

Network Working Group
Request for Comments: 4385
Category: Standards Track

S. Bryant
G. Swallow
L. Martini
Cisco Systems
D. McPherson
Arbor Networks
February 2006

Pseudowire Emulation Edge-to-Edge (PWE3)
Control Word for Use over an MPLS PSN

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

Copyright (C) The Internet Society (2006).

Abstract

This document describes the preferred design of a Pseudowire Emulation Edge-to-Edge (PWE3) Control Word to be used over an MPLS packet switched network, and the Pseudowire Associated Channel Header. The design of these fields is chosen so that an MPLS Label Switching Router performing MPLS payload inspection will not confuse a PWE3 payload with an IP payload.

1. Introduction

The standard MPLS encapsulations have no explicit protocol identifier. In order for a pseudowire (PW) [RFC3985] to operate correctly over an MPLS packet switched network (PSN) that performs MPLS payload inspection, a PW packet must not appear to a label switching router (LSR) as if it were an IP packet [BCP]. An example of an LSR that performs MPLS payload inspection is one that is performing equal-cost multiple-path load-balancing (ECMP) [RFC2992]. If ECMP were performed on PW packets, the packets in the PW may not all follow the same path through the PSN. This may result in misordered packet delivery to the egress PE. The inability to ensure that all packets belonging to a PW follow the same path may also prevent the PW Operations and Management (OAM) [VCCV] mechanism from correctly monitoring the PW.

This document specifies how the PW control word is used to distinguish a PW payload from an IP payload carried over an MPLS PSN. It then describes the preferred design of a PW Control Word to be used over an MPLS PSN, and the Pseudowire Associated Channel Header.

1.1. 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 RFC 2119 [RFC2119].

2. Avoiding ECMP

A PW that is carried over an MPLS PSN that uses the contents of the MPLS payload to select the ECMP path may be subjected to packet misordering [BCP]. In cases where the application using the PW is sensitive to packet misordering, or where packet misordering will disrupt the operation of the PW, it is necessary to prevent the PW being subjected to ECMP.

All IP packets [RFC791] [RFC2460] start with a version number that is checked by LSRs performing MPLS payload inspection. To prevent the incorrect processing of packets carried within a PW, PW packets carried over an MPLS PSN MUST NOT start with the value 4 (IPv4) or the value 6 (IPv6) in the first nibble [BCP], as those are assumed to carry normal IP payloads.

This document defines a PW header and two general formats of that header. These two formats are the PW MPLS Control Word (PVMCW), which is used for data passing across the PW, and a PW Associated Channel Header (PWACH), which can be used for functions such as OAM.

If the first nibble of a PW packet carried over an MPLS PSN has a value of 0, this indicates that the packet starts with a PVMCW. If the first nibble of a packet carried over an MPLS PSN has a value of 1, it starts with a PWACH. The use of any other first nibble value for a PW packet carried over an MPLS PSN is deprecated.

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 that prevents packet misordering. A suitable mechanism is the PVMCW described in Section 3 for data, and the PWACH described in Section 5 for channel-associated traffic.

The PVMCW or the PWACH MUST immediately follow the bottom of the MPLS label stack.

3. Generic PW MPLS Control Word

The Generic PW MPLS Control Word (PWMCW) is shown in Figure 1.

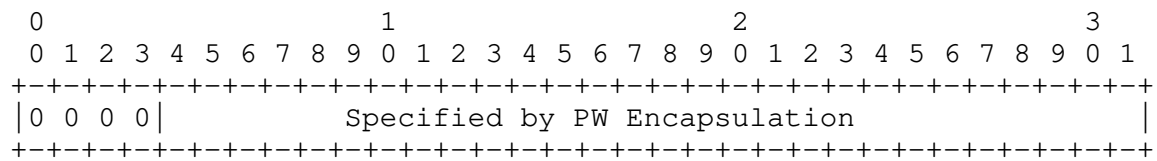


Figure 1: Generic PW MPLS Control Word

The PW set-up protocol or configuration mechanism determines whether a PW uses a PWMCW. Bits 0..3 differ from the first four bits of an IP packet [BCP] and hence provide the necessary MPLS payload discrimination.

