management and monitoring architecture for PWE3.

8.1. Status and Statistics

The PE should report the status of the interface and tabulate statistics that help monitor the state of the network and help measure service-level agreements (SLAs). Typical counters include the following:

- Counts of PW-PDUs sent and received, with and without errors.
- Counts of sequenced PW-PDUs lost.
- Counts of service PDUs sent and received over the PSN, with and without errors (non-TDM).
- Service-specific interface counts.
- One-way delay and delay variation.

These counters would be contained in a PW-specific MIB, and they should not replicate existing MIB counters.

8.2. PW SNMP MIB Architecture

This section describes the general architecture for SNMP MIBs used to manage PW services and the underlying PSN. The intent here is to provide a clear picture of how all the pertinent MIBs fit together to form a cohesive management framework for deploying PWE3 services. Note that the names of MIB modules used below are suggestions and do not necessarily require that the actual modules used to realize the components in the architecture be named exactly so.

8.2.1. MIB Layering

The SNMP MIBs created for PWE3 should fit the architecture shown in Figure 12. The architecture provides a layered modular model into which any supported emulated service can be connected to any supported PSN type. This model fosters reuse of as much functionality as possible. For instance, the emulated service layer MIB modules do not redefine the existing emulated service MIB module; rather, they only associate it with the pseudo wires used to carry the emulated service over the configured PSN. In this way, the PWE3 MIB architecture follows the overall PWE3 architecture.

The architecture does allow for the joining of unsupported emulated service or PSN types by simply defining additional MIB modules to associate new types with existing ones. These new modules can subsequently be standardized. Note that there is a separate MIB module for each emulated service, as well as one for each underlying PSN. These MIB modules may be used in various combinations as needed.

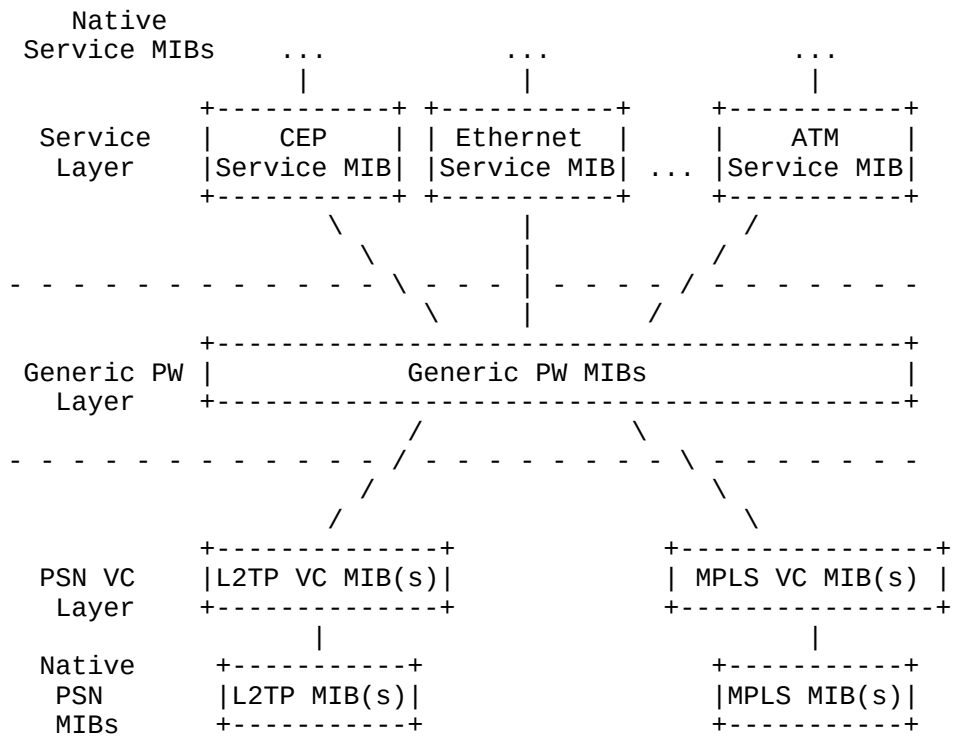


Figure 12. MIB Module Layering Relationship

Figure 13 shows an example for a SONET PW carried over MPLS Traffic Engineering Tunnel and an LDP-signaled LSP.

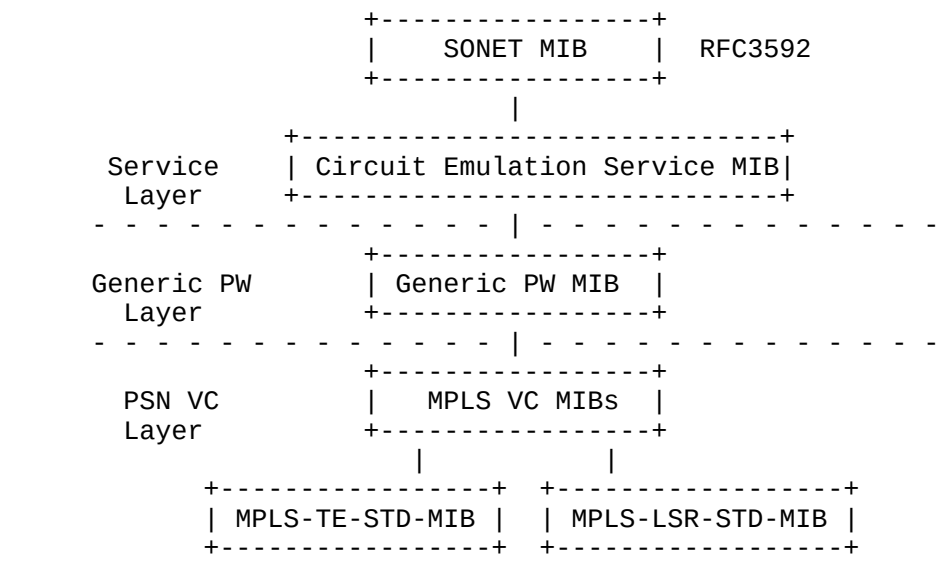


Figure 13. SONET PW over MPLS PSN Service-Specific Example

8.2.2. Service Layer MIB Modules

This conceptual layer in the model contains MIB modules used to represent the relationship between emulated PWE3 services such as Ethernet, ATM, or Frame Relay and the pseudo-wire used to carry that service across the PSN. This layer contains corresponding MIB modules used to mate or adapt those emulated services to the generic pseudo-wire representation these are represented in the "Generic PW MIB" functional block in Figure 13 above. This working group should not produce any MIB modules for managing the general service; rather, it should produce just those modules used to interface or adapt the emulated service onto the PWE3 management framework as shown above. For example, the standard SONET-MIB [RFC3592] is designed and maintained by another working group. The SONET-MIB is designed to manage the native service without PW emulation. However, the PWE3 working group is chartered to produce standards that show how to emulate existing technologies such as SONET/SDH over pseudo-wires rather than reinvent those modules.

8.2.3. Generic PW MIB Modules

The middle layer in the architecture is referred to as the Generic PW Layer. MIBs in this layer are responsible for providing pseudo-wire specific counters and service models used for monitoring and configuration of PWE3 services over any supported PSN service. That is, this layer provides a general model of PWE3 abstraction for management purposes. This MIB is used to interconnect the MIB modules residing in the Service Layer to the PSN VC Layer MIBs (see section 8.2.4).

8.2.4. PSN VC Layer MIB Modules

The third layer in the PWE3 management architecture is referred to as the PSN VC Layer. It is composed of MIBs that are specifically designed to associate pseudo-wires onto those underlying PSN transport technologies that carry the pseudo-wire payloads across the PSN. In general, this means that the MIB module provides a mapping between the emulated service that is mapped to the pseudo-wire via the Service Layer and the Generic PW MIB Layer onto the native PSN service. For example, in the case of MPLS, for example, it is required that the general VC service be mapped into MPLS LSPs via the MPLS-LSR-STD-MIB [RFC3813] or Traffic-Engineered (TE) Tunnels via the MPLS-TE-STD-MIB [RFC3812]. In addition, the MPLS-LDP-STD-MIB [RFC3815] may be used to reveal the MPLS labels that are distributed over the MPLS PSN in order to maintain the PW service. As with the native service MIB modules described earlier, the MIB modules used to manage the native PSN services are produced by other working groups that design and specify the native PSN services. These MIBs should contain the appropriate mechanisms for monitoring and configuring the PSN service that the emulated PWE3 service will function correctly.

8.3. Connection Verification and Traceroute

A connection verification mechanism should be supported by PWs. Connection verification and other alarm mechanisms can alert the operator that a PW has lost its remote connection. The opaque nature of a PW means that it is not possible to specify a generic connection verification or traceroute mechanism that passes this status to the CEs over the PW. If connection verification status of the PW is needed by the CE, it must be mapped to the native connection status method.

For troubleshooting purposes, it is sometimes desirable to know the exact functional path of a PW between PEs. This is provided by the traceroute service of the underlying PSN. The opaque nature of the PW means that this traceroute information is only available within the provider network; e.g., at the PEs.

9. IANA Considerations

IANA considerations will be identified in the PWE3 documents that define the PWE3 encapsulation, control, and management protocols.

10. Security Considerations

PWE3 provides no means of protecting the integrity, confidentiality, or delivery of the native data units. The use of PWE3 can therefore expose a particular environment to additional security threats. Assumptions that might be appropriate when all communicating systems are interconnected via a point-to-point or circuit-switched network may no longer hold when they are interconnected with an emulated wire carried over some types of PSN. It is outside the scope of this specification to fully analyze and review the risks of PWE3, particularly as these risks will depend on the PSN. An example should make the concern clear. A number of IETF standards employ relatively weak security mechanisms when communicating nodes are expected to be connected to the same local area network. The Virtual Router Redundancy Protocol [RFC3768] is one instance. The relatively weak security mechanisms represent a greater vulnerability in an emulated Ethernet connected via a PW.

Exploitation of vulnerabilities from within the PSN may be directed to the PW Tunnel end point so that PW Demultiplexer and PSN tunnel services are disrupted. Controlling PSN access to the PW Tunnel end point is one way to protect against this. By restricting PW Tunnel end point access to legitimate remote PE sources of traffic, the PE may reject traffic that would interfere with the PW Demultiplexing and PSN tunnel services.

Protection mechanisms must also address the spoofing of tunneled PW data. The validation of traffic addressed to the PW Demultiplexer end-point is paramount in ensuring integrity of PW encapsulation. Security protocols such as IPSec [RFC2401] may be used by the PW Demultiplexer Layer in order provide authentication and data integrity of the data between the PW Demultiplexer End-points.

IPSec may provide authentication, integrity, and confidentiality, of data transferred between two PEs. It cannot provide the equivalent services to the native service.

Based on the type of data being transferred, the PW may indicate to the PW Demultiplexer Layer that enhanced security services are required. The PW Demultiplexer Layer may define multiple protection profiles based on the requirements of the PW emulated service. CE-to-CE signaling and control events emulated by the PW and some data types may require additional protection mechanisms. Alternatively,

the PW Demultiplexer Layer may use peer authentication for every PSN packet to prevent spoofed native data units from being sent to the destination CE.

The unlimited transformation capability of the NSP may be perceived as a security risk. In practice the type of operation that the NSP may perform will be limited to those that have been implemented in the data path. A PE designed and managed to best current practice will have controls in place that protect and validate its configuration, and these will be sufficient to ensure that the NSP behaves as expected.

11. Acknowledgements

We thank Sasha Vainshtein for his work on Native Service Processing and advice on bit stream over PW services and Thomas K. Johnson for his work on the background and motivation for PWs.

We also thank Ron Bonica, Stephen Casner, Durai Chinnaiah, Jayakumar Jayakumar, Ghassem Koleyni, Danny McPherson, Eric Rosen, John Rutemiller, Scott Wainner, and David Zelig for their comments and contributions.

12. References

12.1. Normative References

[RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3), RFC 3931, March 2005.

[RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC3592] Tesink, K., "Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type", RFC 3592, September 2003.

[RFC2661] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, September 2000.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

12.2. Informative References

[DVB] EN 300 744 Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television (DVB-T), European Telecommunications Standards Institute (ETSI).

[RFC3815] Cucchiara, J., Sjostrand, H., and J. Luciani, "Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)", RFC 3815, June 2004.

[RFC3813] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB)", RFC 3813, June 2004.

[MPLSIP] Rosen et al, "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", Work in Progress, March 2004.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

[RFC1958] Carpenter, B., "Architectural Principles of the Internet", RFC 1958, June 1996.

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[RFC2022] Armitage, G., "Support for Multicast over UNI 3.0/3.1 based ATM Networks", RFC 2022, November 1996.

[RFC3768] Hinden, R., "Virtual Router Redundancy Protocol (VRRP)", RFC 3768, April 2004.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC3448] Handley, M., Floyd, S., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, January 2003.

[RFC3812] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)", RFC 3812, June 2004.

[RFC3916] Xiao, X., McPherson, D., and P. Pate, Eds, "Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)", RFC 3916, September 2004.

13. Co-Authors

The following are co-authors of this document:

Thomas K. Johnson
Litchfield Communications

Kireeti Kompella
Juniper Networks, Inc.

Andrew G. Malis
Tellabs

Thomas D. Nadeau
Cisco Systems

Tricci So
Caspian Networks

W. Mark Townsley
Cisco Systems

Craig White
Level 3 Communications, LLC.

Lloyd Wood
Cisco Systems

14. Editors' Addresses

Stewart Bryant
Cisco Systems
250, Longwater
Green Park
Reading, RG2 6GB,
United Kingdom

EMail: stbryant@cisco.com

Prayson Pate
Overture Networks, Inc.
507 Airport Boulevard
Morrisville, NC, USA 27560

EMail: prayson.pate@overturenetworks.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

Exhibit 7

Network Working Group
Request for Comments: 4447
Category: Standards Track

L. Martini, Ed.
E. Rosen
Cisco Systems, Inc.
N. El-Aawar
Level 3 Communications, LLC.
T. Smith
Network Appliance, Inc.
G. Heron
Tellabs

April 2006

Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)

Status of This Memo

Copyright Notice

Abstract

Layer 2 services (such as Frame Relay, Asynchronous Transfer Mode, and Ethernet) can be "emulated" over an MPLS backbone by encapsulating the Layer 2 Protocol Data Units (PDU) and transmitting them over "pseudowires". It is also possible to use pseudowires to provide low-rate Time Division Multiplexed and a Synchronous Optical NETworking circuit emulation over an MPLS-enabled network. This document specifies a protocol for establishing and maintaining the pseudowires, using extensions to Label Distribution Protocol (LDP). Procedures for encapsulating Layer 2 PDUs are specified in a set of companion documents.

Table of Contents

1. Introduction

In [FRAME], [ATM], [PPPHDLC], and [ETH], it is explained how to encapsulate a Layer 2 Protocol Data Unit (PDU) for transmission over an MPLS-enabled network. Those documents specify that a "pseudowire header", consisting of a demultiplexor field, will be prepended to the encapsulated PDU. The pseudowire demultiplexor field is prepended before transmitting a packet on a pseudowire. When the packet arrives at the remote endpoint of the pseudowire, the demultiplexor is what enables the receiver to identify the particular pseudowire on which the packet has arrived. To transmit the packet from one pseudowire endpoint to another, the packet may need to travel through a "Packet Switched Network (PSN) tunnel"; this will require that an additional header be prepended to the packet.

Accompanying documents [CEP, SAToP] specify methods for transporting time-division multiplexing (TDM) digital signals (TDM circuit emulation) over a packet-oriented MPLS-enabled network. The transmission system for circuit-oriented TDM signals is the Synchronous Optical Network (SONET)[SDH]/Synchronous Digital Hierarchy (SDH) [ITUG]. To support TDM traffic, which includes voice, data, and private leased-line service, the pseudowires must emulate the circuit characteristics of SONET/SDH payloads. The TDM signals and payloads are encapsulated for transmission over pseudowires. A pseudowire demultiplexor and a PSN tunnel header is prepended to this encapsulation.

[SAToP] describes methods for transporting low-rate time-division multiplexing (TDM) digital signals (TDM circuit emulation) over PSNs, while [CEP] similarly describes transport of high-rate TDM (SONET/SDH). To support TDM traffic, the pseudowires must emulate the circuit characteristics of the original T1, E1, T3, E3, SONET, or SDH signals. [SAToP] does this by encapsulating an arbitrary but constant amount of the TDM data in each packet, and the other methods encapsulate TDM structures.

In this document, we specify the use of the MPLS Label Distribution Protocol, LDP [RFC3036], as a protocol for setting up and maintaining the pseudowires. In particular, we define new TLVs, FEC elements, parameters, and codes for LDP, which enable LDP to identify pseudowires and to signal attributes of pseudowires. We specify how a pseudowire endpoint uses these TLVs in LDP to bind a demultiplexor field value to a pseudowire, and how it informs the remote endpoint of the binding. We also specify procedures for reporting pseudowire status changes, for passing additional information about the pseudowire as needed, and for releasing the bindings.

In the protocol specified herein, the pseudowire demultiplexor field is an MPLS label. Thus, the packets that are transmitted from one end of the pseudowire to the other are MPLS packets, which must be transmitted through an MPLS tunnel. However, if the pseudowire endpoints are immediately adjacent and penultimate hop popping behavior is in use, the MPLS tunnel may not be necessary. Any sort of PSN tunnel can be used, as long as it is possible to transmit MPLS packets through it. The PSN tunnel can itself be an MPLS LSP, or any other sort of tunnel that can carry MPLS packets. Procedures for setting up and maintaining the MPLS tunnels are outside the scope of this document.

This document deals only with the setup and maintenance of point-to-point pseudowires. Neither point-to-multipoint nor multipoint-to-point pseudowires are discussed.

QoS-related issues are not discussed in this document. The following two figures describe the reference models that are derived from [RFC3985] to support the PW emulated services.

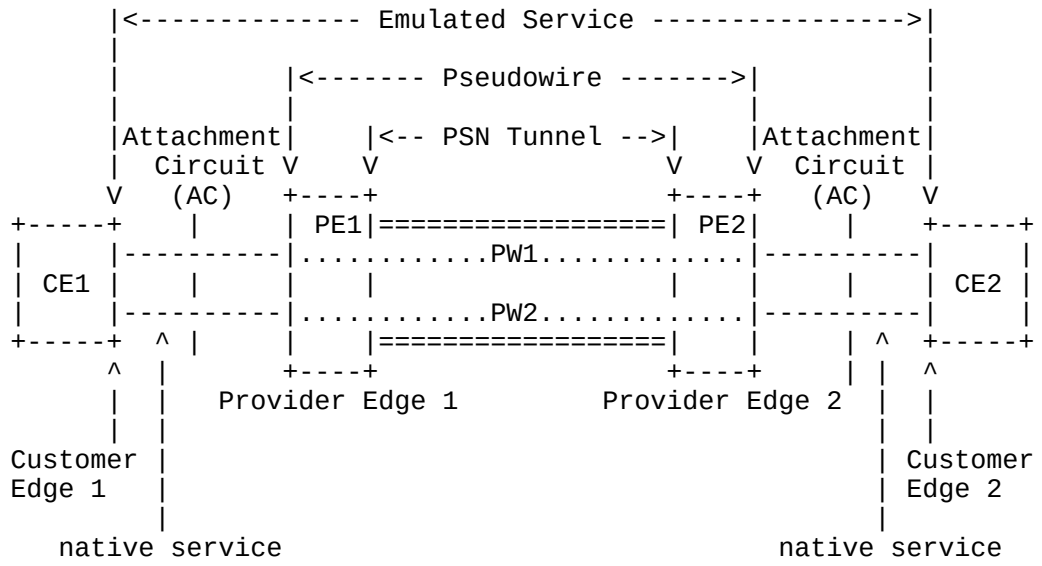


Figure 1: PWE3 Reference Model

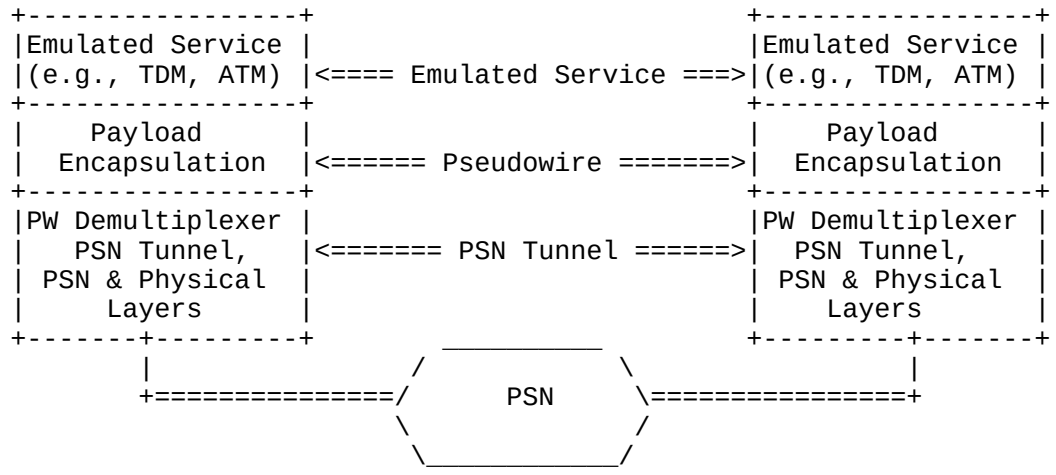


Figure 2: PWE3 Protocol Stack Reference Model

For the purpose of this document, PE1 will be defined as the ingress router, and PE2 as the egress router. A layer 2 PDU will be received at PE1, encapsulated at PE1, transported and decapsulated at PE2, and transmitted out of PE2.

2. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. The Pseudowire Label

Suppose that it is desired to transport Layer 2 PDUs from ingress LSR PE1 to egress LSR PE2, across an intervening MPLS-enabled network. We assume that there is an MPLS tunnel from PE1 to PE2. That is, we assume that PE1 can cause a packet to be delivered to PE2 by encapsulating the packet in an "MPLS tunnel header" and sending the result to one of its adjacencies. The MPLS tunnel is an MPLS Label Switched Path (LSP); thus, putting on an MPLS tunnel encapsulation is a matter of pushing on an MPLS label.

We presuppose that a large number of pseudowires can be carried through a single MPLS tunnel. Thus, it is never necessary to maintain state in the network core for individual pseudowires. We do not presuppose that the MPLS tunnels are point to point; although the pseudowires are point to point, the MPLS tunnels may be multipoint to point. We do not presuppose that PE2 will even be able to determine the MPLS tunnel through which a received packet was transmitted.

