PWE3	Y(J). Stein	
Internet-Draft	I. Mendelsohn	
Intended status: Standards Track	R. Insler	
Expires: May 7, 2009	RAD Data Communications	
	November 03, 2008	

TOC

PW Bonding
draft-stein-pwe3-pwbonding-01.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.
Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as “work in progress.”
The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.
The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.
This Internet-Draft will expire on May 7, 2009.

Abstract

There are times when pseudowires must be transported over physical links with limited bandwidth. We shall use the term "bonding" (also variously known as inverse multiplexing, link aggregation, trunking, teaming, etc.) to mean an efficient mechanism for separating the PW traffic over several links. Unlike load balancing and equal cost multipath, bonding makes no assumption that the PW traffic can be decomposed into distinguishable flows, and thus bonding requires delay compensation and packet reordering. Furthermore, PW bonding can optionally track bandwidth constraints in order to minimize packet loss.

Table of Contents

---

1. Introduction

TOC

Inverse multiplexing is any mechanism for transporting a single high capacity traffic flow over multiple lower capacity paths. Inverse multiplexing is also known as bonding, link load balancing, link aggregation, trunking, teaming, concatenation, and multipath. In the context of pseudowires we will use the term bonding.

Bonding has been defined for many transport technologies (and often more than one mechanism has been developed for a single technology) including TDM (continguous and virtual concatenation VCAT), ATM (ATM forum's IMA and ITU's G.998.1 multi-pair bonding), Ethernet (802.3 link aggregation LAG and EFM PME aggregation), xDSL (the previous two and G. 998.3 time domain inverse multiplexing TDIM), PPP (MLPPP), and in the context of IP transport, equal cost multiplath (ECMP).

Regardless of the transport infrastructure, all bonding mechanisms must confront a fundamental problem, namely that the constituent paths will in general have different (and not necessarily constant) propagation delays. Thus a mechanism must be employed to ensure in-order delivery of the data units. Two solutions have been proposed for this problem, namely performing differential delay compensation, and decomposing the input into mutually distinct flows. Methods using the former solution (e.g., VCAT, TDIM) buffer the data from each path at egress (e.g., VCAT buffers up to 1/2 second), and introduce protocol elements to synchronize the paths before recombining them. Methods using the latter soution (LAG, ECMP) skirt the problem by consistently mapping data units from a given flow onto the same constituent path, assuming that there is only the need to maintain order inside each flow, and not across flows.

Methods employing differential delay compensation tend to more complex and to require large buffers, but are universally applicable. Methods decomposing the input into flows depend on the existence of such flows and sniffing the input for their identification. Thus if the input is a single large flow, or if it is not possible to identify flows (e.g., due to lower layer encryption), or if it is undesirably complex to do so, these methods may not be applicable.

Furthermore, methods decomposing the input into flows tacitly assume that the hashing of flow identifiers onto tunnels results in fair distribution of traffic. This is generally a good assumption when there are a very large number of independent flows. Incorrect distribution causes some underlying paths to become congested and drop packets, while others are relatively underutlized. Direct inverse multiplexing with differential delay compensation one can ensure fairness, and in fact can adapt to underlying paths with unequal and even time varying capacity.

In the context of pseudowires a decomposition mechanism has been previously proposed [I-D.bryant-filsfils-fat-pw] (Bryant, S., Filsfils, C., Drafz, U., Kompella, V., Regan, J., and S. Amante, “Flow Aware Transport of MPLS Pseudowires,” March 2009.). The present draft proposes a PW bonding mechanism based on direct inverse multiplexing with differential delay compensation. In particular, the proposed mechanism may be used when PWs are supported by DSL links.

The simplest scenario for PW-bonding is depicted in Figure 1. Here the entire PW is transported edge to edge over separate PW components, each inside a distinct transport tunnel. A somewhat more complex scenario is partial path bonding, as depicted in Figure 2, where only a portion of the PW path is bandwidth restricted. Here only the PW components are shown, and not the tunnels into which they are placed. Here it is required to separate the PW into components in separate tunnels at some point inside the network. However, since P device where this happens is not PW aware, the PW components must still be defined by the ingress PE.