When a PWMCW is used, it MUST adhere to the Generic format illustrated in Figure 1 above. To provide consistency between the designs of different types of PW, it SHOULD also use the following preferred format:

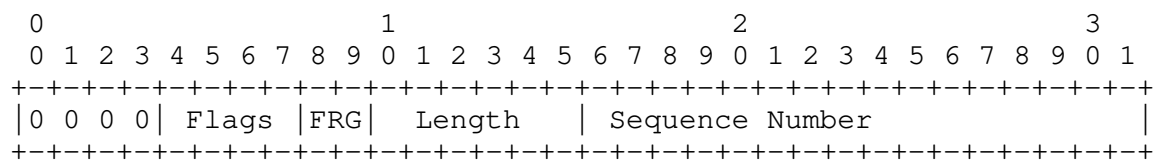


Figure 2: Preferred PW MPLS Control Word

The meaning of the fields of the Preferred PW MPLS Control Word (Figure 2) is as follows:

Flags (bits 4 to 7):

These bits MAY be used by for per-payload signaling. Their semantics MUST be defined in the PW specification.

FRG (bits 8 and 9):

These bits are used when fragmenting a PW payload. Their use is described in [FRAG], which is currently a work in progress. When the PW is of a type that will never need payload fragmentation, these bits may be used as general purpose flags.

Length (bits 10 to 15):

When the PSN path between the PEs includes an Ethernet segment, the PW packet arriving at the CE-bound PE from the PSN may include padding appended by the Ethernet Data Link Layer. The CE-bound PE uses the length field to determine the size of the padding added by the PSN, and hence extract the PW payload from the PW packet.

If the MPLS payload is less than 64 bytes, the length field MUST be set to the length of the PW payload plus the length of the PWMCW. Otherwise it MUST be set to zero.

Sequence number (Bit 16 to 31):

The sequence number implements the sequencing function [RFC3985]. The use of this field is described in Section 4.

4. Sequencing

The sequence number mechanism is PW specific. The PW encapsulation specification MAY define a sequence number mechanism to be used, or it may indicate that the mechanism described here is to be used. A pseudo-code description of this mechanism is given in the non-normative Appendix.

The sequence number mechanism described here uses a circular unsigned 16-bit number space that excludes the value zero.

4.1. Setting the Sequence Number

For a given PW, and a pair of routers PE1 and PE2, if PE1 supports packet sequencing and packet sequencing is enabled for the PW, then the following procedures MUST be used:

- o The initial packet transmitted on the PW MUST be sent with sequence number one.
- o Subsequent packets MUST increment the sequence number by one for each packet.
- o The sequence number that follows 65535 (maximum unsigned 16-bit number) is one.

If the transmitting router PE1 does not support sequence number processing, or packet sequencing is disabled, then the sequence number field in the control word MUST be set to zero for all packets transmitted on the PW.

4.2. Processing the Sequence Number

If a router PE2 supports receive sequence number processing, and packet sequencing is enabled for this PW, then the following procedure is used:

When a PW is initially set up, the "expected sequence number" associated with it **MUST** be initialized to one.

When a packet is received on that PW, the sequence number **SHOULD** be processed as follows:

- o If the sequence number on the packet is zero, the sequence integrity of the packets cannot be determined. In this case, the received packet is considered to be in order.
- o Otherwise if the packet sequence number equals the expected sequence number, the packet is in order.
- o Otherwise if the packet sequence number is greater than the expected sequence number, and the packet sequence number minus the expected sequence number is less than 32768, the packet is within the allowed receive sequence number window. The implementation **MAY** treat the packet as in order.
- o Otherwise if the packet sequence number is less than the expected sequence number and the expected sequence number minus the packet sequence number is greater than or equal to 32768, the packet is within the allowed receive sequence number window. The implementation **MAY** treat the packet as in order.
- o Otherwise the packet is out of order.

If the packet is found to be in order, it **MAY** be delivered immediately.

If the packet sequence number was not zero, then the expected sequence number is set to the packet sequence number plus one. The expected sequence number that follows 65535 (maximum unsigned 16-bit number) is one.

Packets that are received out of order **MAY** either be dropped or reordered. The choice between dropping or reordering an out-of-sequence packet is at the discretion of the receiver.

If a PE negotiated not to use receive sequence number processing, and it received a non-zero sequence number, then it **SHOULD** send a PW status message indicating a receive fault, and disable the PW.