(For example, if the MPLS tunnel is an LSP and penultimate hop popping is used, when the packet arrives at PE2, it will contain no information identifying the tunnel.)

When PE2 receives a packet over a pseudowire, it must be able to determine that the packet was in fact received over a pseudowire, and it must be able to associate that packet with a particular pseudowire. PE2 is able to do this by examining the MPLS label that serves as the pseudowire demultiplexor field shown in Figure 2. Call this label the "PW label".

When PE1 sends a Layer 2 PDU to PE2, it creates an MPLS packet by adding the PW label to the packet, thus creating the first entry of the label stack. If the PSN tunnel is an MPLS LSP, the PE1 pushes another label (the tunnel label) onto the packet as the second entry of the label stack. The PW label is not visible again until the MPLS packet reaches PE2. PE2's disposition of the packet is based on the PW label.

If the payload of the MPLS packet is, for example, an ATM AAL5 PDU, the PW label will generally correspond to a particular ATM VC at PE2. That is, PE2 needs to be able to infer from the PW label the outgoing interface and the VPI/VCI value for the AAL5 PDU. If the payload is a Frame Relay PDU, then PE2 needs to be able to infer from the PW label the outgoing interface and the DLCI value. If the payload is an Ethernet frame, then PE2 needs to be able to infer from the PW label the outgoing interface, and perhaps the VLAN identifier. This process is uni-directional and will be repeated independently for bi-directional operation. It is REQUIRED that the same PW ID and PW type be assigned for a given circuit in both directions. The group ID (see below) MUST NOT be required to match in both directions. The transported frame MAY be modified when it reaches the egress router. If the header of the transported Layer 2 frame is modified, this MUST be done at the egress LSR only. Note that the PW label must always be at the bottom of the packet's label stack, and labels MUST be allocated from the per-platform label space.

This document does not specify a method for distributing the MPLS tunnel label or any other labels that may appear above the PW label on the stack. Any acceptable method of MPLS label distribution will do. This document specifies a protocol for assigning and distributing the PW label. This protocol is LDP, extended as specified in the remainder of this document. An LDP session must be set up between the pseudowire endpoints. LDP MUST be used in its "downstream unsolicited" mode. LDP's "liberal label retention" mode SHOULD be used.

In addition to the protocol specified herein, static assignment of PW labels may be used, and implementations of this protocol SHOULD provide support for static assignment.

This document specifies all the procedures necessary to set up and maintain the pseudowires needed to support "unswitched" point-to-point services, where each endpoint of the pseudowire is provisioned with the identify of the other endpoint. There are also protocol mechanisms specified herein that can be used to support switched services and other provisioning models. However, the use of the protocol mechanisms to support those other models and services is not described in this document.

4. Details Specific to Particular Emulated Services

4.1. IP Layer 2 Transport

This mode carries IP packets over a pseudowire. The encapsulation used is according to [RFC3032]. The PW control word MAY be inserted between the MPLS label stack and the IP payload. The encapsulation of the IP packets for forwarding on the attachment circuit is implementation specific, is part of the native service processing (NSP) function [RFC3985], and is outside the scope of this document.

5. LDP

The PW label bindings are distributed using the LDP downstream unsolicited mode described in [RFC3036]. The PEs will establish an LDP session using the Extended Discovery mechanism described in [LDP, sections 2.4.2 and 2.5].

An LDP Label Mapping message contains an FEC TLV, a Label TLV, and zero or more optional parameter TLVs.

The FEC TLV is used to indicate the meaning of the label. In the current context, the FEC TLV would be used to identify the particular pseudowire that a particular label is bound to. In this specification, we define two new FEC TLVs to be used for identifying pseudowires. When setting up a particular pseudowire, only one of these FEC TLVs is used. The one to be used will depend on the particular service being emulated and on the particular provisioning model being supported.

LDP allows each FEC TLV to consist of a set of FEC elements. For setting up and maintaining pseudowires, however, each FEC TLV MUST contain exactly one FEC element.

The LDP base specification has several kinds of label TLVs, including the Generic Label TLV, as specified in [RFC3036], section 3.4.2.1. For setting up and maintaining pseudowires, the Generic Label TLV MUST be used.

5.1. LDP Extensions

This document specifies no new LDP messages.

This document specifies the following new TLVs to be used with LDP:

TLV	Specified in Section	Defined for Message
PW Status TLV	5.4.2	Notification
PW Interface Parameters TLV	5.3.2.1	FEC
PW Grouping ID TLV	5.3.2.2	FEC

Additionally, the following new FEC element types are defined:

FEC Element Type	Specified in Section	Defined for Message
0x80	5.2	FEC
0x81	5.3	FEC

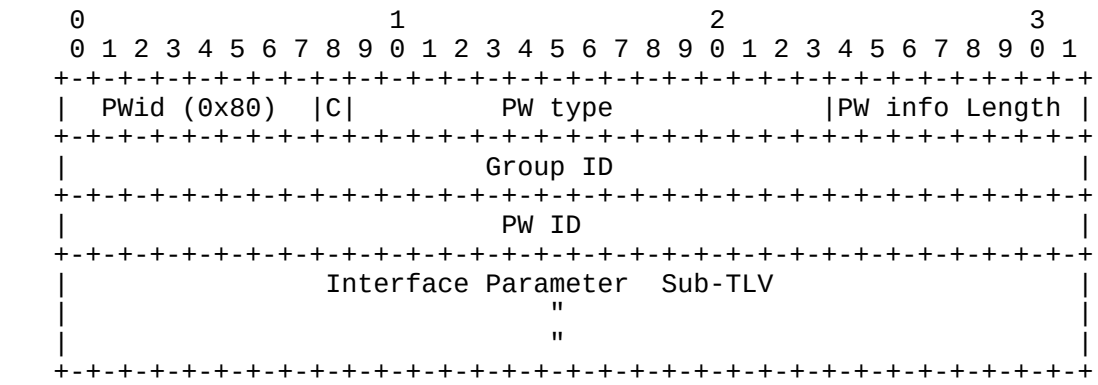
The following new LDP error codes are also defined:

Status Code	Specified in Section
"Illegal C-Bit"	6.1
"Wrong C-Bit"	6.2
"Incompatible bit-rate"	[CEP]
"CEP/TDM mis-configuration"	[CEP]
"PW status"	5.4.2
"Unassigned/Unrecognized TAI"	5.3.3
"Generic Misconfiguration Error"	[SAToP]
"Label Withdraw PW Status Method Not Supported"	5.4.1

5.2. The PWid FEC Element

The PWid FEC element may be used whenever both pseudowire endpoints have been provisioned with the same 32-bit identifier for the pseudowire.

For this purpose, a new type of FEC element is defined. The FEC element type is 0x80 and is defined as follows:



- PW type

  A 15-bit quantity containing a value that represents the type of PW. Assigned values are specified in "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)" [IANA].

- Control word bit (C)

  The bit (C) is used to flag the presence of a control word as follows:

  C = 1 Control word present on this PW.
  C = 0 No control word present on this PW.

  Please see the section "C-Bit Handling Procedures" for further explanation.

- PW information length

  Length of the PW ID field and the interface parameters sub-TLV in octets. If this value is 0, then it references all PWs using the specified group ID, and there is no PW ID present; nor are there any interface parameter sub-TLVs.

- Group ID

  An arbitrary 32-bit value that represents a group of PWs that is used to create groups in the PW space. The group ID is intended to be used as a port index, or a virtual tunnel index. To simplify configuration, a particular PW ID at ingress could be part of the virtual tunnel for transport to the egress router.

The Group ID is very useful for sending wild card label withdrawals, or PW wild card status notification messages to remote PEs upon physical port failure.

- PW ID

  A non-zero 32-bit connection ID that, together with the PW type, identifies a particular PW. Note that the PW ID and the PW type MUST be the same at both endpoints.

- Interface Parameter Sub-TLV

  This variable-length TLV is used to provide interface-specific parameters, such as attachment circuit MTU.

  Note that as the "interface parameter sub-TLV" is part of the FEC, the rules of LDP make it impossible to change the interface parameters once the pseudowire has been set up. Thus, the interface parameters field must not be used to pass information, such as status information, that may change during the life of the pseudowire. Optional parameter TLVs should be used for that purpose.

Using the PWid FEC, each of the two pseudowire endpoints independently initiates the setup of a unidirectional LSP. An outgoing LSP and an incoming LSP are bound together into a single pseudowire if they have the same PW ID and PW type.

5.3. The Generalized PWid FEC Element

The PWid FEC element can be used if a unique 32-bit value has been assigned to the PW, and if each endpoint has been provisioned with that value. The Generalized PWid FEC element requires that the PW endpoints be uniquely identified; the PW itself is identified as a pair of endpoints. In addition, the endpoint identifiers are structured to support applications where the identity of the remote endpoints needs to be auto-discovered rather than statically configured.

The "Generalized PWid FEC Element" is FEC type 0x81.

The Generalized PWid FEC Element does not contain anything corresponding to the "Group ID" of the PWid FEC element. The functionality of the "Group ID" is provided by a separate optional LDP TLV, the "PW Grouping TLV", described below. The Interface Parameters field of the PWid FEC element is also absent; its functionality is replaced by the optional Interface Parameters TLV, described below.

5.3.1. Attachment Identifiers

As discussed in [RFC3985], a pseudowire can be thought of as connecting two "forwarders". The protocol used to set up a pseudowire must allow the forwarder at one end of a pseudowire to identify the forwarder at the other end. We use the term "attachment identifier", or "AI", to refer to the field that the protocol uses to identify the forwarders. In the PWid FEC, the PWid field serves as the AI. In this section, we specify a more general form of AI that is structured and of variable length.

Every Forwarder in a PE must be associated with an Attachment Identifier (AI), either through configuration or through some algorithm. The Attachment Identifier must be unique in the context of the PE router in which the Forwarder resides. The combination <PE router IP address, AI> must be globally unique.

It is frequently convenient to regard a set of Forwarders as being members of a particular "group", where PWs may only be set up among members of a group. In such cases, it is convenient to identify the Forwarders relative to the group, so that an Attachment Identifier would consist of an Attachment Group Identifier (AGI) plus an Attachment Individual Identifier (AII).

An Attachment Group Identifier may be thought of as a VPN-id, or a VLAN identifier, some attribute that is shared by all the Attachment PWs (or pools thereof) that are allowed to be connected.

The details of how to construct the AGI and AII fields identifying the pseudowire endpoints are outside the scope of this specification. Different pseudowire applications, and different provisioning models, will require different sorts of AGI and AII fields. The specification of each such application and/or model must include the rules for constructing the AGI and AII fields.

As previously discussed, a (bidirectional) pseudowire consists of a pair of unidirectional LSPs, one in each direction. If a particular pseudowire connects PE1 with PE2, the PW direction from PE1 to PE2 can be identified as:

<PE1, <AGI, AII1>, PE2, <AGI, AII2>>,

The PW direction from PE2 to PE1 can be identified by:

<PE2, <AGI, AII2>, PE1, <AGI, AII1>>.

Note that the AGI must be the same at both endpoints, but the AII will in general be different at each endpoint. Thus, from the perspective of a particular PE, each pseudowire has a local or "Source AII", and a remote or "Target AII". The pseudowire setup protocol can carry all three of these quantities:

- Attachment Group Identifier (AGI)

- Source Attachment Individual Identifier (SAII)

- Target Attachment Individual Identifier (TAII)

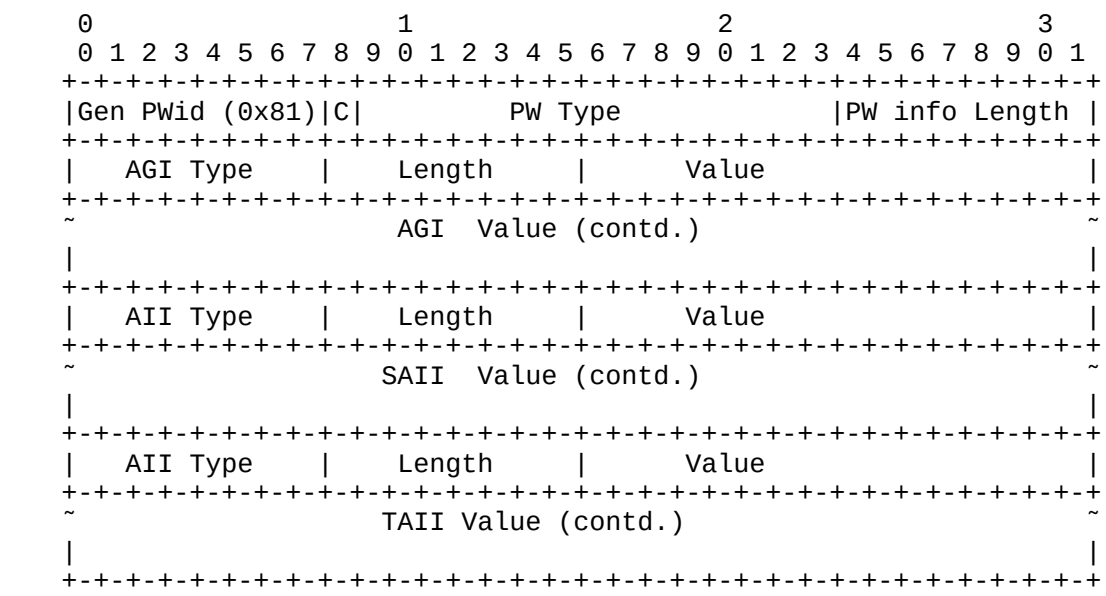
If the AGI is non-null, then the Source AI (SAI) consists of the AGI together with the SAII, and the Target AI (TAI) consists of the TAII together with the AGI. If the AGI is null, then the SAII and TAII are the SAI and TAI, respectively.

The interpretation of the SAI and TAI is a local matter at the respective endpoint.

The association of two unidirectional LSPs into a single bidirectional pseudowire depends on the SAI and the TAI. Each application and/or provisioning model that uses the Generalized ID FEC element must specify the rules for performing this association.

5.3.2. Encoding the Generalized ID FEC Element

FEC element type 0x81 is used. The FEC element is encoded as follows:



This document does not specify the AII and AGI type field values; specification of the type field values to be used for a particular application is part of the specification of that application. IANA has assigned these values using the method defined in the [IANA] document.

The SAII, TAII, and AGI are simply carried as octet strings. The length byte specifies the size of the Value field. The null string can be sent by setting the length byte to 0. If a particular application does not need all three of these sub-elements, it MUST send all the sub-elements but set the length to 0 for the unused sub-elements.

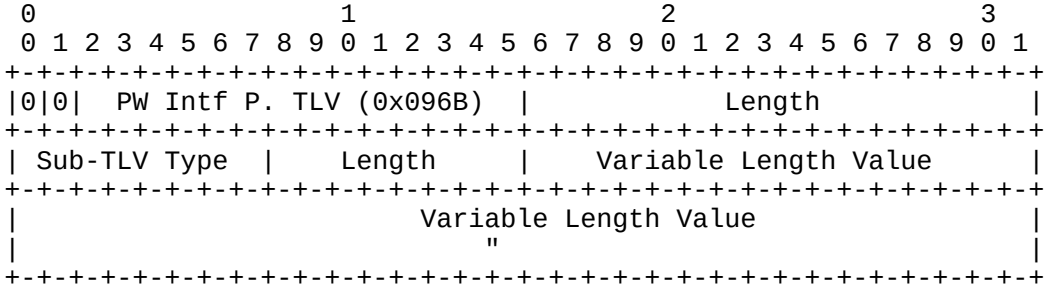
The PW information length field contains the length of the SAII, TAII, and AGI, combined in octets. If this value is 0, then it references all PWs using the specified grouping ID. In this case, there are no other FEC element fields (AGI, SAII, etc.) present, nor any interface parameters TLVs.

Note that the interpretation of a particular field as AGI, SAII, or TAII depends on the order of its occurrence. The type field identifies the type of the AGI, SAII, or TAII. When comparing two

occurrences of an AGI (or SAII or TAII), the two occurrences are considered identical if the type, length, and value fields of one are identical, respectively, to those of the other.

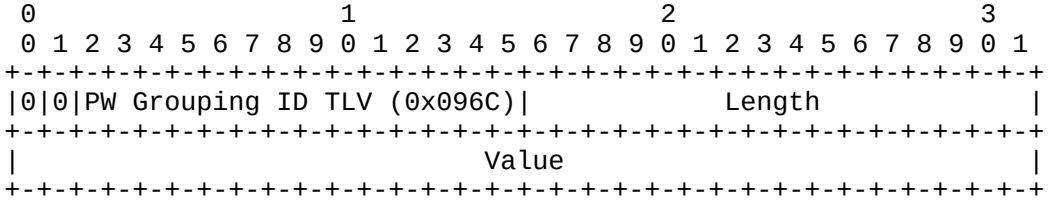
5.3.2.1. Interface Parameters TLV

This TLV MUST only be used when sending the Generalized PW FEC. It specifies interface-specific parameters. Specific parameters, when applicable, MUST be used to validate that the PEs and the ingress and egress ports at the edges of the circuit have the necessary capabilities to interoperate with each other.



A more detailed description of this field can be found in the section "Interface Parameters Sub-TLV", below.

5.3.2.2. PW Grouping TLV



The PW Grouping ID is an arbitrary 32-bit value that represents an arbitrary group of PWs. It is used to create group PWs; for example, a PW Grouping ID can be used as a port index and assigned to all PWs that lead to that port. Use of the PW Grouping ID enables one to send "wild card" label withdrawals, or "wild card" status notification messages, to remote PEs upon physical port failure.

Note Well: The PW Grouping ID is different from, and has no relation to, the Attachment Group Identifier.

The PW Grouping ID TLV is not part of the FEC and will not be advertised except in the PW FEC advertisement. The advertising PE

MAY use the wild card withdraw semantics, but the remote PEs MUST implement support for wild card messages. This TLV MUST only be used when sending the Generalized PW ID FEC.

To issue a wildcard command (status or withdraw):

- Set the PW Info Length to 0 in the Generalized ID FEC Element.

- Send only the PW Grouping ID TLV with the FEC (no AGI/SAII/TAII is sent).

5.3.3. Signaling Procedures

In order for PE1 to begin signaling PE2, PE1 must know the address of the remote PE2, and a TAI. This information may have been configured at PE1, or it may have been learned dynamically via some autodiscovery procedure.

The egress PE (PE1), which has knowledge of the ingress PE, initiates the setup by sending a Label Mapping Message to the ingress PE (PE2). The Label Mapping message contains the FEC TLV, carrying the Generalized PWid FEC Element (type 0x81). The Generalized PWid FEC element contains the AGI, SAII, and TAII information.

Next, when PE2 receives such a Label Mapping message, PE2 interprets the message as a request to set up a PW whose endpoint (at PE2) is the Forwarder identified by the TAI. From the perspective of the signaling protocol, exactly how PE2 maps AIs to Forwarders is a local matter. In some Virtual Private Wire Services (VPWS) provisioning models, the TAI might, for example, be a string that identifies a particular Attachment Circuit, such as "ATM3VPI4VCI5", or it might, for example, be a string, such as "Fred", that is associated by configuration with a particular Attachment Circuit. In VPLS, the AGI could be a VPN-id, identifying a particular VPLS instance.

If PE2 cannot map the TAI to one of its Forwarders, then PE2 sends a Label Release message to PE1, with a Status Code of "Unassigned/Unrecognized TAI", and the processing of the Label Mapping message is complete.

The FEC TLV sent in a Label Release message is the same as the FEC TLV received in the Label Mapping being released (but without the interface parameter TLV). More generally, the FEC TLV is the same in all LDP messages relating to the same PW. In a Label Release, this means that the SAII is the remote peer’s AII and the TAII is the sender’s local AII.

If the Label Mapping Message has a valid TAI, PE2 must decide whether to accept it. The procedures for so deciding will depend on the particular type of Forwarder identified by the TAI. Of course, the Label Mapping message may be rejected due to standard LDP error conditions as detailed in [RFC3036].

If PE2 decides to accept the Label Mapping message, then it has to make sure that a PW LSP is set up in the opposite (PE1-->PE2) direction. If it has already signaled for the corresponding PW LSP in that direction, nothing more needs to be done. Otherwise, it must initiate such signaling by sending a Label Mapping message to PE1. This is very similar to the Label Mapping message PE2 received, but the SAI and TAI are reversed.

Thus, a bidirectional PW consists of two LSPs, where the FEC of one has the SAII and TAII reversed with respect to the FEC of the other.

5.4. Signaling of Pseudowire Status

5.4.1. Use of Label Mappings Messages

The PEs MUST send Label Mapping Messages to their peers as soon as the PW is configured and administratively enabled, regardless of the attachment circuit state. The PW label should not be withdrawn unless the operator administratively configures the pseudowire down (or the PW configuration is deleted entirely). Using the procedures outlined in this section, a simple label withdraw method MAY also be supported as a legacy means of signaling PW status and AC status. In any case, if the label-to-PW binding is not available, the PW MUST be considered in the down state.

Once the PW status negotiation procedures are completed, if they result in the use of the label withdraw method for PW status communication, and this method is not supported by one of the PEs, then that PE must send a Label Release Message to its peer with the following error:

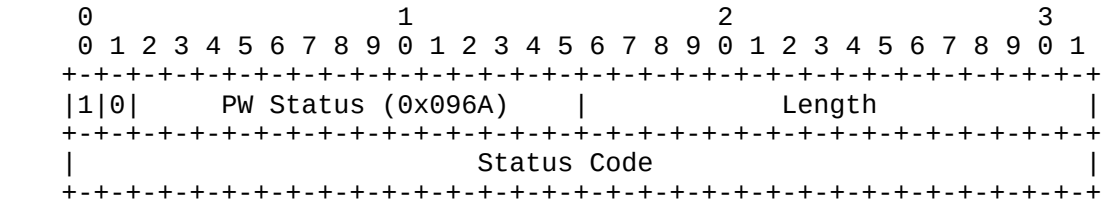
"Label Withdraw PW Status Method Not Supported"

If the label withdraw method for PW status communication is selected for the PW, it will result in the Label Mapping Message being advertised only if the attachment circuit is active. The PW status signaling procedures described in this section MUST be fully implemented.

5.4.2. Signaling PW Status

The PE devices use an LDP TLV to indicate status to their remote peers. This PW Status TLV contains more information than the alternative simple Label Withdraw message.

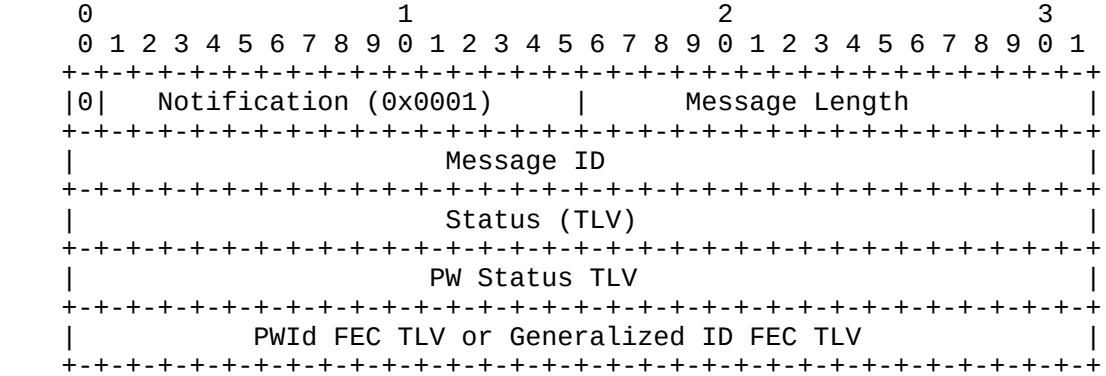
The format of the PW Status TLV is:



The status code is a 4-octet bit field as specified in the PW IANA Allocations document [IANA]. The length specifies the length of the Status Code field in octets (equal to 4).

Each bit in the status code field can be set individually to indicate more than a single failure at once. Each fault can be cleared by sending an appropriate Notification message in which the respective bit is cleared. The presence of the lowest bit (PW Not Forwarding) acts only as a generic failure indication when there is a link-down event for which none of the other bits apply.

The Status TLV is transported to the remote PW peer via the LDP Notification message. The general format of the Notification Message is:



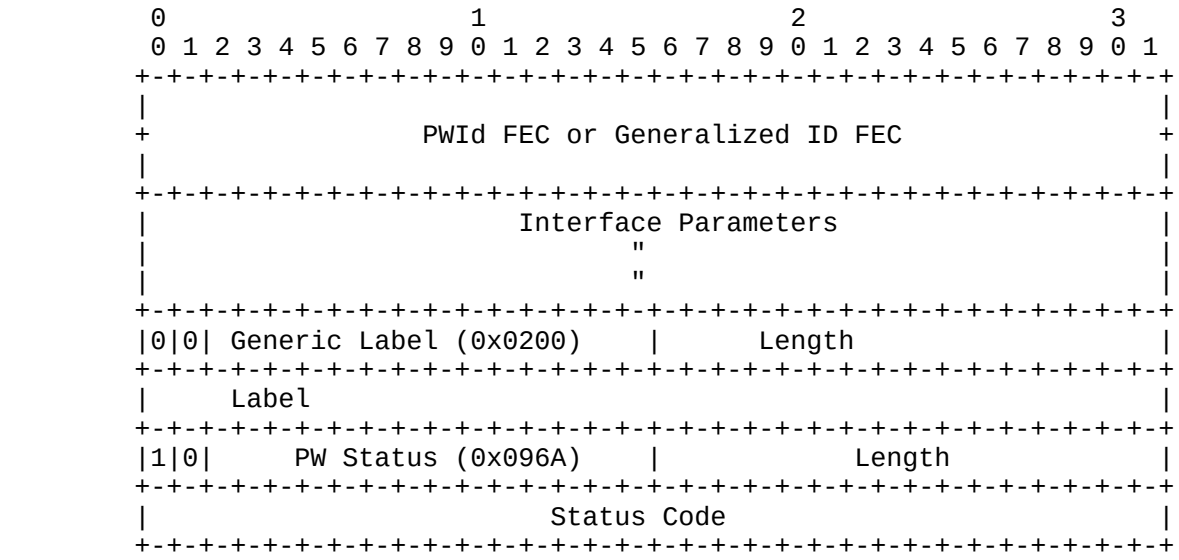
The Status TLV status code is set to 0x00000028, "PW status", to indicate that PW status follows. Since this notification does not refer to any particular message, the Message Id and Message Type fields are set to 0.

The PW FEC TLV SHOULD not include the interface parameter sub-TLVs, as they are ignored in the context of this message. When a PE's attachment circuit encounters an error, use of the PW Notification Message allows the PE to send a single "wild card" status message, using a PW FEC TLV with only the group ID set, to denote this change in status for all affected PW connections. This status message contains either the PW FEC TLV with only the group ID set, or else it contains the Generalized FEC TLV with only the PW Grouping ID TLV.

As mentioned above, the Group ID field of the PWid FEC element, or the PW Grouping ID TLV used with the Generalized ID FEC element, can be used to send a status notification for all arbitrary sets of PWs. This procedure is OPTIONAL, and if it is implemented, the LDP Notification message should be as follows: If the PWid FEC element is used, the PW information length field is set to 0, the PW ID field is not present, and the interface parameter sub-TLVs are not present. If the Generalized FEC element is used, the AGI, SAII, and TAII are not present, the PW information length field is set to 0, the PW Grouping ID TLV is included, and the Interface Parameters TLV is omitted. For the purpose of this document, this is called the "wild card PW status notification procedure", and all PEs implementing this design are REQUIRED to accept such a notification message but are not required to send it.

5.4.3. Pseudowire Status Negotiation Procedures

When a PW is first set up, the PEs MUST attempt to negotiate the usage of the PW status TLV. This is accomplished as follows: A PE that supports the PW Status TLV MUST include it in the initial Label Mapping message following the PW FEC and the interface parameter sub-TLVs. The PW Status TLV will then be used for the lifetime of the pseudowire. This is shown in the following diagram:



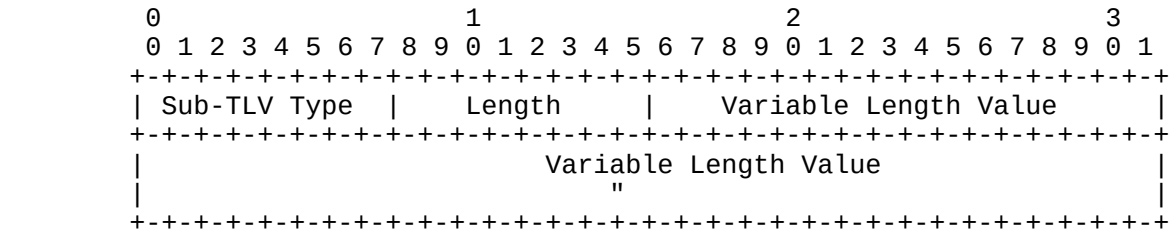
If a PW Status TLV is included in the initial Label Mapping message for a PW, then if the Label Mapping message from the remote PE for that PW does not include a PW status TLV, or if the remote PE does not support the PW Status TLV, the PW will revert to the label withdraw method of signaling PW status. Note that if the PW Status TLV is not supported by the remote peer, the peer will automatically ignore it, since the I (ignore) bit is set in the TLV. The PW Status TLV, therefore, will not be present in the corresponding FEC advertisement from the remote LDP peer, which results in exactly the above behavior.

If the PW Status TLV is not present following the FEC TLV in the initial PW Label Mapping message received by a PE, then the PW Status TLV will not be used, and both PEs supporting the pseudowire will revert to the label withdraw procedure for signaling status changes.

If the negotiation process results in the usage of the PW status TLV, then the actual PW status is determined by the PW status TLV that was sent within the initial PW Label Mapping message. Subsequent updates of PW status are conveyed through the notification message.

5.5. Interface Parameters Sub-TLV

This field specifies interface-specific parameters. When applicable, it MUST be used to validate that the PEs and the ingress and egress ports at the edges of the circuit have the necessary capabilities to interoperate with each other. The field structure is defined as follows:



The interface parameter sub-TLV type values are specified in "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)" [IANA].

The Length field is defined as the length of the interface parameter including the parameter id and length field itself. Processing of the interface parameters should continue when unknown interface parameters are encountered, and they MUST be silently ignored.

- Interface MTU sub-TLV type

  A 2-octet value indicating the MTU in octets. This is the Maximum Transmission Unit, excluding encapsulation overhead, of the egress packet interface that will be transmitting the decapsulated PDU that is received from the MPLS-enabled network. This parameter is applicable only to PWs transporting packets and is REQUIRED for these PW types. If this parameter does not match in both directions of a specific PW, that PW MUST NOT be enabled.

- Optional Interface Description string sub-TLV type

  This arbitrary, and OPTIONAL, interface description string is used to send a human-readable administrative string describing the interface to the remote. This parameter is OPTIONAL and is applicable to all PW types. The interface description parameter string length is variable and can be from 0 to 80 octets. Human-readable text MUST be provided in the UTF-8 charset using the Default Language [RFC2277].

6. Control Word

6.1. PW Types for Which the Control Word is REQUIRED

The Label Mapping messages that are sent in order to set up these PWs MUST have c=1. When a Label Mapping message for a PW of one of these types is received and c=0, a Label Release message MUST be sent, with an "Illegal C-bit" status code. In this case, the PW will not be enabled.

6.2. PW Types for Which the Control Word is NOT Mandatory

If a system is capable of sending and receiving the control word on PW types for which the control word is not mandatory, then each such PW endpoint MUST be configurable with a parameter that specifies whether the use of the control word is PREFERRED or NOT PREFERRED. For each PW, there MUST be a default value of this parameter. This specification does NOT state what the default value should be.

If a system is NOT capable of sending and receiving the control word on PW types for which the control word is not mandatory, then it behaves exactly as if it were configured for the use of the control word to be NOT PREFERRED.

If a Label Mapping message for the PW has already been received but no Label Mapping message for the PW has yet been sent, then the procedure is as follows:

-i. If the received Label Mapping message has c=0, send a Label Mapping message with c=0; the control word is not used.

-ii. If the received Label Mapping message has c=1 and the PW is locally configured such that the use of the control word is preferred, then send a Label Mapping message with c=1; the control word is used.

-iii. If the received Label Mapping message has c=1 and the PW is locally configured such that the use of the control word is not preferred or the control word is not supported, then act as if no Label Mapping message for the PW had been received (i.e., proceed to the next paragraph).

If a Label Mapping message for the PW has not already been received (or if the received Label Mapping message had c=1 and either local configuration says that the use of the control word is not preferred or the control word is not supported), then send a Label Mapping message in which the c bit is set to correspond to the locally configured preference for use of the control word. (That is, set c=1 if locally configured to prefer the control word, and set c=0 if locally configured to prefer not to use the control word or if the control word is not supported).

The next action depends on what control message is next received for that PW. The possibilities are as follows:

-i. A Label Mapping message with the same c bit value as specified in the Label Mapping message that was sent. PW setup is now complete, and the control word is used if c=1 but is not used if c=0.

-ii. A Label Mapping message with c=1, but the Label Mapping message that was sent has c=0. In this case, ignore the received Label Mapping message and continue to wait for the next control message for the PW.

-iii. A Label Mapping message with c=0, but the Label Mapping message that was sent has c=1. In this case, send a Label Withdraw message with a "Wrong C-bit" status code, followed by a Label Mapping message that has c=0. PW setup is now complete, and the control word is not used.

-iv. A Label Withdraw message with the "Wrong c-bit" status code. Treat as a normal Label Withdraw, but do not respond. Continue to wait for the next control message for the PW.

If at any time after a Label Mapping message has been received a corresponding Label Withdraw or Release is received, the action taken is the same as for any Label Withdraw or Release that might be received at any time.

If both endpoints prefer the use of the control word, this procedure will cause it to be used. If either endpoint prefers not to use the control word or does not support the control word, this procedure will cause it not to be used. If one endpoint prefers to use the control word but the other does not, the one that prefers not to use it is has no extra protocol to execute; it just waits for a Label Mapping message that has c=0.

The diagram in Appendix A illustrates the above procedure.

6.3. LDP Label Withdrawal Procedures

As mentioned above, the Group ID field of the PWid FEC element, or the PW Grouping ID TLV used with the Generalized ID FEC element, can be used to withdraw all PW labels associated with a particular PW group. This procedure is OPTIONAL, and if it is implemented, the LDP Label Withdraw message should be as follows: If the PWid FEC element is used, the PW information length field is set to 0, the PW ID field is not present, the interface parameter sub-TLVs are not present, and the Label TLV is not present.

If the Generalized FEC element is used, the AGI, SAII, and TAII are not present, the PW information length field is set to 0, the PW Grouping ID TLV is included, the Interface Parameters TLV is not present, and the Label TLV is not present. For the purpose of this document, this is called the "wild card withdraw procedure", and all PEs implementing this design are REQUIRED to accept such withdrawn message but are not required to send it. Note that the PW Grouping ID TLV only applies to PWs using the Generalized ID FEC element, while the Group ID only applies to PWid FEC element.

The interface parameter sub-TLVs, or TLV, MUST NOT be present in any LDP PW Label Withdraw or Label Release message. A wild card Label Release message MUST include only the group ID, or Grouping ID TLV. A Label Release message initiated by a PE router must always include the PW ID.

6.4. Sequencing Considerations

In the case where the router considers the sequence number field in the control word, it is important to note the following details when advertising labels.

6.4.1. Label Advertisements

After a label has been withdrawn by the output router and/or released by the input router, care must be taken not to advertise (re-use) the same released label until the output router can be reasonably certain that old packets containing the released label no longer persist in the MPLS-enabled network.

This precaution is required to prevent the imposition router from restarting packet forwarding with a sequence number of 1 when it receives a Label Mapping message that binds the same FEC to the same label if there are still older packets in the network with a sequence number between 1 and 32768. For example, if there is a packet with sequence number=n, where n is in the interval [1,32768] traveling through the network, it would be possible for the disposition router to receive that packet after it re-advertises the label. Since the label has been released by the imposition router, the disposition router SHOULD be expecting the next packet to arrive with a sequence number of 1. Receipt of a packet with a sequence number equal to n will result in n packets potentially being rejected by the disposition router until the imposition router imposes a sequence number of n+1 into a packet. Possible methods to avoid this are for the disposition router always to advertise a different PW label, or for the disposition router to wait for a sufficient time before

attempting to re-advertise a recently released label. This is only an issue when sequence number processing is enabled at the disposition router.

6.4.2. Label Release

In situations where the imposition router wants to restart forwarding of packets with sequence number 1, the router shall 1) send to the disposition router a Label Release Message, and 2) send to the disposition router a Label Request message. When sequencing is supported, advertisement of a PW label in response to a Label Request message MUST also consider the issues discussed in the section on Label Advertisements.

7. IANA Considerations

7.1. LDP TLV TYPE

This document uses several new LDP TLV types; IANA already maintains a registry of name "TLV TYPE NAME SPACE" defined by RFC 3036. The following values are suggested for assignment:

TLV type	Description
0x096A	PW Status TLV
0x096B	PW Interface Parameters TLV
0x096C	Group ID TLV

7.2. LDP Status Codes

This document uses several new LDP status codes; IANA already maintains a registry of name "STATUS CODE NAME SPACE" defined by RFC 3036. The following values are suggested for assignment:

Range/Value	E	Description	Reference
0x00000024	0	Illegal C-Bit	[RFC4447]
0x00000025	0	Wrong C-Bit	[RFC4447]
0x00000026	0	Incompatible bit-rate	[RFC4447]
0x00000027	0	CEP-TDM mis-configuration	[RFC4447]
0x00000028	0	PW Status	[RFC4447]
0x00000029	0	Unassigned/Unrecognized TAI	[RFC4447]
0x0000002A	0	Generic Misconfiguration Error	[RFC4447]
0x0000002B	0	Label Withdraw PW Status Method	[RFC4447]

7.3. FEC Type Name Space

This document uses two new FEC element types, 0x80 and 0x81, from the registry "FEC Type Name Space" for the Label Distribution Protocol (LDP RFC 3036).

8. Security Considerations

This document specifies the LDP extensions that are needed for setting up and maintaining pseudowires. The purpose of setting up pseudowires is to enable Layer 2 frames to be encapsulated in MPLS and transmitted from one end of a pseudowire to the other. Therefore, we treat the security considerations for both the data plane and the control plane.

8.1. Data-Plane Security

With regard to the security of the data plane, the following areas must be considered:

- MPLS PDU inspection
- MPLS PDU spoofing
- MPLS PDU alteration
- MPLS PSN protocol security
- Access Circuit security
- Denial-of-service prevention on the PE routers

When an MPLS PSN is used to provide pseudowire service, there is a perception that security MUST be at least equal to the currently deployed Layer 2 native protocol networks that the MPLS/PW network combination is emulating. This means that the MPLS-enabled network SHOULD be isolated from outside packet insertion in such a way that it SHOULD not be possible to insert an MPLS packet into the network directly. To prevent unwanted packet insertion, it is also important to prevent unauthorized physical access to the PSN, as well as unauthorized administrative access to individual network elements.

As mentioned above, as MPLS enabled network should not accept MPLS packets from its external interfaces (i.e., interfaces to CE devices or to other providers' networks) unless the top label of the packet was legitimately distributed to the system from which the packet is being received. If the packet's incoming interface leads to a different SP (rather than to a customer), an appropriate trust relationship must also be present, including the trust that the other SP also provides appropriate security measures.

The three main security problems faced when using an MPLS-enabled network to transport PWs are spoofing, alteration, and inspection.

First, there is a possibility that the PE receiving PW PDUs will get a PDU that appears to be from the PE transmitting the PW into the PSN, but that was not actually transmitted by the PE originating the PW. (That is, the specified encapsulations do not by themselves enable the decapsulator to authenticate the encapsulator.) A second problem is the possibility that the PW PDU will be altered between the time it enters the PSN and the time it leaves the PSN (i.e., the specified encapsulations do not by themselves assure the decapsulator of the packet's integrity.) A third problem is the possibility that the PDU's contents will be seen while the PDU is in transit through the PSN (i.e., the specification encapsulations do not ensure privacy.) How significant these issues are in practice depends on the security requirements of the applications whose traffic is being sent through the tunnel, and how secure the PSN itself is.

8.2. Control-Plane Security

General security considerations with regard to the use of LDP are specified in section 5 of RFC 3036. Those considerations also apply to the case where LDP is used to set up pseudowires.

A pseudowire connects two attachment circuits. It is important to make sure that LDP connections are not arbitrarily accepted from anywhere, or else a local attachment circuit might get connected to an arbitrary remote attachment circuit. Therefore, an incoming LDP session request MUST NOT be accepted unless its IP source address is known to be the source of an "eligible" LDP peer. The set of eligible peers could be pre-configured (either as a list of IP addresses, or as a list of address/mask combinations), or it could be discovered dynamically via an auto-discovery protocol that is itself trusted. (Obviously, if the auto-discovery protocol were not trusted, the set of "eligible peers" it produces could not be trusted.)

Even if an LDP connection request appears to come from an eligible peer, its source address may have been spoofed. Therefore, some means of preventing source address spoofing must be in place. For example, if all the eligible peers are in the same network, source address filtering at the border routers of that network could eliminate the possibility of source address spoofing.

The LDP MD5 authentication key option, as described in section 2.9 of RFC 3036, MUST be implemented, and for a greater degree of security, it must be used. This provides integrity and authentication for the LDP messages and eliminates the possibility of source address spoofing. Use of the MD5 option does not provide privacy, but privacy of the LDP control messages is not usually considered important. As the MD5 option relies on the configuration of pre-

shared keys, it does not provide much protection against replay attacks. In addition, its reliance on pre-shared keys may make it very difficult to deploy when the set of eligible neighbors is determined by an auto-configuration protocol.

When the Generalized ID FEC Element is used, it is possible that a particular LDP peer may be one of the eligible LDP peers but may not be the right one to connect to the particular attachment circuit identified by the particular instance of the Generalized ID FEC element. However, given that the peer is known to be one of the eligible peers (as discussed above), this would be the result of a configuration error, rather than a security problem. Nevertheless, it may be advisable for a PE to associate each of its local attachment circuits with a set of eligible peers rather than have just a single set of eligible peers associated with the PE as a whole.

9. Acknowledgements

The authors wish to acknowledge the contributions of Vach Kompella, Vanson Lim, Wei Luo, Himanshu Shah, and Nick Weeds.

10. Normative References

[RFC2119] Bradner S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997

[RFC3036] Andersson, L., Doolan, P., Feldman, N., Fredette, A., and B. Thomas, "LDP Specification", RFC 3036, January 2001.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.

[IANA] Martini, L., "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)", BCP 116, RFC 4446, April 2006.

11. Informative References

[CEP] Malis, A., Pate, P., Cohen, R., Ed., and D. Zelig, "SONET/SDH Circuit Emulation Service Over Packet (CEP)", Work in Progress.

[SAToP] Vainshtein, A., Ed. and Y. Stein, Ed., "Structure-Agnostic TDM over Packet (SAToP)", Work in Progress.

[FRAME] Martini, L., Ed. and C. Kawa, Ed., "Encapsulation Methods for Transport of Frame Relay Over MPLS Networks", Work in Progress.

[ATM] Martini, L., Ed., El-Aawar, N., and M. Bocci, Ed., "Encapsulation Methods for Transport of ATM Over MPLS Networks", Work in Progress.

[PPPHDLC] Martini, L., Rosen, E., Heron, G., and A. Malis, "Encapsulation Methods for Transport of PPP/HDLC Frames Over IP and MPLS Networks", Work in Progress.

[ETH] Martini, L., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet Over MPLS Networks", RFC 4448, April 2006.

[SDH] American National Standards Institute, "Synchronous Optical Network Formats," ANSI T1.105-1995.

[ITUG] ITU Recommendation G.707, "Network Node Interface For The Synchronous Digital Hierarchy", 1996.

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

[RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, January 1998.