7. Security Considerations

An application using a PW Associated Channel must be aware that the channel can potentially be misused. Any application using the Associated Channel MUST therefore fully consider the resultant security issues, and provide mechanisms to prevent an attacker from using this as a mechanism to disrupt the operation of the PW or the PE, and to stop this channel from being used as a conduit to deliver packets elsewhere. The selection of a suitable security mechanism for an application using a PW Associated Channel is outside the scope of this document.

If a PW has been configured to operate without a CW, the PW Associated Channel Type mechanism described in the document MUST NOT be used. This is to prevent user payloads being fabricated in such a way that they mimic the PW Associated Channel Header, and thereby provide a method of attacking the application that is using the Associated Channel.

8. Acknowledgements

The authors wish to thank David Allan, Thomas Nadeau, Yaakov Stein, and Mark Townsley for their input to this work.

9. Normative References

- [RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

10. Informative References

- [BCP] Swallow, G., Bryant, S., and L. Andersson, "Avoiding Equal Cost Multipath Treatment in MPLS Networks", Work in Progress, September 2005.
- [FRAG] Malis, A. and M. Townsley, "PWE3 Fragmentation and Reassembly", Work in Progress, November 2005.
- [IANA] Martini, L., "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)", Work in Progress, November 2005.
- [RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.
- [RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", RFC 2992, November 2000.
- [RFC3985] Bryant, S. and P. Pate, "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, March 2005.
- [VCCV] Nadeau, T. and R. Aggarwal, "Pseudowire Virtual Circuit Connectivity Verification (VCCV)", Work in Progress, August 2005.

Appendix. Sequence Number Processing

This appendix is non-normative.

This appendix provides a pseudo-code description of the sequence number processing mechanism described in Section 4.2.

```
unsigned16 RECEIVED      /* packet sequence number
unsigned16 EXPECTED = 1 /* expected sequence number
                        /* initialized to one
boolean sequencingDisabled
boolean dropOutOfOrder /* policy on in-window out of sequence
                        /* packets
```

```
updateExpected()
begin
    EXPECTED := RECEIVED + 1;
    /* Because EXPECTED is an unsigned16 it will wrap
    /* from 65535 to 0
    /* zero is skipped
    if (EXPECTED = 0)
        EXPECTED := 1;
    return;
end;
```

On receipt of a PW packet from PSN:

```
begin
    if (RECEIVED = 0) then begin
        processPacket();
        return;
    end;

    if (sequencingDisabled) then begin
        /* A packet was received with non-zero sequence number, but
        /* sequencing is disabled
        indicateReceiveFault();
        disablePW();
        return;
    end;

    /* The received sequence is the expected sequence number
    if ((RECEIVED = EXPECTED) then begin
        /* packet is in order
        processPacket();
        updateExpected();
        return;
    end;
```

```
/* Test for received sequence number is greater than
/* the expected sequence number and is within the
/* allowed receive sequence number window
if ((RECEIVED > EXPECTED) and
    ((RECEIVED - EXPECTED) < 32768) then begin
    /* packet is in the window, but there are late/missing
    /* packets
    if (dropOutOfOrder) then begin
        /* policy is to receive immediately, dropping
        /* out of sequence packets
        processPacket();
        updateExpected();
        return;
    end else begin
        /* policy is to wait for late packets
        processMissingPackets();
        return;
    end;
end;

/* Test for the received sequence is less than the
/* expected sequence number and is within the allowed
/* receive sequence number window
if ((RECEIVED < EXPECTED) and
    ((EXPECTED - RECEIVED) >= 32768) then begin
    /* packet is in the window, but there are late/missing
    /* packets

    if (dropOutOfOrder) then begin
        /* policy is to receive immediately, dropping
        /* out of sequence packets
        processPacket();
        updateExpected();
        return;
    end else begin
        /* policy is to wait for late packets
        processMissingPackets();
        return;
    end;
end;

/* Received packet was outside the allowed receive
/* sequence number window
processOutOfWindow();
end;
```

Authors' Addresses

Stewart Bryant
Cisco Systems,
250, Longwater,
Green Park,
Reading, RG2 6GB,
United Kingdom.

EMail: stbryant@cisco.com

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

EMail: swallow@cisco.com

Luca Martini
Cisco Systems, Inc.
9155 East Nichols Avenue, Suite 400
Englewood, CO, 80112

EMail: lmartini@cisco.com

Danny McPherson
Arbor Networks, Inc.

EMail: danny@arbor.net

Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