12. Additional Contributing Authors

Dimitri Stratton Vlachos
Mazu Networks, Inc.
125 Cambridgepark Drive
Cambridge, MA 02140

EMail: d@mazunetworks.com

Jayakumar Jayakumar,
Cisco Systems Inc.
225, E.Tasman, MS-SJ3/3,
San Jose, CA, 95134

EMail: jjayakum@cisco.com

Alex Hamilton,
Cisco Systems Inc.
285 W. Tasman, MS-SJCI/3/4,
San Jose, CA, 95134

EMail: tahamilt@cisco.com

Steve Vogelsang
ECI Telecom
Omega Corporate Center
1300 Omega Drive
Pittsburgh, PA 15205

EMail: stephen.vogelsang@ecitele.com

John Shirron
ECI Telecom
Omega Corporate Center
1300 Omega Drive
Pittsburgh, PA 15205

EMail: john.shirron@ecitele.com

Andrew G. Malis
Tellabs
90 Rio Robles Dr.
San Jose, CA 95134

EMail: Andy.Malis@tellabs.com

Vinai Sirkay
Redback Networks
300 Holger Way
San Jose, CA 95134

EMail: vsirkay@redback.com

Vasile Radoaca
Nortel Networks
600 Technology Park
Billerica MA 01821

EMail: vasile@nortelnetworks.com

Chris Liljenstolpe
Alcatel
11600 Sallie Mae Dr.
9th Floor
Reston, VA 20193

EMail: chris.liljenstolpe@alcatel.com

Dave Cooper
Global Crossing
960 Hamlin Court
Sunnyvale, CA 94089

EMail: dcooper@gblx.net

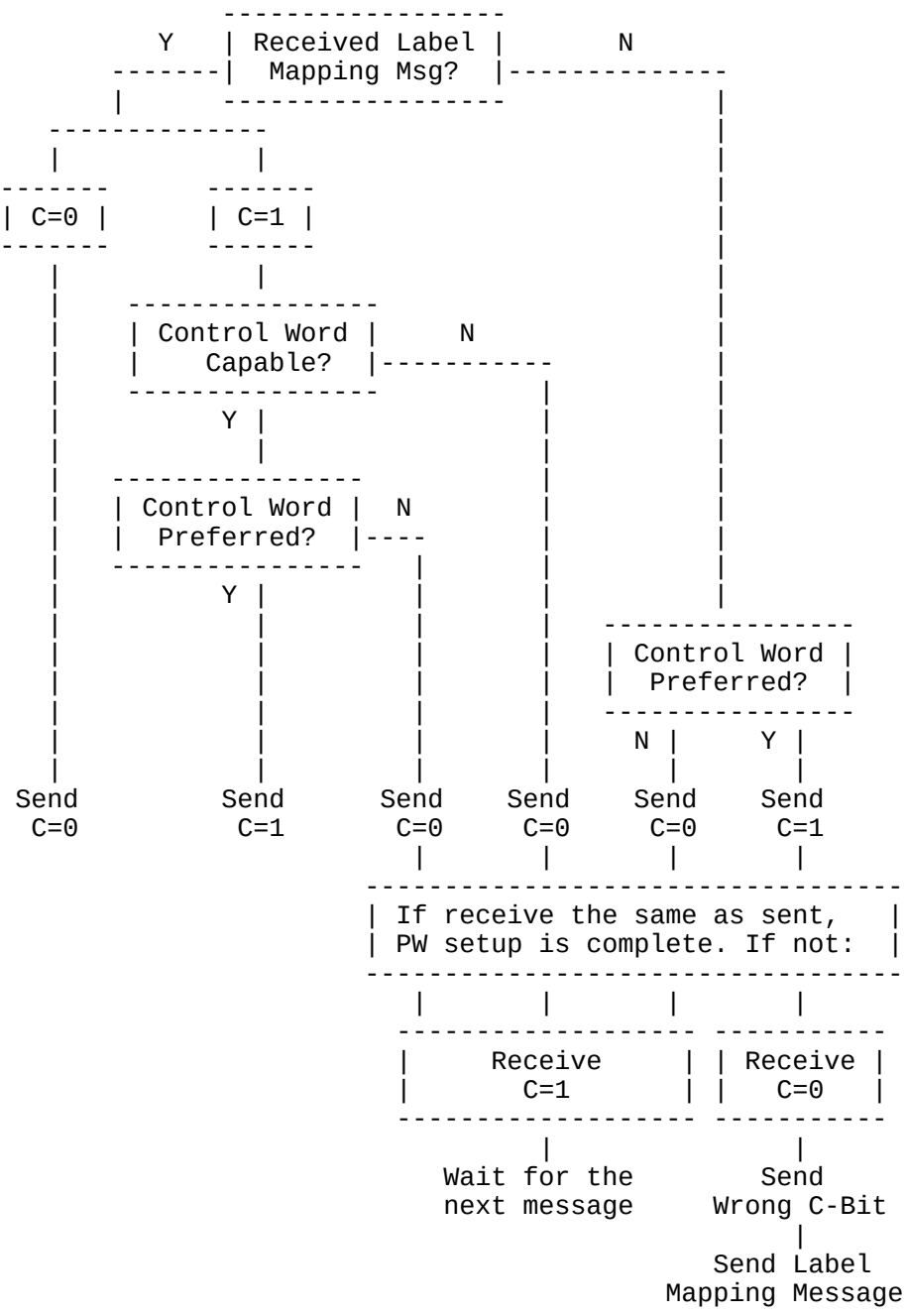
Kireeti Kompella
Juniper Networks
1194 N. Mathilda Ave
Sunnyvale, CA 94089

EMail: kireeti@juniper.net

Dan Tappan
Cisco Systems, Inc.
1414 Massachusetts Avenue
Boxborough, MA 01719

EMail: tappan@cisco.com

Appendix A. C-bit Handling Procedures Diagram



Authors' Addresses

Luca Martini
Cisco Systems, Inc.
9155 East Nichols Avenue, Suite 400
Englewood, CO, 80112

EMail: lmartini@cisco.com

Nasser El-Aawar
Level 3 Communications, LLC.
1025 Eldorado Blvd.
Broomfield, CO, 80021

EMail: nna@level3.net

Giles Heron
Tellabs
Abbey Place
24-28 Easton Street
High Wycombe
Bucks
HP11 1NT
UK

EMail: giles.heron@tellabs.com

Eric C. Rosen
Cisco Systems, Inc.
1414 Massachusetts Avenue
Boxborough, MA 01719

EMail: erosen@cisco.com

Toby Smith
Network Appliance, Inc.
800 Cranberry Woods Drive
Suite 300
Cranberry Township, PA 16066

EMail: tob@netapp.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

Exhibit 8

Network Working Group
Request for Comments: 4448
Category: Standards Track

L. Martini, Ed.
E. Rosen
Cisco Systems, Inc.
N. El-Aawar
Level 3 Communications, LLC
G. Heron
Tellabs

April 2006

Encapsulation Methods for Transport of Ethernet over MPLS Networks

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

An Ethernet pseudowire (PW) is used to carry Ethernet/802.3 Protocol Data Units (PDUs) over an MPLS network. This enables service providers to offer "emulated" Ethernet services over existing MPLS networks. This document specifies the encapsulation of Ethernet/802.3 PDUs within a pseudowire. It also specifies the procedures for using a PW to provide a "point-to-point Ethernet" service.

Table of Contents

1. Introduction

An Ethernet pseudowire (PW) allows Ethernet/802.3 [802.3] Protocol Data Units (PDUs) to be carried over a Multi-Protocol Label Switched [MPLS-ARCH] network. In addressing the issues associated with carrying an Ethernet PDU over a packet switched network (PSN), this document assumes that a pseudowire (PW) has been set up by using a control protocol such as the one as described in [PWE3-CTRL]. The design of Ethernet pseudowire described in this document conforms to the pseudowire architecture described in [RFC3985]. It is also assumed in the remainder of this document that the reader is familiar with RFC 3985.

The Pseudowire Emulation Edge-to-Edge (PWE3) Ethernet PDU consists of the Destination Address, Source Address, Length/Type, MAC Client Data, and padding extracted from a MAC frame as a concatenated octet sequence in their original order [PDU].

In addition to the Ethernet PDU format used within the pseudowire, this document discusses:

- Procedures for using a PW in order to provide a pair of Customer Edge (CE) routers with an emulated (point-to-point) Ethernet service, including the procedures for the processing of Provider Edge (PE)-bound and CE-bound Ethernet PDUs [RFC3985]

- Ethernet-specific quality of service (QoS) and security considerations

- Inter-domain transport considerations for Ethernet PW

The following two figures describe the reference models that are derived from [RFC3985] to support the Ethernet PW emulated services.

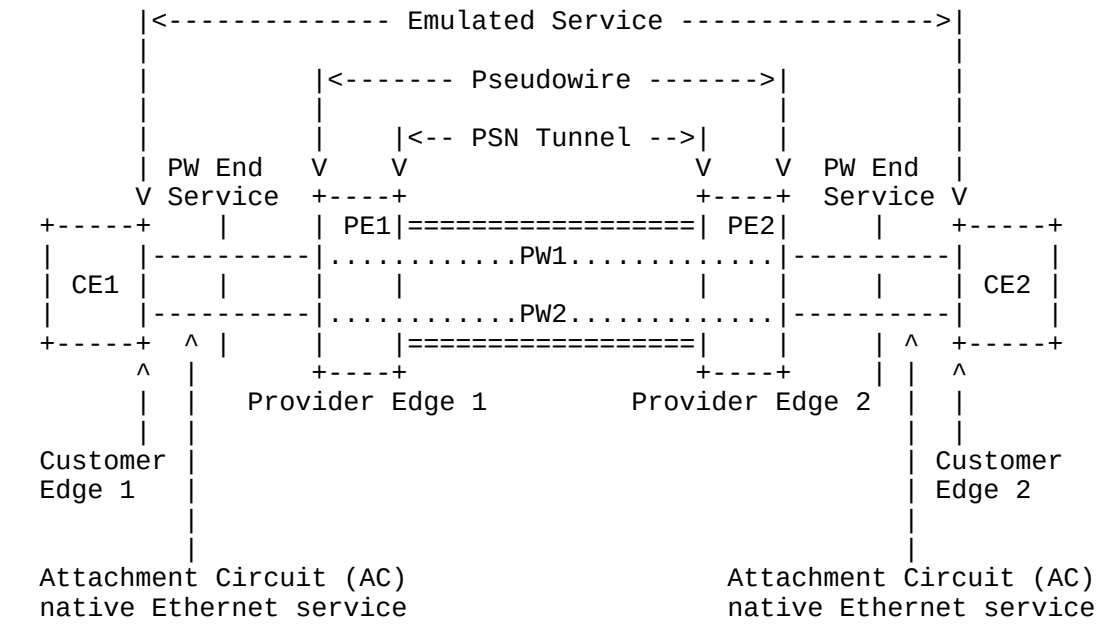


Figure 1: PWE3 Ethernet/VLAN Interface Reference Configuration

The "emulated service" shown in Figure 1 is, strictly speaking, a bridged LAN; the PEs have MAC interfaces, consume MAC control frames, etc. However, the procedures specified herein only support the case in which there are two CEs on the "emulated LAN". Hence we refer to this service as "emulated point-to-point Ethernet". Specification of the procedures for using pseudowires to emulate LANs with more than two CEs are out of the scope of the current document.

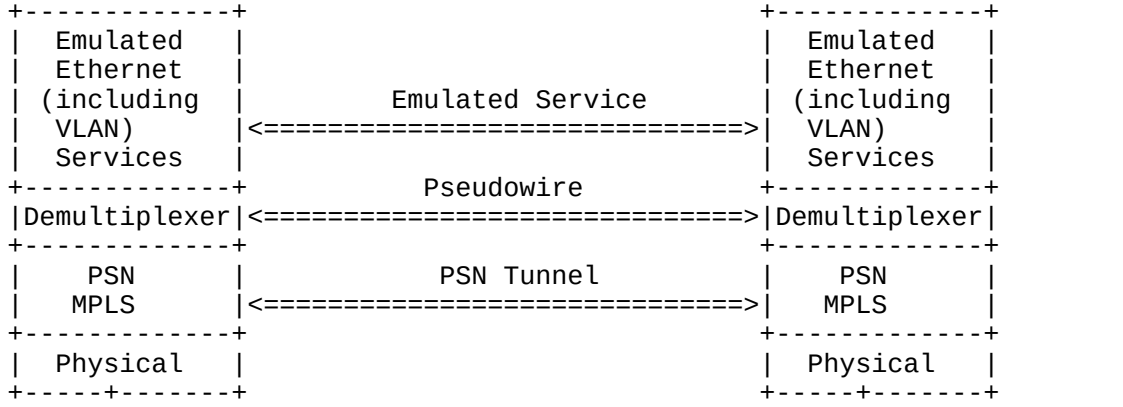


Figure 2: Ethernet PWE3 Protocol Stack Reference Model

For the purpose of this document, PE1 will be defined as the ingress router, and PE2 as the egress router. A layer 2 PDU will be received at PE1, encapsulated at PE1, transported, decapsulated at PE2, and transmitted out on the attachment circuit of PE2.

An Ethernet PW emulates a single Ethernet link between exactly two endpoints. The mechanisms described in this document are agnostic to that which is beneath the "Pseudowire" level in Figure 2, concerning itself only with the "Emulated Service" portion of the stack.

The following reference model describes the termination point of each end of the PW within the PE:

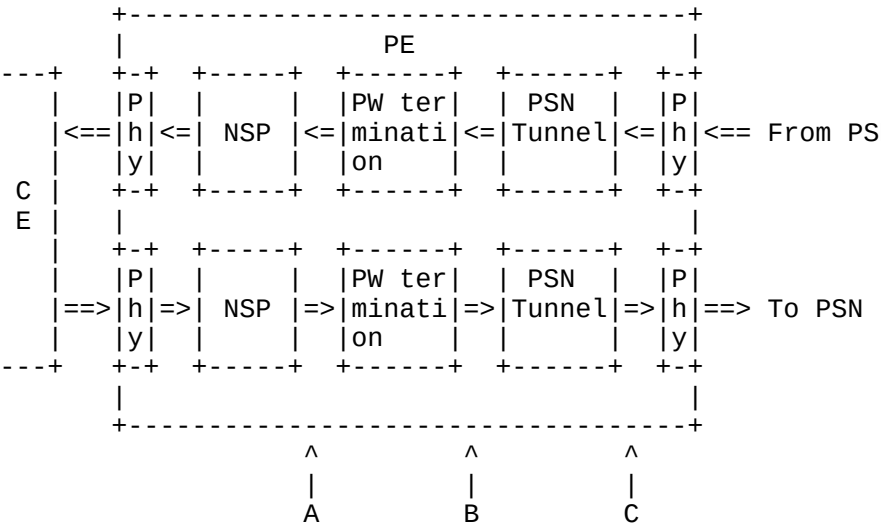


Figure 3: PW Reference Diagram

The PW terminates at a logical port within the PE, defined at point B in the above diagram. This port provides an Ethernet MAC service that will deliver each Ethernet frame that is received at point A, unaltered, to the point A in the corresponding PE at the other end of the PW.

The Native Service Processing (NSP) function includes frame processing that is required for the Ethernet frames that are forwarded to the PW termination point. Such functions may include stripping, overwriting or adding VLAN tags, physical port multiplexing and demultiplexing, PW-PW bridging, L2 encapsulation, shaping, policing, etc. These functions are specific to the Ethernet technology, and may not be required for the PW emulation service.

The points to the left of A, including the physical layer between the CE and PE, and any adaptation (NSP) functions between it and the PW terminations, are outside of the scope of PWE3 and are not defined here.

"PW Termination", between A and B, represents the operations for setting up and maintaining the PW, and for encapsulating and decapsulating the Ethernet frames as necessary to transmit them across the MPLS network.

An Ethernet PW operates in one of two modes: "raw mode" or "tagged mode". In tagged mode, each frame MUST contain at least one 802.1Q [802.1Q] VLAN tag, and the tag value is meaningful to the NSPs at the two PW termination points. That is, the two PW termination points must have some agreement (signaled or manually configured) on how to process the tag. On a raw mode PW, a frame MAY contain an 802.1Q VLAN tag, but if it does, the tag is not meaningful to the NSPs, and passes transparently through them.

Additional terminology relevant to pseudowires and Layer 2 Virtual Private Networking may be found in [RFC4026].

2. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Applicability Statement

The Ethernet PW emulation allows a service provider to offer a "port to port" Ethernet-based service across an MPLS packet switched network (PSN) while the Ethernet VLAN PW emulation allows an "Ethernet VLAN to VLAN" based service across an MPLS packet switched network (PSN).

The Ethernet or Ethernet VLAN PW has the following characteristics in relationship to the respective native service:

- An Ethernet PW connects two Ethernet ACs while an Ethernet VLAN PW connects two Ethernet VLAN ACs, supporting bidirectional transport of variable length Ethernet frames. The ingress Native Service Processing (NSP) function strips the preamble and frame check sequence (FCS) from the Ethernet frame and transports the frame in its entirety across the PW. This is done regardless of the presence of the 802.1Q tag in the frame. The egress NSP function receives the Ethernet frame from the PW and regenerates the preamble or FCS before forwarding the frame

to the attachment circuit. Since the FCS is not transported across either Ethernet or Ethernet VLAN PWs, payload integrity transparency may be lost. The OPTIONAL method described in [FCS] can be used to achieve payload integrity transparency on Ethernet or Ethernet VLAN PWs.

- For an Ethernet VLAN PW, VLAN tag rewrite can be achieved by NSP at the egress PE, which is outside the scope of this document.

- The Ethernet or Ethernet VLAN PW only supports homogeneous Ethernet frame type across the PW; both ends of the PW must be either tagged or untagged. Heterogeneous frame type support achieved with NSP functionality is outside the scope of this document.

- Ethernet port or Ethernet VLAN status notification is provided using the PW Status TLV in the Label Distribution Protocol (LDP) status notification message. Loss of connectivity between PEs can be detected by the LDP session closing, or by using [VCCV] mechanisms. The PE can convey these indications back to its attached Remote System.

- The maximum frame size that can be supported is limited by the PSN MTU minus the MPLS header size, unless fragmentation and reassembly are used [FRAG].

- The packet switched network may reorder, duplicate, or silently drop packets. Sequencing MAY be enabled in the Ethernet or Ethernet VLAN PW to detect lost, duplicate, or out-of-order packets on a per-PW basis.

- The faithfulness of an Ethernet or Ethernet VLAN PW may be increased by leveraging Quality of Service features of the PEs and the underlying PSN. (See Section 4.7, "QoS Considerations".)

4. Details Specific to Particular Emulated Services

4.1. Ethernet Tagged Mode

The Ethernet frame will be encapsulated according to the procedures defined later in this document for tagged mode. It should be noted that if the VLAN identifier is modified by the egress PE, the Ethernet spanning tree protocol might fail to work properly. If this issue is of significance, the VLAN identifier MUST be selected in such a way that it matches on the attachment circuits at both ends of the PW.

If the PE detects a failure on the Ethernet physical port, or the port is administratively disabled, it MUST send a PW status notification message for all PWs associated with the port.

This mode uses service-delimiting tags to map input Ethernet frames to respective PWs and corresponds to PW type 0x0004 "Ethernet Tagged Mode" [IANA].

4.2. Ethernet Raw Mode

The Ethernet frame will be encapsulated according to the procedures defined later in this document for raw mode. If the PE detects a failure on the Ethernet input port, or the port is administratively disabled, the PE MUST send an appropriate PW status notification message to the corresponding remote PE.

In this mode, all Ethernet frames received on the attachment circuit of PE1 will be transmitted to PE2 on a single PW. This service corresponds to PW type 0x0005 "Ethernet" [IANA].

4.3. Ethernet-Specific Interface Parameter LDP Sub-TLV

This LDP sub-Type Length Value [LDP] specifies interface-specific parameters. When applicable, it MUST be used to validate that the PEs, and the ingress and egress ports at the edges of the circuit, have the necessary capabilities to interoperate with each other. The Interface parameter TLV is defined in [PWE3-CTRL], the IANA registry with initial values for interface parameter sub-TLV types is defined in [IANA], but the Ethernet-specific interface parameters are specified as follows:

- 0x06 Requested VLAN ID Sub-TLV

  An Optional 16-bit value indicating the requested VLAN ID. This parameter MUST be used by a PE that is incapable of rewriting the 802.1Q Ethernet VLAN tag on output. If the ingress PE receives this request, it MUST rewrite the VLAN ID contained inside the VLAN Tag at the input to match the requested VLAN ID. If this is not possible, and the VLAN ID does not already match the configured ingress VLAN ID, the PW MUST not be enabled. This parameter is applicable only to PW type 0x0004.

4.4. Generic Procedures

When the NSP/Forwarder hands a frame to the PW termination function:

- The preamble (if any) and FCS are stripped off.
- The control word as defined in Section 4.6, "The Control Word", is, if necessary, prepended to the resulting frame. The conditions under which the control word is or is not used are specified below.
- The proper pseudowire demultiplexer (PW Label) is prepended to the resulting packet.
- The proper tunnel encapsulation is prepended to the resulting packet.
- The packet is transmitted.

The way in which the proper tunnel encapsulation and pseudowire demultiplexer is chosen depends on the procedures that were used to set up the pseudowire.

The tunnel encapsulation depends on how the MPLS PSN is set up. This can include no label, one label, or multiple labels. The proper pseudowire demultiplexer is an MPLS label whose value is determined by the PW setup and maintenance protocols.

When a packet arrives over a PW, the tunnel encapsulation and PW demultiplexer are stripped off. If the control word is present, it is processed and stripped off. The resulting frame is then handed to the Forwarder/NSP. Regeneration of the FCS is considered to be an NSP responsibility.

4.4.1. Raw Mode vs. Tagged Mode

When the PE receives an Ethernet frame, and the frame has a VLAN tag, we can distinguish two cases:

1. The tag is service-delimiting. This means that the tag was placed on the frame by some piece of service provider-operated equipment, and the tag is used by the service provider to distinguish the traffic. For example, LANs from different customers might be attached to the same service provider switch, which applies VLAN tags to distinguish one customer’s traffic from another’s, and then forwards the frames to the PE.

2. The tag is not service-delimiting. This means that the tag was placed in the frame by a piece of customer equipment, and is not meaningful to the PE.

Whether or not the tag is service-delimiting is determined by local configuration on the PE.

If an Ethernet PW is operating in raw mode, service-delimiting tags are NEVER sent over the PW. If a service-delimiting tag is present when the frame is received from the attachment circuit by the PE, it MUST be stripped (by the NSP) from the frame before the frame is sent to the PW.

If an Ethernet PW is operating in tagged mode, every frame sent on the PW MUST have a service-delimiting VLAN tag. If the frame as received by the PE from the attachment circuit does not have a service-delimiting VLAN tag, the PE must prepend the frame with a dummy VLAN tag before sending the frame on the PW. This is the default operating mode. This is the only REQUIRED mode.

In both modes, non-service-delimiting tags are passed transparently across the PW as part of the payload. It should be noted that a single Ethernet packet may contain more than one tag. At most, one of these tags may be service-delimiting. In any case, the NSP function may only inspect the outermost tag for the purpose of adapting the Ethernet frame to the pseudowire.

In both modes, the service-delimiting tag values have only local significance, i.e., are meaningful only at a particular PE-CE interface. When tagged mode is used, the PE that receives a frame from the PW may rewrite the tag value, or may strip the tag entirely, or may leave the tag unchanged, depending on its configuration. When raw mode is used, the PE that receives a frame may or may not need to add a service-delimiting tag before transmitting the frame on the attachment circuit; however, it MUST not rewrite or remove any tags that are already present.

The following table illustrates the operations that might be performed at input from the attachment circuit:

Tag->	service delimiting	non service delimiting
Raw Mode	1st VLAN Tag Removed	no operation performed
Tagged Mode	NO OP or Tag Added	Tag Added

4.4.2. MTU Management on the PE/CE Links

The Ethernet PW MUST NOT be enabled unless it is known that the MTUs of the CE-PE links are the same at both ends of the PW. If an egress router receives an encapsulated layer 2 PDU whose payload length (i.e., the length of the PDU itself without any of the encapsulation headers) exceeds the MTU of the destination layer 2 interface, the PDU MUST be dropped.

4.4.3. Frame Ordering

In general, applications running over Ethernet do not require strict frame ordering. However, the IEEE definition of 802.3 [802.3] requires that frames from the same conversation in the context of link aggregation (clause 43) are delivered in sequence. Moreover, the PSN cannot (in the general case) be assumed to provide or to guarantee frame ordering. An Ethernet PW can, through use of the control word, provide strict frame ordering. If this option is enabled, any frames that get misordered by the PSN will be dropped or reordered by the receiving PW endpoint. If strict frame ordering is a requirement for a particular PW, this option MUST be enabled.

4.4.4. Frame Error Processing

An encapsulated Ethernet frame traversing a pseudowire may be dropped, corrupted, or delivered out-of-order. As described in [PWE3-REQ], frame loss, corruption, and out-of-order delivery are considered to be a "generalized bit error" of the pseudowire. PW frames that are corrupted will be detected at the PSN layer and dropped.

At the ingress of the PW, the native Ethernet frame error processing mechanisms MUST be enabled. Therefore, if a PE device receives an Ethernet frame containing hardware-level Cyclic Redundancy Check (CRC) errors, framing errors, or a runt condition, the frame MUST be discarded on input. Note that defining this processing is part of the NSP function and is outside the scope of this document.

4.4.5. IEEE 802.3x Flow Control Interworking

In a standard Ethernet network, the flow control mechanism is optional and typically configured between the two nodes on a point-to-point link (e.g., between the CE and the PE). IEEE 802.3x PAUSE frames MUST NOT be carried across the PW. See Appendix A for notes on CE-PE flow control.

4.5. Management

The Ethernet PW management model follows the general PW management model defined in [RFC3985] and [PWE3-MIB]. Many common PW management facilities are provided here, with no additional Ethernet specifics necessary. Ethernet-specific parameters are defined in an additional MIB module, [PW-MIB].

4.6. The Control Word

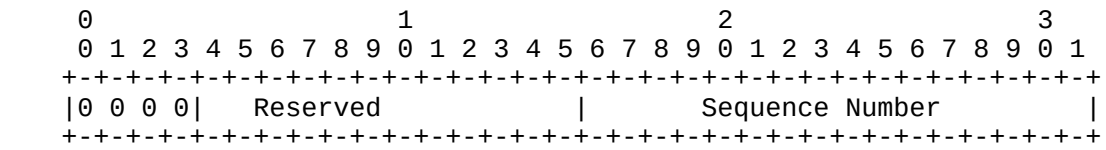
The control word defined in this section is based on the Generic PW MPLS Control Word as defined in [PWE3-CW]. It provides the ability to sequence individual frames on the PW, avoidance of equal-cost multiple-path load-balancing (ECMP) [RFC2992], and Operations and Management (OAM) mechanisms including VCCV [VCCV].

[PWE3-CW] states, "If a PW is sensitive to packet misordering and is being carried over an MPLS PSN that uses the contents of the MPLS payload to select the ECMP path, it MUST employ a mechanism which prevents packet misordering." This is necessary because ECMP implementations may examine the first nibble after the MPLS label stack to determine whether the labelled packet is IP or not. Thus, if the source MAC address of an Ethernet frame carried over the PW without a control word present begins with 0x4 or 0x6, it could be mistaken for an IPv4 or IPv6 packet. This could, depending on the configuration and topology of the MPLS network, lead to a situation where all packets for a given PW do not follow the same path. This may increase out-of-order frames on a given PW, or cause OAM packets to follow a different path than actual traffic (see Section 4.4.3, "Frame Ordering").

The features that the control word provides may not be needed for a given Ethernet PW. For example, ECMP may not be present or active on a given MPLS network, strict frame sequencing may not be required, etc. If this is the case, the control word provides little value and is therefore optional. Early Ethernet PW implementations have been deployed that do not include a control word or the ability to process one if present. To aid in backwards compatibility, future implementations MUST be able to send and receive frames without the control word present.

In all cases, the egress PE MUST be aware of whether the ingress PE will send a control word over a specific PW. This may be achieved by configuration of the PEs, or by signaling, as defined in [PWE3-CTRL].

The control word is defined as follows:



In the above diagram, the first 4 bits MUST be set to 0 to indicate PW data. The rest of the first 16 bits are reserved for future use. They MUST be set to 0 when transmitting, and MUST be ignored upon receipt.

The next 16 bits provide a sequence number that can be used to guarantee ordered frame delivery. The processing of the sequence number field is OPTIONAL.

The sequence number space is a 16-bit, unsigned circular space. The sequence number value 0 is used to indicate that the sequence number check algorithm is not used. The sequence number processing algorithm is found in [PWE3-CW].

4.7. QoS Considerations

The ingress PE MAY consider the user priority (PRI) field [802.1Q] of the VLAN tag header when determining the value to be placed in a QoS field of the encapsulating protocol (e.g., the EXP fields of the MPLS label stack). In a similar way, the egress PE MAY consider the QoS field of the encapsulating protocol (e.g., the EXP fields of the MPLS label stack) when queuing the frame for transmission towards the CE.

A PE MUST support the ability to carry the Ethernet PW as a best-effort service over the MPLS PSN. PRI bits are kept transparent between PE devices, regardless of the QoS support of the PSN.

If an 802.1Q VLAN field is added at the PE, a default PRI setting of zero MUST be supported, a configured default value is recommended, or the value may be mapped from the QoS field of the PSN, as referred to above.

A PE may support additional QoS support by means of one or more of the following methods:

i. One class of service (CoS) per PW End Service (PWES), mapped to a single CoS PW at the PSN.
ii. Multiple CoS per PWES mapped to a single PW with multiple CoS at the PSN.
iii. Multiple CoS per PWES mapped to multiple PWs at the PSN.

Examples of the cases above and details of the service mapping considerations are described in Appendix B.

The PW guaranteed rate at the MPLS PSN level is PW service provider policy based on agreement with the customer, and may be different from the Ethernet physical port rate.

5. Security Considerations

The Ethernet pseudowire type is subject to all of the general security considerations discussed in [RFC3985] and [PWE3-CTRL].

The Ethernet pseudowire is transported on an MPLS PSN; therefore, the security of the pseudowire itself will only be as good as the security of the MPLS PSN. The MPLS PSN can be secured by various methods, as described in [MPLS-ARCH].

Security achieved by access control of MAC addresses is out of the scope of this document. Additional security requirements related to the use of PW in a switching (virtual bridging) environment are not discussed here as they are not within the scope of this document.

6. PSN MTU Requirements

The MPLS PSN MUST be configured with an MTU that is large enough to transport a maximum-sized Ethernet frame that has been encapsulated with a control word, a pseudowire demultiplexer, and a tunnel encapsulation. With MPLS used as the tunneling protocol, for example, this is likely to be 8 or more bytes greater than the largest frame size. The methodology described in [FRAG] MAY be used to fragment encapsulated frames that exceed the PSN MTU. However, if [FRAG] is not used and if the ingress router determines that an encapsulated layer 2 PDU exceeds the MTU of the PSN tunnel through which it must be sent, the PDU MUST be dropped.

7. Normative References

[PWE3-CW] Bryant, S., Swallow, G., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, February 2006.

[IANA] Martini, L., "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)", BCP 116, RFC 4446, April 2006.

[PWE3-CTRL] Martini, L., El-Aawar, N., Heron, G., Rosen, E., Tappan, D., and T. Smith, "Pseudowire Setup and Maintenance using the Label Distribution Protocol (LDP)", RFC 4447, April 2006.

[MPLS-ARCH] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[802.3] IEEE802.3-2005, ISO/IEC 8802-3: 2000 (E), "IEEE Standard for Information technology -- Telecommunications and information exchange between systems -- Local and metropolitan
area networks -- Specific requirements -- Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications", 2005.

[802.1Q] ANSI/IEEE Standard 802.1Q-2005, "IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks", 2005.

[PDU] IEEE Std 802.3, 1998 Edition, "Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications" figure 3.1, 1998

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

8. Informative References

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

[PW-MIB] Zelig, D. and T. Nadeau, "Ethernet Pseudo Wire (PW) Management Information Base", Work in Progress, February 2006.

[PWE3-REQ] Xiao, X., McPherson, D., and P. Pate, "Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)", RFC 3916, September 2004.

[PWE3-MIB] Zelig, D., Ed. and T. Nadeau, Ed., "Pseudo Wire (PW) Management Information Base", Work in Progress, February 2006.

[LDP] Andersson, L., Doolan, P., Feldman, N., Fredette, A., and B. Thomas, "LDP Specification", RFC 3036, January 2001.

[FRAG] Malis, A. and W. Townsley, "PWE3 Fragmentation and Reassembly", Work in Progress, February 2005.

[FCS] Malis, A., Allan, D., and N. Del Regno, "PWE3 Frame Check Sequence Retention", Work in Progress, September 2005.

[VCCV] Nadeau, T., Ed. and R. Aggarwal, Ed., "Pseudo Wire Virtual Circuit Connectivity Verification (VCCV)", Work in Progress, August 2005.

[RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", RFC 2992, November 2000.

[RFC4026] Andersson, L. and T. Madsen, "Provider Provisioned Virtual Private Network (VPN) Terminology", RFC 4026, March 2005.

[L2TPv3] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.

9. Significant Contributors

Andrew G. Malis
Tellabs
90 Rio Robles Dr.
San Jose, CA 95134

EMail: Andy.Malis@tellabs.com

Dan Tappan
Cisco Systems, Inc.
1414 Massachusetts Avenue
Boxborough, MA 01719

EMail: tappan@cisco.com

Steve Vogelsang
ECI Telecom
Omega Corporate Center
1300 Omega Drive
Pittsburgh, PA 15205

EMail: stephen.vogelsang@ecitele.com

Vinai Sirkay
Reliance Infocomm
Dhirubai Ambani Knowledge City
Navi Mumbai 400 709
India

EMail: vinai@sirkay.com

Vasile Radoaca
Nortel Networks
600 Technology Park
Billerica MA 01821

EMail: vasile@nortelnetworks.com

Chris Liljenstolpe
Alcatel
11600 Sallie Mae Dr.
9th Floor
Reston, VA 20193

EMail: chris.liljenstolpe@alcatel.com

Kireeti Kompella
Juniper Networks
1194 N. Mathilda Ave
Sunnyvale, CA 94089

EMail: kireeti@juniper.net

Tricci So
Nortel Networks 3500 Carling Ave.,
Nepean, Ontario,
Canada, K2H 8E9.

EMail: tso@nortelnetworks.com

XiPeng Xiao
Riverstone Networks
5200 Great America Parkway
Santa Clara, CA 95054

EMail: xxiao@riverstonenet.com

Christopher O. Flores
T-Systems
10700 Parkridge Boulevard
Reston, VA 20191
USA

EMail: christopher.flores@usa.telekom.de

David Zelig
Corrigent Systems
126, Yigal Alon St.
Tel Aviv, ISRAEL

EMail: davidz@corrigent.com

Raj Sharma
Luminous Networks, Inc.
10460 Bubb Road
Cupertino, CA 95014

EMail: raj@luminous.com

Nick Tingle
TiMetra Networks
274 Ferguson Drive
Mountain View, CA 94043

EMail: nick@timetra.com

Sunil Khandekar
TiMetra Networks
274 Ferguson Drive
Mountain View, CA 94043

EMail: sunil@timetra.com

Loa Andersson
TLA-group

EMail: loa@pi.se

Appendix A. Interoperability Guidelines

A.1. Configuration Options

The following is a list of the configuration options for a point-to-point Ethernet PW based on the reference points of Figure 3:

Service and Encap on A	Encap on C	Operation at B ingress/egress	Remarks
1) Raw	Raw - Same as A		
2) Tag1	Tag2	Optional change of VLAN value	VLAN can be 0-4095 Change allowed in both directions
3) No Tag	Tag	Add/remove Tag field	Tag can be 0-4095 (note i)
4) Tag	No Tag	Remove/add Tag field	(note ii)

Figure 4: Configuration Options

Allowed combinations:

Raw and other services are not allowed on the same NSP virtual port (A). All other combinations are allowed, except that conflicting VLANs on (A) are not allowed. Note that in most point-to-point PW applications the NSP virtual port is the same entity as the physical port.

Notes:

i. Mode #3 MAY be limited to adding VLAN NULL only, since change of VLAN or association to specific VLAN can be done at the PW CE-bound side.

ii. Mode #4 exists in layer 2 switches, but is not recommended when operating with PW since it may not preserve the user's PRI bits. If there is a need to remove the VLAN tag (for TLS at the other end of the PW), it is recommended to use mode #2 with tag2=0 (NULL VLAN) on the PW and use mode #3 at the other end of the PW.

A.2. IEEE 802.3x Flow Control Considerations

If the receiving node becomes congested, it can send a special frame, called the PAUSE frame, to the source node at the opposite end of the connection. The implementation MUST provide a mechanism for terminating PAUSE frames locally (i.e., at the local PE). It MUST operate as follows: PAUSE frames received on a local Ethernet port SHOULD cause the PE device to buffer, or to discard, further Ethernet frames for that port until the PAUSE condition is cleared. Optionally, the PE MAY simply discard PAUSE frames.

If the PE device wishes to pause data received on a local Ethernet port (perhaps because its own buffers are filling up or because it has received notification of congestion within the PSN), then it MAY issue a PAUSE frame on the local Ethernet port, but MUST clear this condition when willing to receive more data.

Appendix B. QoS Details

Section 4.7, "QoS Considerations", describes various modes for supporting PW QOS over the PSN. Examples of the above for a point-to-point VLAN service are:

- The classification to the PW is based on VLAN field, but the user PRI bits are mapped to different CoS markings (and network behavior) at the PW level. An example of this is a PW mapped to an E-LSP in an MPLS network.

- The classification to the PW is based on VLAN field and the PRI bits, and frames with different PRI bits are mapped to different PWs. An example is to map a PWES to different L-LSPs in MPLS PSN in order to support multiple CoS over an L-LSP-capable network, or to map a PWES to multiple L2TPv3 sessions [L2TPv3].

  The specific value to be assigned at the PSN for various CoS is out of the scope of this document.

B.1. Adaptation of 802.1Q CoS to PSN CoS

It is not required that the PSN will have the same CoS definition of CoS as defined in [802.1Q], and the mapping of 802.1Q CoS to PSN CoS is application specific and depends on the agreement between the customer and the PW provider. However, the following principles adopted from 802.1Q, Table 8-2, MUST be met when applying the set of PSN CoS based on user's PRI bits.

	#of available classes of service							
User Priority	1	2	3	4	5	6	7	8
0 Best Effort (Default)	0	0	0	1	1	1	1	2
1 Background	0	0	0	0	0	0	0	0
2 Spare	0	0	0	0	0	0	0	1
3 Excellent Effort	0	0	0	1	1	2	2	3
4 Controlled Load	0	1	1	2	2	3	3	4
5 Interactive Multimedia	0	1	1	2	3	4	4	5
6 Interactive Voice	0	1	2	3	4	5	5	6
7 Network Control	0	1	2	3	4	5	6	7

Figure 5: IEEE 802.1Q CoS Mapping

B.2. Drop Precedence

The 802.1P standard does not support drop precedence; therefore, from the PW PE-bound point of view there is no mapping required. It is, however, possible to mark different drop precedence for different PW frames based on the operator policy and required network behavior. This functionality is not discussed further here.

PSN QoS support and signaling of QoS are out of the scope of this document.

Authors' Addresses

Luca Martini, Editor
Cisco Systems, Inc.
9155 East Nichols Avenue, Suite 400
Englewood, CO, 80112

EMail: lmartini@cisco.com

Nasser El-Aawar
Level 3 Communications, LLC.
1025 Eldorado Blvd.
Broomfield, CO, 80021

EMail: nna@level3.net

Giles Heron
Tellabs
Abbey Place
24-28 Easton Street
High Wycombe
Bucks
HP11 1NT
UK

EMail: giles.heron@tellabs.com

Eric C. Rosen
Cisco Systems, Inc.
1414 Massachusetts Avenue
Boxborough, MA 01719

EMail: erosen@cisco.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

Exhibit 9

Network Working Group
Request for Comments: 4619
Category: Standards Track

L. Martini, Ed.
Cisco Systems, Inc.
C. Kawa, Ed.
Oz Communications
A. Malis, Ed.
Tellabs

September 2006

Encapsulation Methods for Transport of Frame Relay over Multiprotocol Label Switching (MPLS) Networks

Status of This Memo

Copyright Notice

Abstract

A frame relay pseudowire is a mechanism that exists between a provider’s edge network nodes and that supports as faithfully as possible frame relay services over an MPLS packet switched network (PSN). This document describes the detailed encapsulation necessary to transport frame relay packets over an MPLS network.

Table of Contents

1. Introduction

In an MPLS or IP network, it is possible to use control protocols such as those specified in [RFC4447] to set up "pseudowires" (PWs) that carry the Protocol Data Units of layer 2 protocols across the network. A number of these emulated PWs may be carried in a single tunnel. The main functions required to support frame relay PW by a Provider Edge (PE) include:

- encapsulation of frame relay specific information in a suitable pseudowire (PW) packet;

- transfer of a PW packet across an MPLS network for delivery to a peer PE;

- extraction of frame relay specific information from a PW packet by the remote peer PE;

- regeneration of native frame relay frames for forwarding across an egress port of the remote peer PE; and

- execution of any other operations as required to support frame relay service.

This document specifies the encapsulation for the emulated frame relay VC over an MPLS PSN. Although different layer 2 protocols require different information to be carried in this encapsulation, an attempt has been made to make the encapsulation as common as possible for all layer 2 protocols. Other layer 2 protocols are described in separate documents. [ATM] [RFC4448] [RFC4618]

The following figure describes the reference models that are derived from [RFC3985] to support the frame relay PW emulated services.

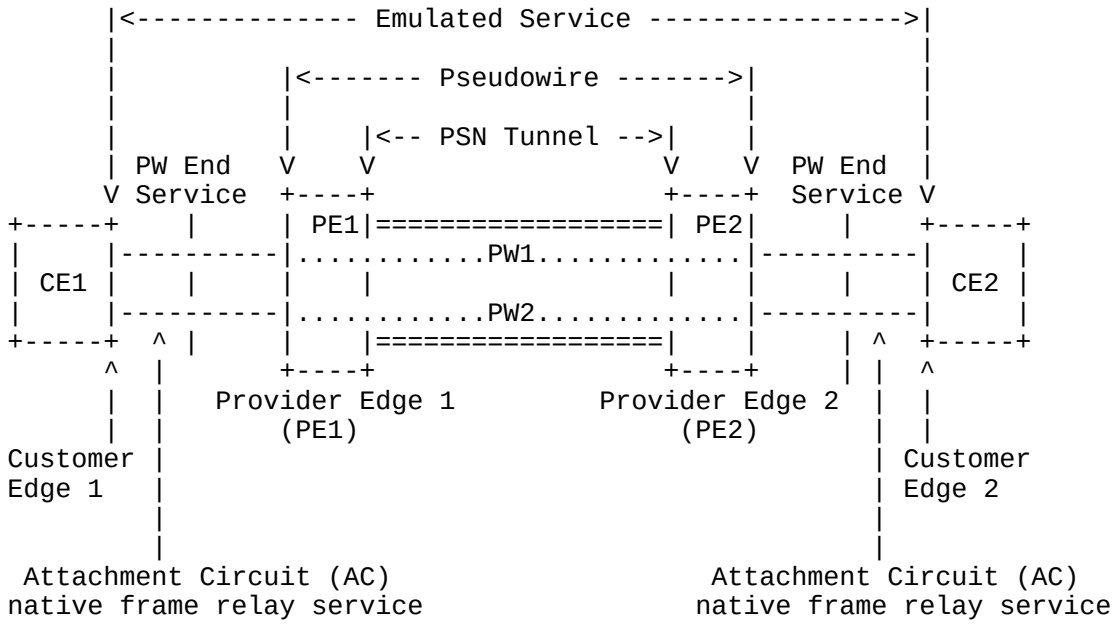


Figure 1. PWE3 frame relay PVC interface reference configuration

Two mapping modes can be defined between frame relay VCs and pseudowires: The first one is called "one-to-one" mapping, because there is a one-to-one correspondence between a frame relay VC and one pseudowire. The second mapping is called "many-to-one" mapping or "port mode" because multiple frame relay VCs assigned to a port are mapped to one pseudowire. The "port mode" encapsulation is identical to High-Level Data Link Control (HDLC) pseudowire encapsulation, which is described in [RFC4618].

2. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

Below are the definitions for the terms used throughout the document. PWE3 definitions can be found in [RFC3916, RFC3985]. This section defines terms specific to frame relay.

- Forward direction

  The forward direction is the direction taken by the frame being forwarded.

- Backward direction

  In frame relay, it is the direction opposite to the direction taken by a frame being forwarded (see also forward direction).

3. Co-authors

The following are co-authors of this document:

Nasser El-Aawar	Level 3 Communications, LLC
Eric C. Rosen	Cisco Systems
Daniel Tappan	Cisco Systems
Thomas K. Johnson	Litchfield Communications
Kireeti Kompella	Juniper Networks, Inc.
Steve Vogelsang	Laurel Networks, Inc.
Vinai Sirkay	Reliance Infocomm
Ravi Bhat	Nokia
Nishit Vasavada	Nokia
Giles Heron	Tellabs
Dimitri Stratton Vlachos	Mazu Networks,Inc.
Chris Liljenstolpe	Cable & Wireless
Prayson Pate	Overture Networks, Inc

4. Acronyms and Abbreviations

BECN	Backward Explicit Congestion Notification
CE	Customer Edge
C/R	Command/Response
DE	Discard Eligibility
DLCI	Data Link Connection Identifier
FCS	Frame Check Sequence
FECN	Forward Explicit Congestion Notification
FR	Frame Relay
LSP	Label Switched Path
LSR	Label Switching Router
MPLS	Multiprotocol Label Switching
MTU	Maximum Transfer Unit
NNI	Network-Network Interface
PE	Provider Edge
PSN	Packet Switched Network
PW	Pseudowire
PWE3	Pseudowire Emulation Edge to Edge
POS	Packet over SONET/SDH
PVC	Permanent Virtual Circuit
QoS	Quality of Service
SVC	Switched Virtual Circuit
UNI	User-Network Interface
VC	Virtual Circuit

5. Applicability Statement

Frame relay over PW service is not intended to emulate the traditional frame relay service perfectly, but it can be used for applications that need frame relay transport service.

The following are notable differences between traditional frame relay service and the protocol described in this document:

- Frame ordering can be preserved using the OPTIONAL sequence field in the control word; however, implementations are not required to support this feature.

- The Quality of Service model for traditional frame relay can be emulated; however, this is outside the scope of this document.

- A Frame relay port mode PW does not process any frame relay status messages or alarms as described in [Q922] [Q933]

- The frame relay BECN and FECN bit are transparent to the MPLS network and cannot reflect the status of the MPLS network.

- Support for frame relay SVC and Switched Permanent Virtual Circuit (SPVC) is outside the scope of this document.

- Frame relay Local Management Interface (LMI) is terminated locally in the PE connected to the frame relay attachment circuit.

- The support of PVC link integrity check is outside the scope of this document.

6. General Encapsulation Method

The general frame relay pseudowire packet format for carrying frame relay information (user's payload and frame relay control information) between two PEs is shown in Figure 2.

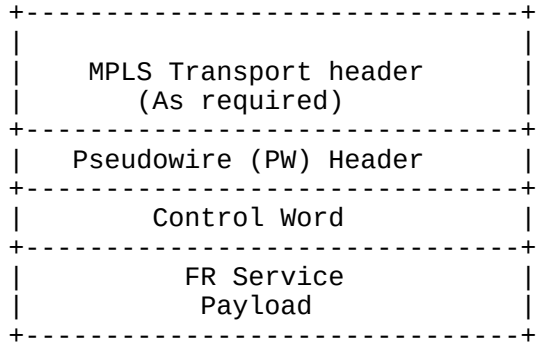


Figure 2. General format of frame relay encapsulation over PSN

The PW packet consists of the following fields: Control word and Payload, preceded by the MPLS Transport and pseudowire header. The meaning of the different fields is as follows:

-i. MPLS Transport header is specific to the MPLS network. This header is used to switch the PW packet through the MPLS core.

-ii. PW header contains an identifier for multiplexing PWs within an MPLS tunnel.

-iii. Control Word contains protocol control information for providing a frame relay service. Its structure is provided in the following sections.

-iv. The content of the frame relay service payload field depends on the mapping mode. In general it contains the layer 2 frame relay frame.

7. Frame Relay over MPLS PSN for the One-to-One Mode

7.1. MPLS PSN Tunnel and PW

MPLS label switched paths (LSPs) called "MPLS Tunnels" are used between PEs and are used within the MPLS core network to forward PW packets. An MPLS tunnel corresponds to "PSN Tunnel" of Figure 1.

Several PWs may be nested inside one MPLS tunnel. Each PW carries the traffic of a single frame relay VC. In this case, the PW header is an MPLS label called the PW label.

7.2. Packet Format over MPLS PSN

For the one-to-one mapping mode for frame relay over an MPLS network, the PW packet format is as shown in Figure 3.

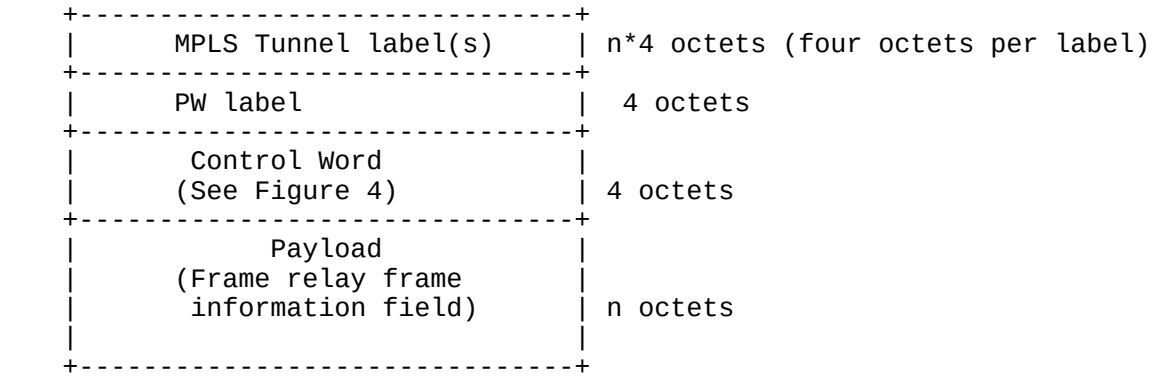


Figure 3. Frame Relay over MPLS PSN Packet for the One-to-One Mapping

The meaning of the different fields is as follows:

- MPLS Tunnel label(s)

  The MPLS Tunnel label(s) corresponds to the MPLS transport header of Figure 2. The label(s) is/are used by MPLS LSRs to forward a PW packet from one PE to the other.

- PW Label

  The PW label identifies one PW (i.e., one LSP) assigned to a frame relay VC in one direction. It corresponds to the PW header of Figure 2. Together the MPLS Tunnel label(s) and PW label form an MPLS label stack [RFC3032].

- Control Word

  The Control Word contains protocol control information. Its structure is shown in Figure 4.

- Payload

  The payload field corresponds to X.36/X.76 frame relay frame information field with the following components removed: bit/byte stuffing, frame relay header, and FCS. It is RECOMMENDED to support a frame size of at least 1600 bytes. The maximum length of the payload field MUST be agreed upon by the two PEs. This can be achieved by using the MTU interface parameter when the PW is established. [RFC4447]

7.3. The Control Word

The control word defined below is REQUIRED for frame relay one-to-one mode. The control word carries certain frame relay specific information that is necessary to regenerate the frame relay frame on the egress PE. Additionally, the control word also carries a sequence number that can be used to preserve sequentiality when carrying frame relay over an MPLS network. Its structure is as follows:

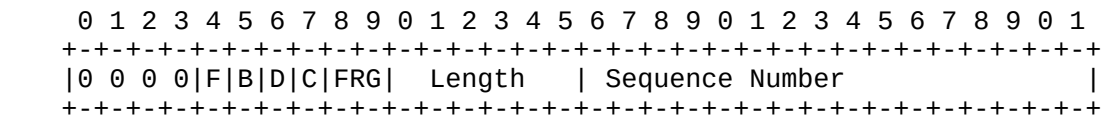


Figure 4. Control Word structure for the one-to-one mapping mode

The meaning of the Control Word fields (Figure 4) is as follows (see also [X36 and X76] for frame relay bits):

- Bits 0 to 3

  In the above diagram, the first 4 bits MUST be set to 0 to indicate PW data.

- F (bit 4) FR FECN (Forward Explicit Congestion Notification) bit.

- B (bit 5) FR BECN (Backward Explicit Congestion Notification) bit.

- D (bit 6) FR DE bit (Discard Eligibility) bit.

- C (bit 7) FR frame C/R (Command/Response) bit.

- FRG (bits 8 and 9): These bits are defined by [RFC4623].

- Length (bits 10 to 15)

  If the PW traverses a network link that requires a minimum frame size (a notable example is Ethernet), padding is required to reach its minimum frame size. If the frame's length (defined as the length of the layer 2 payload plus the length of the control word) is less than 64 octets, the length field MUST be set to the PW payload length. Otherwise, the length field MUST be set to zero. The value of the length field, if non-zero, is used to remove the padding characters by the egress PE.

- Sequence number (Bit 16 to 31)

  Sequence numbers provide one possible mechanism to ensure the ordered delivery of PW packets. The processing of the sequence number field is OPTIONAL. The sequence number space is a 16-bit unsigned circular space. The sequence number value 0 is used to indicate that the sequence number check algorithm is not used.

7.4. The Martini Legacy Mode Control Word

For backward compatibility to existing implementations, the following version of the control word is defined as the "martini mode CW" for frame relay.

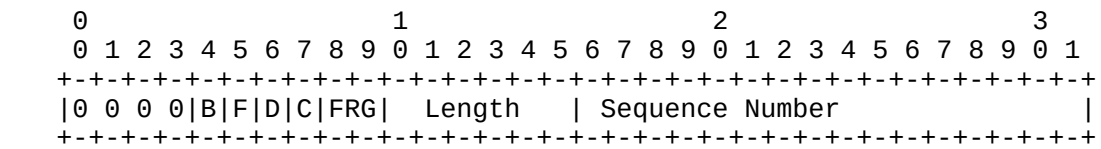


Figure 5. Control Word structure for the frame relay martini mode

Note that the "B" and "F" bits are reversed.

This control word format is used for PW type "Frame Relay DLCI (Martini Mode)"

7.5. PW Packet Processing

7.5.1. Encapsulation of Frame Relay Frames

The encapsulation process of a frame relay frame is initiated when a PE receives a frame relay frame from one of its frame relay UNI or NNI [FRF1] [FRF2] interfaces. The PE generates the following fields

of the control word from the corresponding fields of the frame relay frame as follows:

- Command/Response (C/R or C) bit: The C bit is copied unchanged in the PW Control Word.

- The DE bit of the frame relay frame is copied into the D bit field. However, if the D bit is not already set, it MAY be set as a result of ingress frame policing. If it is not already set by the copy operation, setting of this bit by a PE is OPTIONAL. The PE MUST NOT clear this bit (set it to 0 if it was received with the value of 1).

- The FECN bit of the frame relay frame is copied into the F bit field. However, if the F bit is not already set, it MAY be set to reflect a congestion situation detected by the PE. If it is not already set by the copy operation, setting of this bit by a PE is OPTIONAL. The PE MUST NOT clear this bit (set it to 0 if it was received with the value of 1)

- The BECN bit of the frame relay frame is copied into the B bit field. However, if the B bit is not already set, it MAY be set to reflect a congestion situation detected by the PE. If it is not already set by the copy operation, setting of this bit by a PE is OPTIONAL. The PE MUST NOT clear this bit (set it to 0 if it was received with the value of 1).

- If the PW packet length (defined as the length of the payload plus the length of the control word) is less than 64 octets, the length field MUST be set to the packet’s length. Otherwise, the length field MUST be set to zero.

- The sequence number field is processed if the PW uses sequence numbers. [RFC4385]

- The payload of the PW packet is the contents of ITU-T Recommendations X.36/X.76 [X36] [X76] frame relay frame information field stripped from any bit or byte stuffing.

7.5.2. Setting the Sequence Number

For a given PW and a pair of routers PE1 and PE2, if PE1 supports packet sequencing, then the procedures in [RFC4385], Section 4.1, MUST be followed.

7.6. Decapsulation of PW Packets

When a PE receives a PW packet, it processes the different fields of the control word in order to decapsulate the frame relay frame for transmission to a CE on a frame relay UNI or NNI. The PE performs the following actions (not necessarily in the order shown):

- It generates the following frame relay frame header fields from the corresponding fields of the PW packet.

- The C/R bit MUST be copied in the frame relay header.

- The D bit MUST be copied into the frame relay header DE bit.

- The F bit MUST be copied into the frame relay header FECN bit. If the F bit is set to zero, the FECN bit may be set to one, depending on the congestion state of the PE device in the forward direction. Changing the state of this bit by a PE is OPTIONAL.

- The B bit MUST be copied into the frame relay header BECN bit. If the B bit is set to zero, the BECN bit may be set to one, depending on the congestion state of the PE device in the backward direction. Changing the state of this bit by a PE is OPTIONAL.

- It processes the length and sequence field, the details of which are in the following sub-sections.

- It copies the frame relay information field from the contents of the PW packet payload after removing any padding.

Once the above fields of a FR frame have been processed, the standard HDLC operations are performed on the frame relay frame: the HDLC header is added, any bit or byte stuffing is added as required, and the FCS is also appended to the frame. The FR frame is then queued for transmission on the selected frame relay UNI or NNI interface.

7.6.1. Processing the Sequence Number

If a router PE2 supports received sequence number processing, then the procedures in [RFC4385], Section 4.2, MUST be used.

7.6.2. Processing of the Length Field by the Receiver

Any padding octet, if present, in the payload field of a PW packet received MUST be removed before forwarding the data.

- If the Length field is set to zero, then there are no padding octets following the payload field.

- Otherwise, if the payload is longer, then the length specified in the control word padding characters are removed according to the length field.

7.7. MPLS Shim EXP Bit Values

If it is desired to carry Quality of Service information, the Quality of Service information SHOULD be represented in the Experimental Use Bits (EXP) field of the PW MPLS label [RFC3032]. If more than one MPLS label is imposed by the ingress LSR, the EXP field of any labels higher in the stack SHOULD also carry the same value.

7.8. MPLS Shim S Bit Value

The ingress LSR, PE1, MUST set the S bit of the PW label to a value of 1 to denote that the PW label is at the bottom of the stack.

7.9. Control Plane Details for Frame Relay Service

The PE MUST provide frame relay PVC status signaling to the frame relay network. If the PE detects a service-affecting condition for a particular DLCI, as defined in [Q933] Q.933, Annex A.5, sited in IA FRF1.1, the PE MUST communicate to the remote PE the status of the PW that corresponds to the frame relay DLCI status. The Egress PE SHOULD generate the corresponding errors and alarms as defined in [Q922] [Q933] on the egress Frame relay PVC.

There are two frame relay flags to control word bit mappings described below. The legacy bit ordering scheme will be used for a PW of type 0x0001, "Frame Relay DLCI (Martini Mode)", and the new bit ordering scheme will be used for a PW of type 0x0019, "Frame Relay DLCI". The IANA allocation registry of "Pseudowire Type" is defined in [RFC4446] along with initial allocated values.

7.9.1. Frame Relay Specific Interface Parameter Sub-TLV

A separate document, [RFC4447], describes the PW control and maintenance protocol in detail, including generic interface parameter sub-TLVs. The interface parameter information, when applicable, MUST be used to validate that the PEs and the ingress and egress ports at the edges of the circuit have the necessary capabilities to interoperate with each other. The Interface parameter TLV is defined in [RFC4447], and the IANA registry with initial values for interface parameter sub-TLV types is defined in [RFC4446], but the frame relay specific interface parameter sub-TLV types are specified as follows:

- 0x08 Frame Relay Header Length Sub-TLV

An optional 16-bit value indicating the length of the FR Header, expressed in octets. This OPTIONAL interface parameter Sub-TLV can have value of 2, 3, or 4, the default being 2. If this Sub-TLV is not present, the default value of 2 is assumed.

8. Frame Relay Port Mode

The frame relay port mode PW shares the same encapsulation as the HDLC PW and is described in the respective document. [RFC4618]

9. Congestion Control

As explained in [RFC3985], the PSN carrying the PW may be subject to congestion, the characteristics of which depend on PSN type, network architecture, configuration, and loading. During congestion, the PSN may exhibit packet loss that will impact the service carried by the frame relay PW. In addition, since frame relay PWs carry a variety of services across the PSN, including but not restricted to TCP/IP, they may or may not behave in a TCP-friendly manner prescribed by [RFC2914]. In the presence of services that reduce transmission rate, frame relay PWs may thus consume more than their fair share and in that case SHOULD be halted.

Whenever possible, frame relay PWs should be run over traffic-engineered PSNs providing bandwidth allocation and admission control mechanisms. IntServ-enabled domains providing the Guaranteed Service (GS) or DiffServ-enabled domains using EF (expedited forwarding) are examples of traffic-engineered PSNs. Such PSNs will minimize loss and delay while providing some degree of isolation of the frame relay PW’s effects from neighboring streams.

Note that when transporting frame relay, DiffServ-enabled domains may use AF (Assured Forwarding) and/or DF (Default Forwarding) instead of EF, in order to place less burden on the network and to gain additional statistical multiplexing advantage. In particular, if the Committed Information Rate (CIR) of a frame relay VC is zero, then it is equivalent to a best-effort UDP over IP stream regarding congestion: the network is free to drop frames as necessary. In this case, the "DF" Per Hop Behavior (PHB) would be appropriate in a diff-serv-TE domain. Alternatively, if the CIR of a frame relay VC is nonzero and the DE bit is zero in the FR header, then "AF31" would be appropriate to be used, and if the CIR of a frame relay VC is nonzero but the DE bit is on, then "AF32" would be appropriate [RFC3270].

The PEs SHOULD monitor for congestion (by using explicit congestion notification, [VCCV], or by measuring packet loss) in order to ensure that the service using the frame relay PW may be maintained. When a

PE detects significant congestion while receiving the PW PDUs, the BECN bits of the frame relay frame transmitted on the same PW SHOULD be set to notify the remote PE and the remote frame relay switch of the congestion situation. In addition, the FECN bits SHOULD be set in the FR frames sent out the attachment circuit, to give the FR DTE a chance to adjust its transport layer advertised window, if possible.

If the PW has been set up using the protocol defined in [RFC4447], then procedures specified in [RFC4447] for status notification can be used to disable packet transmission on the ingress PE from the egress PE. The PW may be restarted by manual intervention, or by automatic means after an appropriate waiting time.

10. Security Considerations

PWE3 provides no means of protecting the contents or delivery of the PW packets on behalf of the native service. PWE3 may, however, leverage security mechanisms provided by the MPLS Tunnel Layer. A more detailed discussion of PW security is given in [RFC3985, RFC4447, RFC3916].

11. Normative References

[RFC4447] Martini, L., Rosen, E., El-Aawar, N., Smith, T., and G. Heron, "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", RFC 4447, April 2006.

[RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, February 2006.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.

[RFC4446] Martini, L., "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)", BCP 116, RFC 4446, April 2006.

[RFC4618] Martini, L., Rosen, E., Heron, G., and A. Malis, "Encapsulation Methods for Transport of Point to Point Protocol/High-Level Data Link Control (PPP/HDLC) over Multiprotocol Label Switching (MPLS) Networks", RFC 4618, September 2006.

[RFC4623] Malis, A. and M. Townsley, "Pseudowire Emulation Edge-to-Edge (PWE3) Fragmentation and Reassembly", RFC 4623, August 2006.

12. Informative References

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

[VCCV] Nadeau, T., et al., "Pseudo Wire Virtual Circuit Connection Verification (VCCV)", Work in Progress, October 2005.

[ATM] Martini, L., et al., "Encapsulation Methods for Transport of ATM Over MPLS Networks", Work in Progress, April 2005.

[RFC4448] Martini, L., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, April 2006.

[FRF1] FRF.1.2, Frame relay PVC UNI Implementation Agreement, Frame Relay Forum, April 2000.

[FRF2] FRF.2.2, Frame relay PVC UNI Implementation Agreement, Frame Relay Forum, April 2002

[RFC3916] Xiao, X., McPherson, D., and P. Pate, "Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)", RFC 3916, September 2004.

[X36] ITU-T Recommendation X.36, Interface between a DTE and DCE for public data networks providing frame relay, Geneva, 2000.

[X76] ITU-T Recommendation X.76, Network-to-network interface between public data networks providing frame relay services, Geneva,2000

[Q922] ITU-T Recommendation Q.922 Specification for Frame Mode Basic call control, ITU Geneva 1995

[Q933] ITU-T Recommendation Q.933 Specification for Frame Mode Basic call control, ITU Geneva 2003

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

[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.

Contributing Author Information

Kireeti Kompella
Juniper Networks
1194 N. Mathilda Ave
Sunnyvale, CA 94089

EMail: kireeti@juniper.net

Giles Heron
Tellabs
Abbey Place
24-28 Easton Street
High Wycombe
Bucks
HP11 1NT
UK

EMail: giles.heron@tellabs.com

Rao Cherukuri
Juniper Networks
1194 N. Mathilda Ave
Sunnyvale, CA 94089

Dimitri Stratton Vlachos
Mazu Networks, Inc.
125 Cambridgepark Drive
Cambridge, MA 02140

EMail: d@mazunetworks.com

Chris Liljenstolpe
Alcatel
11600 Sallie Mae Dr.
9th Floor
Reston, VA 20193

EMail: chris.liljenstolpe@alcatel.com

Nasser El-Aawar
Level 3 Communications, LLC.
1025 Eldorado Blvd.
Broomfield, CO, 80021

EMail: nna@level3.net

Eric C. Rosen
Cisco Systems, Inc.
1414 Massachusetts Avenue
Boxborough, MA 01719

EMail: erosen@cisco.com

Dan Tappan
Cisco Systems, Inc.
1414 Massachusetts Avenue
Boxborough, MA 01719

EMail: tappan@cisco.com

Prayson Pate
Overture Networks, Inc.
507 Airport Boulevard
Morrisville, NC, USA 27560

EMail: prayson.pate@overturenetworks.com

David Sinicrope
Ericsson IPI

EMail: david.sinicrope@ericsson.com

Ravi Bhat
Nokia

EMail: ravi.bhat@nokia.com

Nishit Vasavada
Nokia

EMail: nishit.vasavada@nokia.com

Steve Vogelsang
ECI Telecom
Omega Corporate Center
1300 Omega Drive
Pittsburgh, PA 15205

EMail: stephen.vogelsang@ecitele.com

Vinai Sirkay
Redback Networks
300 Holger Way,
San Jose, CA 95134

EMail: sirkay@technologist.com

Authors' Addresses

Luca Martini
Cisco Systems, Inc.
9155 East Nichols Avenue, Suite 400
Englewood, CO, 80112

EMail: lmartini@cisco.com

Claude Kawa
OZ Communications
Windsor Station
1100, de la Gauchetie're St West
Montreal QC Canada
H3B 2S2

EMail: claude.kawa@oz.com

Andrew G. Malis
Tellabs
1415 West Diehl Road
Naperville, IL 60563

EMail: Andy.Malis@tellabs.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

Exhibit 10

Network Working Group
Request for Comments: 5659
Category: Informational

M. Bocci
Alcatel-Lucent
S. Bryant
Cisco Systems

October 2009

An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge

Abstract

This document describes an architecture for extending pseudowire emulation across multiple packet switched network (PSN) segments. Scenarios are discussed where each segment of a given edge-to-edge emulated service spans a different provider's PSN, as are other scenarios where the emulated service originates and terminates on the same provider's PSN, but may pass through several PSN tunnel segments in that PSN. It presents an architectural framework for such multi-segment pseudowires, defines terminology, and specifies the various protocol elements and their functions.

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright and License Notice

Table of Contents

1. Introduction

RFC 3985 [1] defines the architecture for pseudowires, where a pseudowire (PW) both originates and terminates on the edge of the same packet switched network (PSN). The PW label is unchanged between the originating and terminating provider edges (PEs). This is now known as a single-segment pseudowire (SS-PW).

This document extends the architecture in RFC 3985 to enable point-to-point pseudowires to be extended through multiple PSN tunnels. These are known as multi-segment pseudowires (MS-PWs). Use cases for multi-segment pseudowires (MS-PWs), and the consequent requirements, are defined in RFC 5254 [5].

1.1. Motivation and Context

RFC 3985 addresses the case where a PW spans a single segment between two PEs. Such PWs are termed single-segment pseudowires (SS-PWs) and provide point-to-point connectivity between two edges of a provider network. However, there is now a requirement to be able to construct multi-segment pseudowires. These requirements are specified in RFC 5254 [5] and address three main problems:

i. How to constrain the density of the mesh of PSN tunnels when the number of PEs grows to many hundreds or thousands, while minimizing the complexity of the PEs and P-routers.

ii. How to provide PWs across multiple PSN routing domains or areas in the same provider.

iii. How to provide PWs across multiple provider domains and different PSN types.

Consider a single PW domain, such as that shown in Figure 1. There are 4 PEs, and PWs must be provided from any PE to any other PE. PWs can be supported by establishing a full mesh of PSN tunnels between the PEs, requiring a full mesh of LDP signaling adjacencies between the PEs. PWs can therefore be established between any PE and any other PE via a single, direct PSN tunnel that is switched only by intermediate P-routers (not shown in the figure). In this case, each PW is an SS-PW. A PE must terminate all the pseudowires that are carried on the PSN tunnels that terminate on that PE, according to the architecture of RFC 3985. This solution is adequate for small numbers of PEs, but the number of PEs, PSN tunnels, and signaling adjacencies will grow in proportion to the square of the number of PEs.

For reasons of economy, the edge PEs that terminate the attachment circuits (ACs) are often small devices built to very low cost with limited processing power. Consider an example where a particular PE, residing at the edge of a provider network, terminates N PWs to/from N different remote PEs. This needs N PW signaling adjacencies to be set up and maintained. If the edge PE attaches to a single intermediate PE that is able to switch the PW, that edge PE only needs a single adjacency to signal and maintain all N PWs. The intermediate switching PE (which is a larger device) needs M signaling adjacencies, but statistically this is less than tN, where t is the number of edge PEs that it is serving. Similarly, if the PWs are running over TE PSN tunnels, there is a statistical reduction in the number of TE PSN tunnels that need to be set up and maintained between the various PEs.

One possible solution that is more efficient for large numbers of PEs, in particular for the control plane, is therefore to support a partial mesh of PSN tunnels between the PEs, as shown in Figure 1. For example, consider a PW service whose endpoints are PE1 and PE4. Pseudowires for this can take the path PE1->PE2->PE4 and, rather than terminating at PE2, be switched between ingress and egress PSN tunnels on that PE. This requires a capability in PE2 that can concatenate PW segments PE1-PE2 to PW segments PE2-PE4. The end-to-end PW is known as a multi-segment PW.

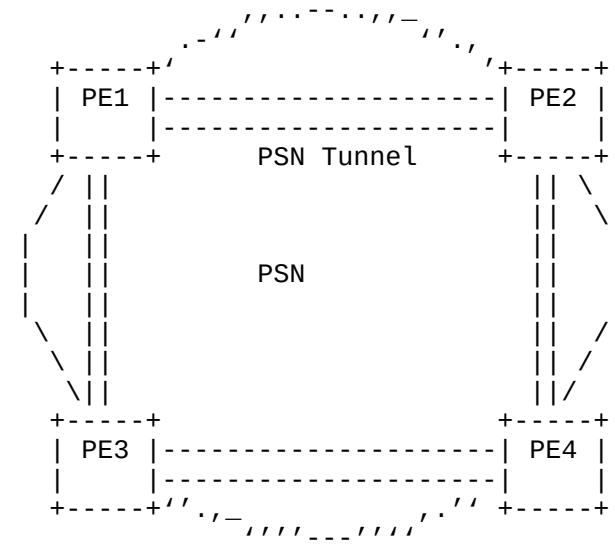


Figure 1: PWs Spanning a Single PSN with Partial Mesh of PSN Tunnels

Figure 1 shows a simple, flat PSN topology. However, large provider networks are typically not flat, consisting of many domains that are connected together to provide edge-to-edge services. The elements in each domain are specialized for a particular role, for example, supporting different PSN types or using different routing protocols.

An example application is shown in Figure 2. Here, the provider's network is divided into three domains: two access domains and the core domain. The access domains represent the edge of the provider's network at which services are delivered. In the access domain, simplicity is required in order to minimize the cost of the network. The core domain must support all of the aggregated services from the access domains, and the design requirements here are for scalability, performance, and information hiding (i.e., minimal state). The core must not be exposed to the state associated with large numbers of individual edge-to-edge flows. That is, the core must be simple and fast.

In a traditional layer 2 network, the interconnection points between the domains are where services in the access domains are aggregated for transport across the core to other access domains. In an IP network, the interconnection points could also represent interworking points between different types of IP networks, e.g., those with MPLS and those without, and points where network policies can be applied.

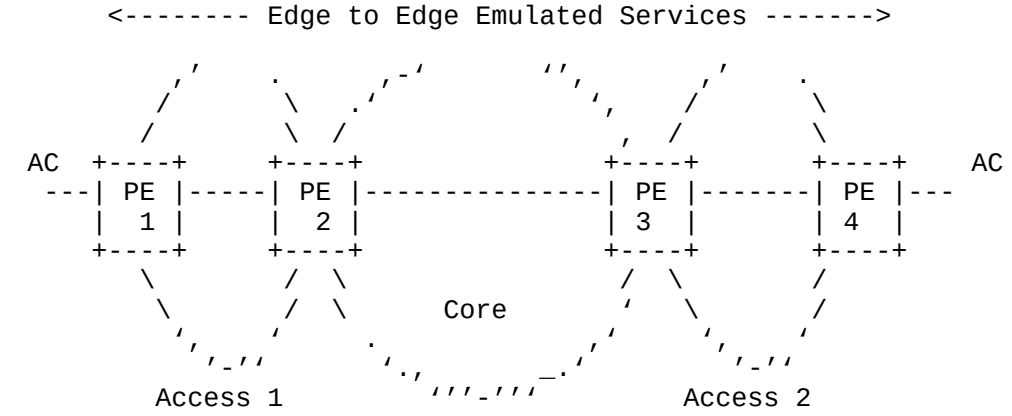


Figure 2: Multi-Domain Network Model

A similar model can also be applied to inter-provider services, where a single PW spans a number of separate provider networks in order to connect ACs residing on PEs in disparate provider networks. In this case, each provider will typically maintain their own PE at the border of their network in order to apply policies such as security

and Quality of Service (QoS) to PWs entering their network. Thus, the connection between the domains will normally be a link between two PEs on the border of each provider's network.

Consider the application of this model to PWs. PWs use tunneling mechanisms such as MPLS to enable the underlying PSN to emulate characteristics of the native service. One solution to the multi-domain network model above is to extend PSN tunnels edge-to-edge between all of the PEs in access domain 1 and all of the PEs in access domain 2, but this requires a large number of PSN tunnels, as described above, and also exposes the access and the core of the network to undesirable complexity. An alternative is to constrain the complexity to the network domain interconnection points (PE2 and PE3 in the example above). Pseudowires between PE1 and PE4 would then be switched between PSN tunnels at the interconnection points, enabling PWs from many PEs in the access domains to be aggregated across only a few PSN tunnels in the core of the network. PEs in the access domains would only need to maintain direct signaling sessions and PSN tunnels, with other PEs in their own domain, thus minimizing complexity of the access domains.

1.2. Non-Goals of This Document

The following are non-goals for this document:

- The on-the-wire specification of PW encapsulations.
- The detailed specification of mechanisms for establishing and maintaining multi-segment pseudowires.

1.3. Terminology

The terminology specified in RFC 3985 [1] and RFC 4026 [2] applies. In addition, we define the following terms:

- PW Terminating Provider Edge (T-PE). A PE where the customer-facing attachment circuits (ACs) are bound to a PW forwarder. A terminating PE is present in the first and last segments of an MS-PW. This incorporates the functionality of a PE as defined in RFC 3985.
- Single-Segment Pseudowire (SS-PW). A PW set up directly between two T-PE devices. The PW label is unchanged between the originating and terminating T-PEs.

- Multi-Segment Pseudowire (MS-PW). A static or dynamically configured set of two or more contiguous PW segments that behave and function as a single point-to-point PW. Each end of an MS-PW, by definition, terminates on a T-PE.

- PW Segment. A part of a single-segment or multi-segment PW, which traverses one PSN tunnel in each direction between two PE devices, T-PEs, and/or S-PEs (switching PE).

- PW Switching Provider Edge (S-PE). A PE capable of switching the control and data planes of the preceding and succeeding PW segments in an MS-PW. The S-PE terminates the PSN tunnels of the preceding and succeeding segments of the MS-PW. It therefore includes a PW switching point for an MS-PW. A PW switching point is never the S-PE and the T-PE for the same MS-PW. A PW switching point runs necessary protocols to set up and manage PW segments with other PW switching points and terminating PEs. An S-PE can exist anywhere a PW must be processed or policy applied. It is therefore not limited to the edge of a provider network.

  Note that it was originally anticipated that S-PEs would only be deployed at the edge of a provider network where they would be used to switch the PWs of different service providers. However, as the design of MS-PW progressed, other applications for MS-PW were recognized. By this time S-PE had become the accepted term for the equipment, even though they were no longer universally deployed at the provider edge.

- PW Switching. The process of switching the control and data planes of the preceding and succeeding PW segments in a MS-PW.

- PW Switching Point. The reference point in an S-PE where the switching takes place, e.g., where PW label swap is executed.

- Eligible S-PE or T-PE. An eligible S-PE or T-PE is a PE that meets the security and privacy requirements of the MS-PW, according to the network operator’s policy.

- Trusted S-PE or T-PE. A trusted S-PE or T-PE is a PE that is understood to be eligible by its next-hop S-PE or T-PE, while a trust relationship exists between two S-PEs or T-PEs if they mutually consider each other to be eligible.

2. Applicability

An MS-PW is a single PW that, for technical or administrative reasons, is segmented into a number of concatenated hops. From the perspective of a Layer 2 Virtual Private Network (L2VPN), an MS-PW is indistinguishable from an SS-PW. Thus, the following are equivalent from the perspective of the T-PE:

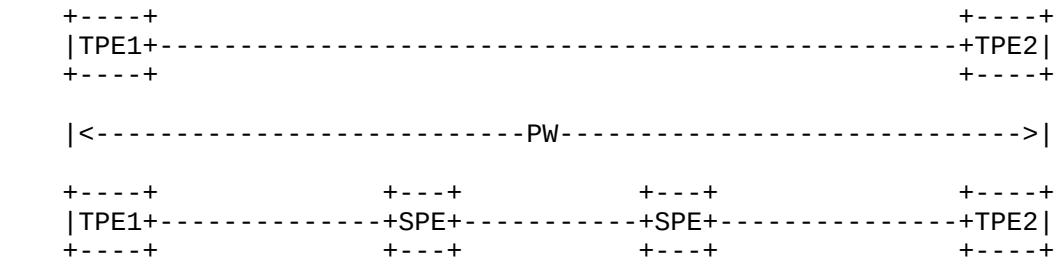


Figure 3: MS-PW Equivalence

Although an MS-PW may require services such as node discovery and path signaling to construct the PW, it should not be confused with an L2VPN system, which also requires these services. A Virtual Private Wire Service (VPWS) connects its endpoints via a set of PWs. MS-PW is a mechanism that abstracts the construction of complex PWs from the construction of a L2VPN. Thus, a T-PE might be an edge device optimized for simplicity and an S-PE might be an aggregation device designed to absorb the complexity of continuing the PW across the core of one or more service provider networks to another T-PE located at the edge of the network.

As well as supporting traditional L2VPNs, an MS-PW is applicable to providing connectivity across a transport network based on packet switching technology, e.g., the MPLS Transport Profile (MPLS-TP) [6], [8]. Such a network uses pseudowires to support the transport and aggregation of all services. This application requires deterministic characteristics and behavior from the network. The operational requirements of such networks may need pseudowire segments that can be established and maintained in the absence of a control plane, and may also need the operational independence of PW maintenance from the underlying PSN.

3. Protocol Layering Model

The protocol layering model specified in RFC 3985 applies to MS-PWs with the following clarification: the pseudowires may be considered to be a separate layer to the PSN tunnel. That is, although a PW segment will follow the path of the PSN tunnel between S-PEs, the

MS-PW is independent of the PSN tunnel routing, operations, signaling, and maintenance. The design of PW routing domains should not imply that the underlying PSN routing domains are the same. However, MS-PWs will reuse the protocols of the PSN and may, if applicable, use information that is extracted from the PSN, e.g., reachability.

3.1. Domain of MS-PW Solutions

PWs provide the Encapsulation Layer, i.e., the method of carrying various payload types, and the interface to the PW Demultiplexer Layer. Other layers provide the following:

- PSN tunnel setup, maintenance, and routing
- T-PE discovery

Not all PEs may be capable of providing S-PE functionality. Connectivity to the next-hop S-PE or T-PE must be provided by a PSN tunnel, according to [1]. The selection of which set of S-PEs to use to reach a given T-PE is considered to be within the scope of MS-PW solutions.

3.2. Payload Types

MS-PWs are applicable to all PW payload types. Encapsulations defined for SS-PWs are also used for MS-PW without change. Where the PSN types for each segment of an MS-PW are identical, the PW types of each segment must also be identical. However, if different segments run over different PSN types, the encapsulation may change but the PW segments must be of an equivalent PW type, i.e., the S-PE must not need to process the PW payload to provide translation.

4. Multi-Segment Pseudowire Reference Model

The pseudowire emulation edge-to-edge (PWE3) reference architecture for the single-segment case is shown in [1]. This architecture applies to the case where a PSN tunnel extends between two edges of a single PSN domain to transport a PW with endpoints at these edges.

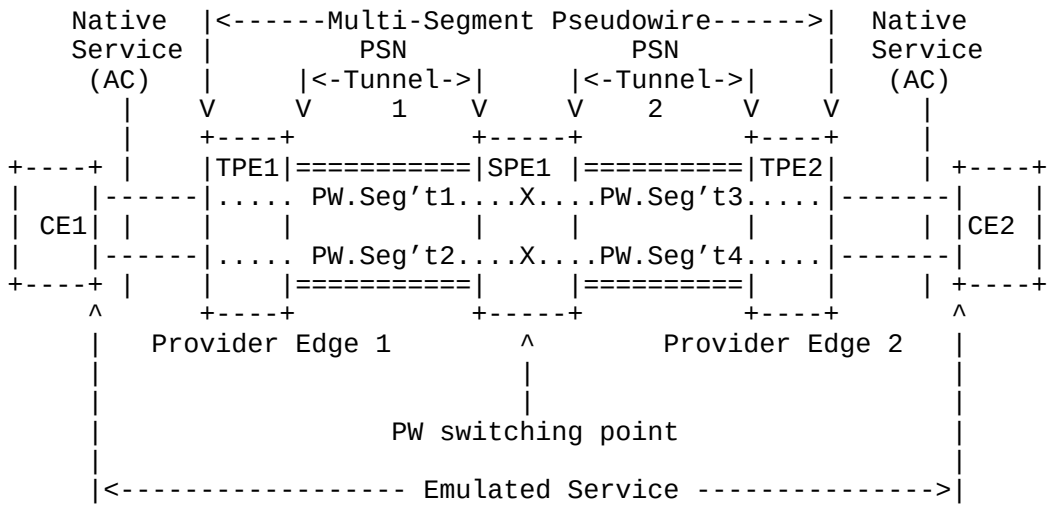


Figure 4: MS-PW Reference Model

Figure 4 extends this architecture to show a multi-segment case. The PEs that provide services to CE1 and CE2 are Terminating PE1 (T-PE1) and Terminating PE2 (T-PE2), respectively. A PSN tunnel extends from T-PE1 to Switching PE1 (S-PE1) across PSN1, and a second PSN tunnel extends from S-PE1 to T-PE2 across PSN2. PWs are used to connect the attachment circuits (ACs) attached to PE1 to the corresponding ACs attached to T-PE2.

Each PW segment on the tunnel across PSN1 is switched to a PW segment in the tunnel across PSN2 at S-PE1 to complete the multi-segment PW (MS-PW) between T-PE1 and T-PE2. S-PE1 is therefore the PW switching point. PW segment 1 and PW segment 3 are segments of the same MS-PW, while PW segment 2 and PW segment 4 are segments of another MS-PW. PW segments of the same MS-PW (e.g., PW segment 1 and PW segment 3) must be of equivalent PW types, as described in Section 3.2, while PSN tunnels (e.g., PSN1 and PSN2) may be of the same or different PSN types. An S-PE switches an MS-PW from one segment to another based on the PW demultiplexer, i.e., a PW label that may take one of the forms defined in Section 5.4.1 of RFC 3985 [1].

Note that although Figure 4 only shows a single S-PE, a PW may transit more than one S-PE along its path. This architecture is applicable when the S-PEs are statically chosen, or when they are chosen using a dynamic path-selection mechanism. Both directions of an MS-PW must traverse the same set of S-PEs on a reciprocal path. Note that although the S-PE path is therefore reciprocal, the path taken by the PSN tunnels between the T-PEs and S-PEs might not be reciprocal due to choices made by the PSN routing protocol.

4.1. Intra-Provider Connectivity Architecture

There is a requirement to deploy PWs edge-to-edge in large service provider networks (RFC 5254 [5]). Such networks typically encompass hundreds or thousands of aggregation devices at the edge, each of which would be a PE. These networks may be partitioned into separate metro and core PW domains, where the PEs are interconnected by a sparse mesh of tunnels.

Whether or not the network is partitioned into separate PW domains, there is also a requirement to support a partial mesh of traffic-engineered PSN tunnels.

The architecture shown in Figure 4 can be used to support such cases. PSN1 and PSN2 may be in different administrative domains or access regions, core regions, or metro regions within the same provider's network. PSN1 and PSN2 may also be of different types. For example, S-PEs may be used to connect PW segments traversing metro networks of one technology, e.g., statically allocated labels, with segments traversing an MPLS core network.

Alternatively, T-PE1, S-PE1, and T-PE2 may reside at the edges of the same PSN.

4.1.1. Intra-Provider Switching Using ACs

In this model, the PW reverts to the native service AC at the domain boundary PE. This AC is then connected to a separate PW on the same PE. In this case, the reference models of RFC 3985 apply to each segment and to the PEs. The remaining PE architectural considerations in this document do not apply to this case.

4.1.2. Intra-Provider Switching Using PWs

In this model, PW segments are switched between PSN tunnels that span portions of a provider's network, without reverting to the native service at the boundary. For example, in Figure 4, PSN1 and PSN2 would be portions of the same provider's network.

4.2. Inter-Provider Connectivity Architecture

Inter-provider PWs may need to be switched between PSN tunnels at the provider boundary in order to minimize the number of tunnels required to provide PW-based services to CEs attached to each provider's network. In addition, the following may need to be implemented on a per-PW basis at the provider boundary:

- Operations, Administration, and Maintenance (OAM). Note that this is synonymous with 'Operations and Maintenance' referred to in RFC 5254 [5].
- Authentication, Authorization, and Accounting (AAA)
- Security mechanisms

Further security-related architectural considerations are described in Section 12.

4.2.1. Inter-Provider Switching Using ACs

In this model, the PW reverts to the native service at the provider boundary PE. This AC is then connected to a separate PW at the peer provider boundary PE. In this case, the reference models of RFC 3985 apply to each segment and to the PEs. This is similar to the case in Section 4.1.1, except that additional security and policy enforcement measures will be required. The remaining PE architectural considerations in this document do not apply to this case.

4.2.2. Inter-Provider Switching Using PWs

In this model, PW segments are switched between PSN tunnels in each provider's network, without reverting to the native service at the boundary. This architecture is shown in Figure 5. Here, S-PE1 and S-PE2 are provider border routers. PW segment 1 is switched to PW segment 2 at S-PE1. PW segment 2 is then carried across an inter-provider PSN tunnel to S-PE2, where it is switched to PW segment 3 in PSN2.

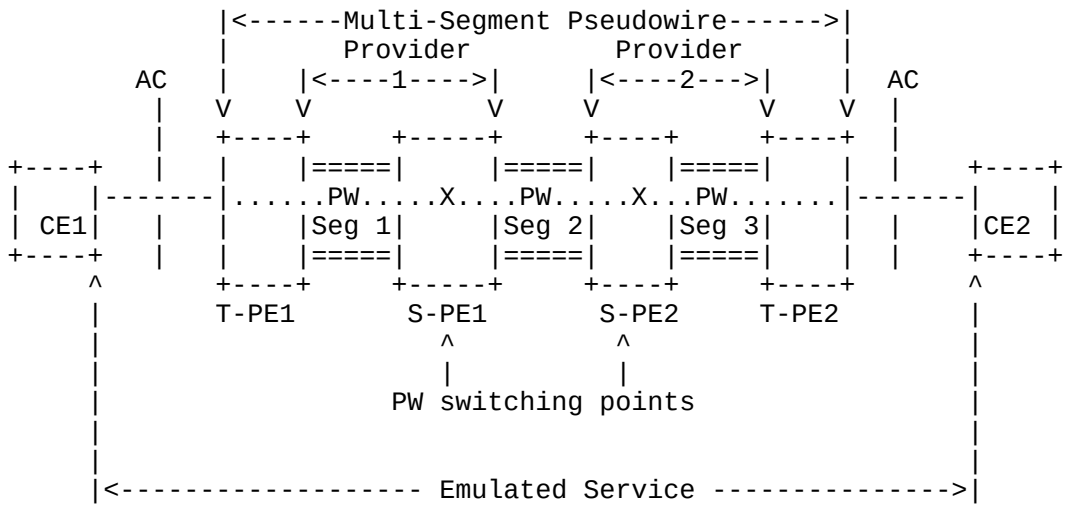


Figure 5: Inter-Provider Reference Model

5. PE Reference Model

5.1. Pseudowire Pre-Processing

Pseudowire pre-processing is applied in the T-PEs as specified in RFC 3985. Processing at the S-PEs is specified in the following sections.

5.1.1. Forwarding

Each forwarder in the S-PE forwards packets from one PW segment on the ingress PSN-facing interface of the S-PE to one PW segment on the egress PSN-facing interface of the S-PE.

The forwarder selects the egress segment PW based on the ingress PW label. The mapping of ingress to egress PW label may be statically or dynamically configured. Figure 6 shows how a single forwarder is associated with each PW segment at the S-PE.

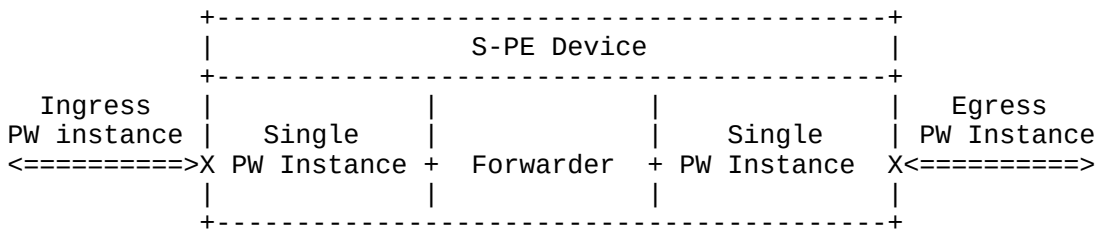


Figure 6: Point-to-Point Service

Other mappings of PW-to-forwarder are for further study.

5.1.2. Native Service Processing

There is no native service processing in the S-PEs.

6. Protocol Stack Reference Model

Figure 7 illustrates the protocol stack reference model for multi-segment PWs.

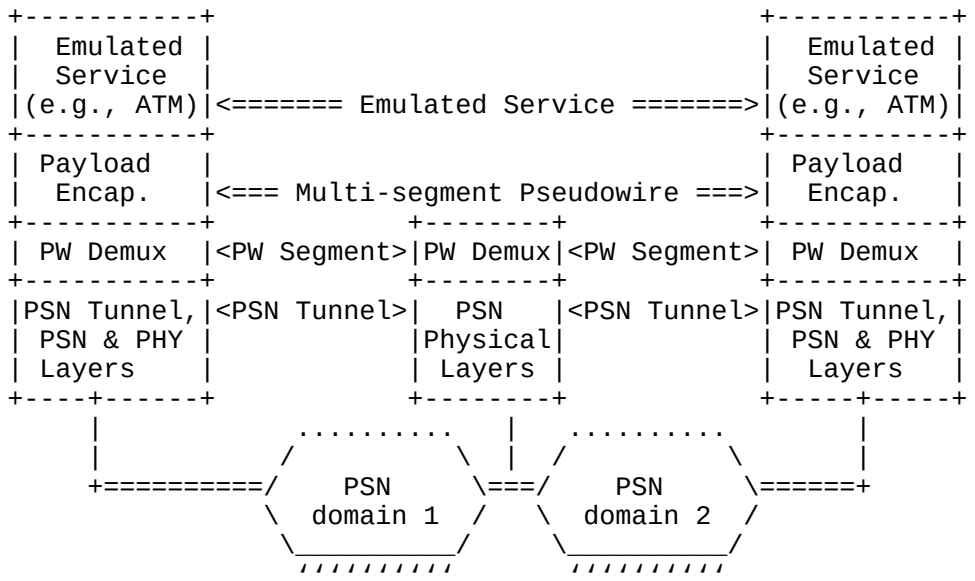


Figure 7: Multi-Segment PW Protocol Stack

The MS-PW provides the CE with an emulated physical or virtual connection to its peer at the far end. Native service PDUs from the CE are passed through an Encapsulation Layer and a PW demultiplexer

is added at the sending T-PE. The PDU is sent over PSN domain via the PSN transport tunnel. The receiving S-PE swaps the existing PW demultiplexer for the demultiplexer of the next segment and then sends the PDU over transport tunnel in PSN2. Where the ingress and egress PSN domains of the S-PE are of the same type, e.g., they are both MPLS PSNs, a simple label swap operation is performed, as described in Section 3.13 of RFC 3031 [3]. However, where the ingress and egress PSNs are of different types, e.g., MPLS and L2TPv3, the ingress PW demultiplexer is removed (or popped), and a mapping to the egress PW demultiplexer is performed and then inserted (or pushed).

Policies may also be applied to the PW at this point. Examples of such policies include admission control, rate control, QoS mappings, and security. The receiving T-PE removes the PW demultiplexer and restores the payload to its native format for transmission to the destination CE.

Where the encapsulation format is different, e.g., MPLS and L2TPv3, the payload encapsulation may be translated at the S-PE.

7. Maintenance Reference Model

Figure 8 shows the maintenance reference model for multi-segment pseudowires.

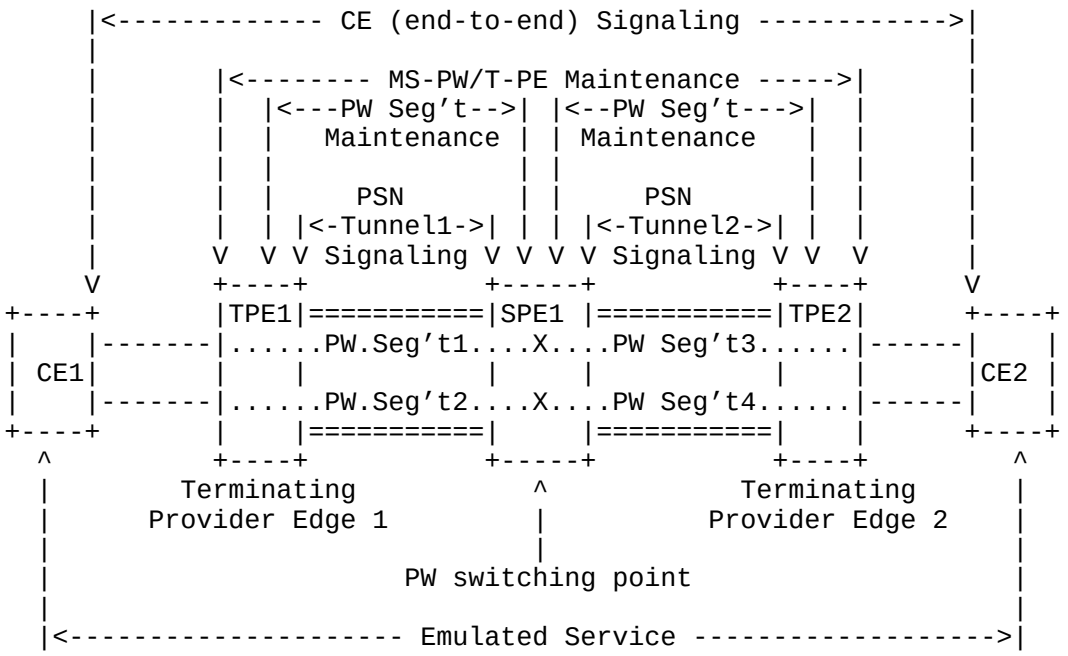


Figure 8: MS-PW Maintenance Reference Model

RFC 3985 specifies the use of CE (end-to-end) and PSN tunnel signaling as well as PW/PE maintenance. CE and PSN tunnel signaling is as specified in RFC 3985. However, in the case of MS-PWs, signaling between the PEs now has both an edge-to-edge and a hop-by-hop context. That is, signaling and maintenance between T-PEs and S-PEs and between adjacent S-PEs is used to set up, maintain, and tear down the MS-PW segments, which includes the coordination of parameters related to each switching point as well as to the MS-PW endpoints.

8. PW Demultiplexer Layer and PSN Requirements

8.1. Multiplexing

The purpose of the PW Demultiplexer Layer at the S-PE is to demultiplex PWs from ingress PSN tunnels and to multiplex them into egress PSN tunnels. Although each PW may contain multiple native service circuits, e.g., multiple ATM virtual circuits (VCs), the S-PEs do not have visibility of, and hence do not change, this level of multiplexing because they contain no Native Service Processor (NSP).

8.2. Fragmentation

If fragmentation is to be used in an MS-PW, T-PEs and S-PEs must satisfy themselves that fragmented PW payloads can be correctly reassembled for delivery to the destination attachment circuit.

An S-PE is not required to make any attempt to reassemble a fragmented PW payload. However, it may choose to do so if, for example, it knows that a downstream PW segment does not support reassembly.

An S-PE may fragment a PW payload using [4].

9. Control Plane

9.1. Setup and Placement of MS-PWs

For multi-segment pseudowires, the intermediate PW switching points may be statically provisioned or chosen dynamically.

For the static case, there are two options for exchanging the PW labels:

- By configuration at the T-PEs or S-PEs.
- By signaling across each segment using a dynamic maintenance protocol.

A multi-segment pseudowire may thus consist of segments where the labels are statically configured and segments where the labels are signaled.

For the case of dynamic choice of the PW switching points, there are two options for selecting the path of the MS-PW:

- T-PEs determine the full path of the PW through intermediate switching points. This may be either static or based on a dynamic PW path-selection mechanism.
- Each T-PE and S-PE makes a local decision as to which next-hop S-PE to choose to reach the target T-PE. This choice is made either using locally configured information or by using a dynamic PW path-selection mechanism.

9.2. Pseudowire Up/Down Notification

Since a multi-segment PW consists of a number of concatenated PW segments, the emulated service can only be considered as being up when all of the constituting PW segments and PSN tunnels are functional and operational along the entire path of the MS-PW.

If a native service requires bi-directional connectivity, the corresponding emulated service can only be signaled as being up when the PW segments and PSN tunnels (if used), are functional and operational in both directions.

RFC 3985 describes the architecture of failure and other status notification mechanisms for PWs. These mechanisms are also needed in multi-segment pseudowires. In addition, if a failure notification mechanism is provided for consecutive segments of the same PW, the S-PE must propagate such notifications between the consecutive concatenated segments.

9.3. Misconnection and Payload Type Mismatch

Misconnection and payload type mismatch can occur with PWs. Misconnection can breach the integrity of the system. Payload mismatch can disrupt the customer network. In both instances, there are security and operational concerns.

The services of the underlying tunneling mechanism or the PW control and OAM protocols can be used to ensure that the identity of the PW next hop is as expected. As part of the PW setup, a PW-TYPE identifier is exchanged. This is then used by the forwarder and the NSP of the T-PEs to verify the compatibility of the ACs. This can also be used by S-PEs to ensure that concatenated segments of a given MS-PW are compatible or that an MS-PW is not misconnected into a local AC. In addition, it is possible to perform an end-to-end connection verification to check the integrity of the PW, to verify the identity of S-PEs and check the correct connectivity at S-PEs, and to verify the identity of the T-PE.

10. Management and Monitoring

The management and monitoring as described in RFC 3985 applies here.

The MS-PW architecture introduces additional considerations related to management and monitoring, which need to be reflected in the design of maintenance tools and additional management objects for MS-PWs.

The first is that each S-PE is a new point at which defects may occur along the path of the PW. In order to troubleshoot MS-PWs, management and monitoring should be able to operate on a subset of the segments of an MS-PW, as well as edge-to-edge. That is, connectivity verification mechanisms should be able to troubleshoot and differentiate the connectivity between T-PEs and intermediate S-PEs, as well as the connectivity between T-PE and T-PE.

The second is that the set of S-PEs and P-routers along the MS-PW path may be less optimal than a path between the T-PEs chosen solely by the underlying PSN routing protocols. This is because the S-PEs are chosen by the MS-PW path selection mechanism and not by the PSN routing protocols. Troubleshooting mechanisms should therefore be provided to verify the set of S-PEs that are traversed by an MS-PW to reach a T-PE.

Some of the S-PEs and the T-PEs for an MS-PW may reside in a different service provider's PSN domain from that of the operator who initiated the establishment of the MS-PW. These situations may necessitate the use of remote management of the MS-PW, which is able to securely operate across provider boundaries.

11. Congestion Considerations

The following congestion considerations apply to MS-PWs. These are in addition to the considerations for PWs described in RFC 3985 [1], [7], and the respective RFCs specifying each PW type.

The control plane and the data plane fate-share in traditional IP networks. The implication of this is that congestion in the data plane can cause degradation of the operation of the control plane. Under quiescent operating conditions, it is expected that the network will be designed to avoid such problems. However, MS-PW mechanisms should also consider what happens when congestion does occur, when the network is stretched beyond its design limits, for example, during unexpected network failure conditions.

Although congestion within a single provider's network can be mitigated by suitable engineering of the network so that the traffic imposed by PWs can never cause congestion in the underlying PSN, a significant number of MS-PWs are expected to be deployed for inter-provider services. In this case, there may be no way of a provider who initiates the establishment of an MS-PW at a T-PE guaranteeing that it will not cause congestion in a downstream PSN. A specific PSN may be able to protect itself from excess PW traffic by policing all PWs at the S-PE at the provider border. However, this may not be

effective when the PSN tunnel across a provider utilizes the transit services of another provider that cannot distinguish PW traffic from ordinary, TCP-controlled IP traffic.

Each segment of an MS-PW therefore needs to implement congestion detection and congestion control mechanisms where it is not possible to explicitly provision sufficient capacity to avoid congestion.

In many cases, only the T-PEs may have sufficient information about each PW to fairly apply congestion control. Therefore, T-PEs need to be aware of which of their PWs are causing congestion in a downstream PSN and of their native service characteristics, and to apply congestion control accordingly. S-PEs therefore need to propagate PSN congestion state information between their downstream and upstream directions. If the MS-PW transits many S-PEs, it may take some time for congestion state information to propagate from the congested PSN segment to the source T-PE, thus delaying the application of congestion control. Congestion control in the S-PE at the border of the congested PSN can enable a more rapid response and thus potentially reduce the duration of congestion.

In addition to protecting the operation of the underlying PSN, consistent QoS and traffic engineering mechanisms should be used on each segment of an MS-PW to support the requirements of the emulated service. The QoS treatment given to a PW packet at an S-PE may be derived from context information of the PW (e.g., traffic or QoS parameters signaled to the S-PE by an MS-PW control protocol) or from PSN-specific QoS flags in the PSN tunnel label or PW demultiplexer, e.g., TC bits in either the label switched path (LSP) or PW label for an MPLS PSN or the DS field of the outer IP header for L2TPv3.

12. Security Considerations

The security considerations described in RFC 3985 [1] apply here. Detailed security requirements for MS-PWs are specified in RFC 5254 [5]. This section describes the architectural implications of those requirements.

The security implications for T-PEs are similar to those for PEs in single-segment pseudowires. However, S-PEs represent a point in the network where the PW label is exposed to additional processing. An S-PE or T-PE must trust that the context of the MS-PW is maintained by a downstream S-PE. OAM tools must be able to verify the identity of the far end T-PE to the satisfaction of the network operator. Additional consideration needs to be given to the security of the S-PEs, both at the data plane and the control plane, particularly when these are dynamically selected and/or when the MS-PW transits the networks of multiple operators.

An implicit trust relationship exists between the initiator of an MS-PW, the T-PEs, and the S-PEs along the MS-PW's path. That is, the T-PE trusts the S-PEs to process and switch PWs without compromising the security or privacy of the PW service. An S-PE should not select a next-hop S-PE or T-PE unless it knows it would be considered eligible, as defined in Section 1.3, by the originator of the MS-PW. For dynamically placed MS-PWs, this can be achieved by allowing the T-PE to explicitly specify the path of the MS-PW. When the MS-PW is dynamically created by the use of a signaling protocol, an S-PE or T-PE should determine the authenticity of the peer entity from which it receives the request and the compliance of that request with policy.

Where an MS-PW crosses a border between one provider and another provider, the MS-PW segment endpoints (S-PEs or T-PEs) or, for the PSN tunnel, P-routers typically reside on the same nodes as the Autonomous System Border Router (ASBRs) interconnecting the two providers. In either case, an S-PE in one provider is connected to a limited number of trusted T-PEs or S-PEs in the other provider. The number of such trusted T-PEs or S-PEs is bounded and not anticipated to create a scaling issue for the control plane authentication mechanisms.

Directly interconnecting the S-PEs/T-PEs using a physically secure link and enabling signaling and routing authentication between the S-PEs/T-PEs eliminates the possibility of receiving an MS-PW signaling message or packet from an untrusted peer. The S-PEs/T-PEs represent security policy enforcement points for the MS-PW, while the ASBRs represent security policy enforcement points for the provider's PSNs. This architecture is illustrated in Figure 9.

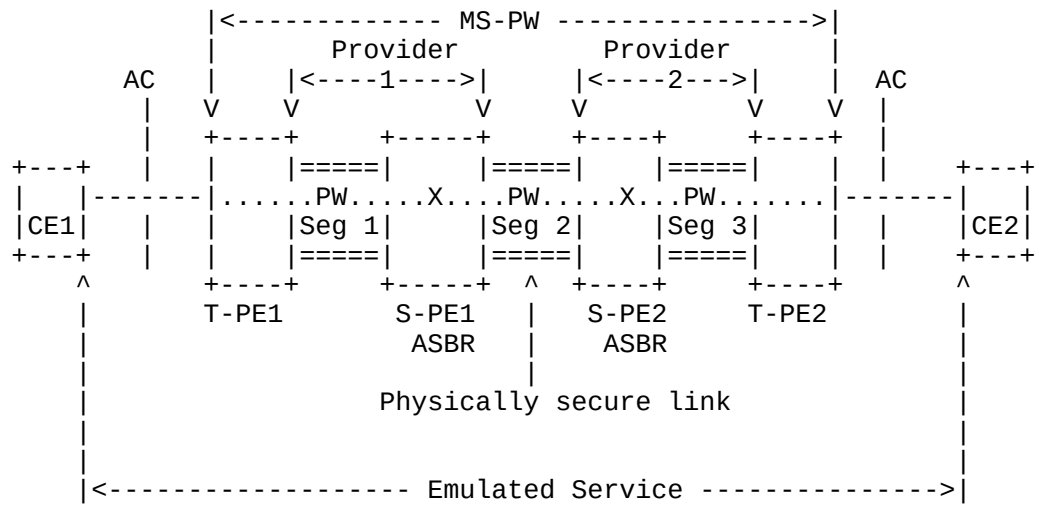


Figure 9: Directly Connected Inter-Provider Reference Model

Alternatively, the P-routers for the PSN tunnel may reside on the ASBRs, while the S-PEs or T-PEs reside behind the ASBRs within each provider's network. A limited number of trusted inter-provider PSN tunnels interconnect the provider networks. This is illustrated in Figure 10.

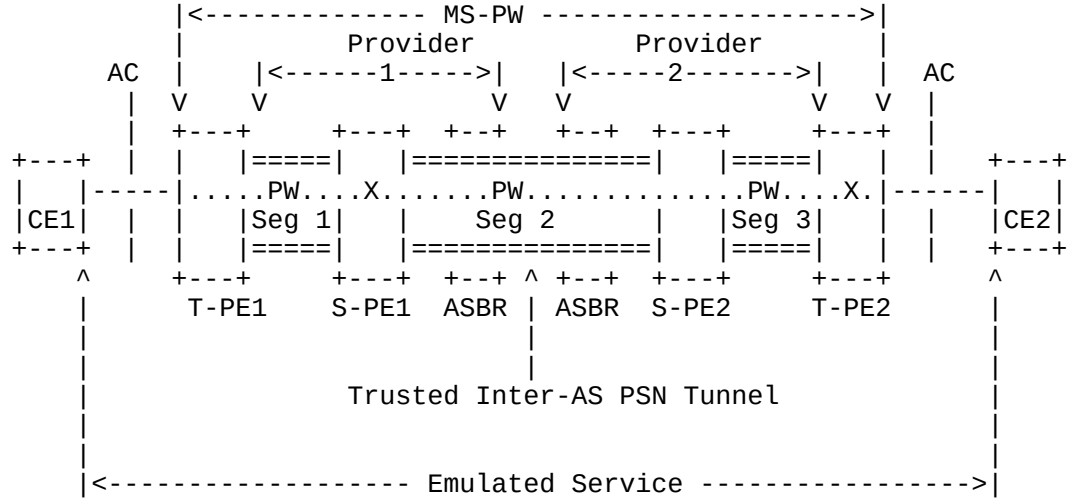


Figure 10: Indirectly Connected Inter-Provider Reference Model

Particular consideration needs to be given to Quality of Service requests because the inappropriate use of priority may impact any service guarantees given to other PWs. Consideration also needs to be given to the avoidance of spoofing the PW demultiplexer.

Where an S-PE provides interconnection between different providers, security considerations that are similar to the security considerations for ASBRs apply. In particular, peer entity authentication should be used.

Where an S-PE also supports T-PE functionality, mechanisms should be provided to ensure that MS-PWs are switched correctly to the appropriate outgoing PW segment, rather than to a local AC. Other mechanisms for PW endpoint verification may also be used to confirm the correct PW connection prior to enabling the attachment circuits.

13. Acknowledgments

The authors gratefully acknowledge the input of Mustapha Aissaoui, Dimitri Papadimitrou, Sasha Vainshtein, and Luca Martini.

14. References

14.1. Normative References

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

[2] Andersson, L. and T. Madsen, "Provider Provisioned Virtual Private Network (VPN) Terminology", RFC 4026, March 2005.

[3] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[4] Malis, A. and M. Townsley, "Pseudowire Emulation Edge-to-Edge (PWE3) Fragmentation and Reassembly", RFC 4623, August 2006.

14.2. Informative References

[5] Bitar, N., Ed., Bocci, M., Ed., and L. Martini, Ed., "Requirements for Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3)", RFC 5254, October 2008.

[6] Niven-Jenkins, B., Ed., Brungard, D., Ed., Betts, M., Ed., Sprecher, N., and S. Ueno, "Requirements of an MPLS Transport Profile", RFC 5654, September 2009.

[7] Bryant, S., Davie, B., Martini, L., and E. Rosen, "Pseudowire Congestion Control Framework", Work in Progress, June 2009.

[8] Bocci, M., Bryant, S., and L. Levrau, "A Framework for MPLS in Transport Networks", Work in Progress, August 2009.

Authors' Addresses

Matthew Bocci
Alcatel-Lucent
Voyager Place, Shoppenhangers Road,
Maidenhead, Berks, UK
Phone: +44 1633 413600
EMail: matthew.bocci@alcatel-lucent.com

Stewart Bryant
Cisco Systems
250, Longwater,
Green Park,
Reading, RG2 6GB,
United Kingdom
EMail: stbryant@cisco.com