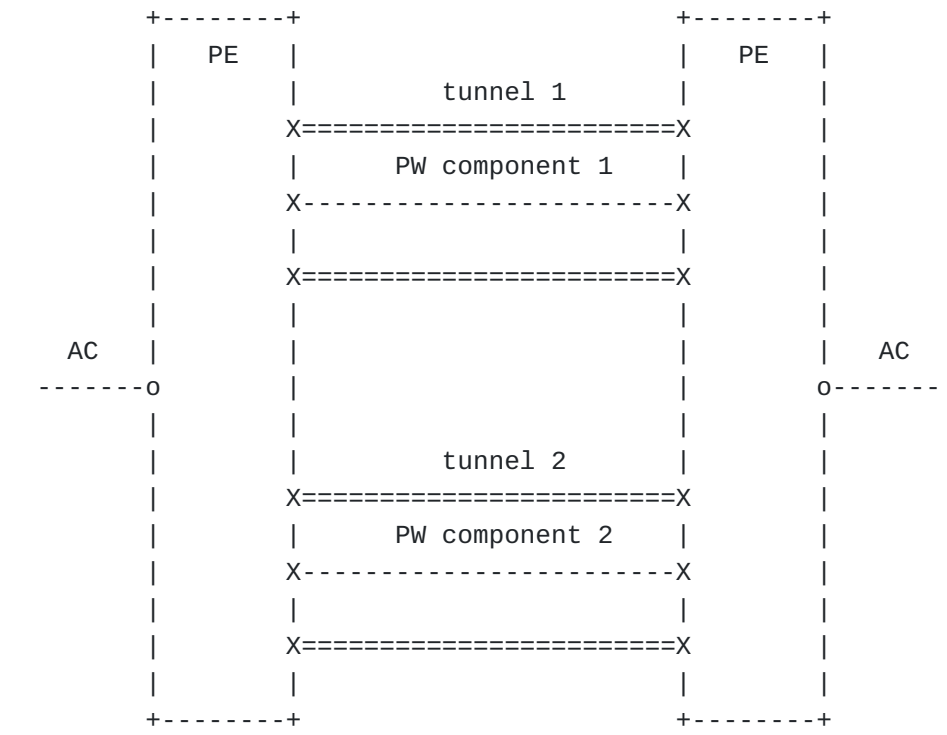

Figure 1. edge-to-edge PW bonding - 2 PW components in tunnels

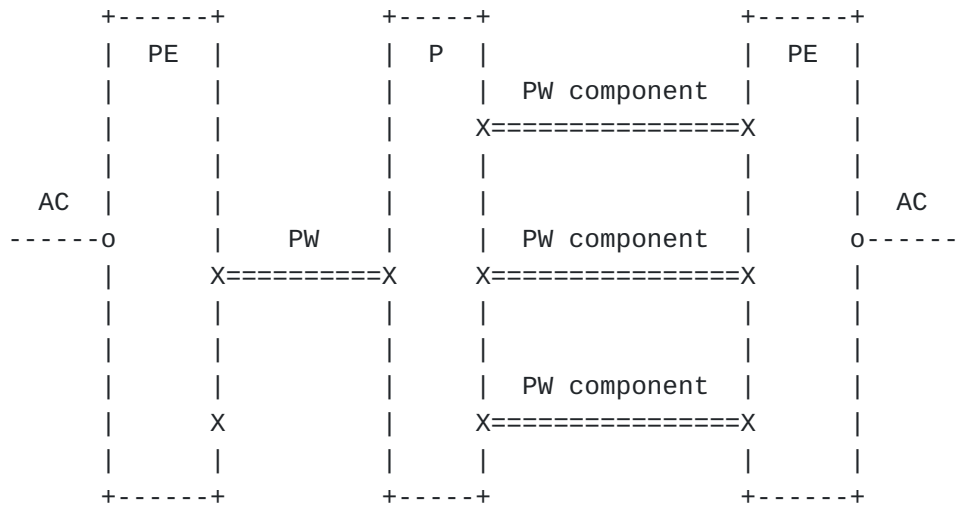

Figure 2. partial path PW bonding - 3 PW components

Each PW component will normally receive a distinct PW label, and thus seem to the network to be a distinct PW. Furthermore, PW components MUST use the PW control word [RFC4385] (Bryant, S., Swallow, G., Martini, L., and D. McPherson, “Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN,” February 2006.). However, as we shall see in the next section, the sequence number generation and processing is different for PW components that for true PWs.

2. PW Bonding mechanism

TOC

As discussed in the previous section, at the egress PE the traffic from each PW component is buffered, and the protocol is responsible for ensuring that packets constituting the PW are reassembled in correct order. This is accomplished by mandating use of the PW control word, and sharing the same sequence number sequence for all PW components making up the PW. The sequence numbers are used by the egress PE to ensure properly ordering. The idea is depicted in Figure 3, for the simple case of edge-to-edge bonding. Here eight packets are divided amongst three PW components by the ingress PE, according to a bandwidth allocation algorithm to be described later. Due to different link latencies, the packets arrive at the egress out of order, but are easily reordered by the egress PE by observing the sequence number.

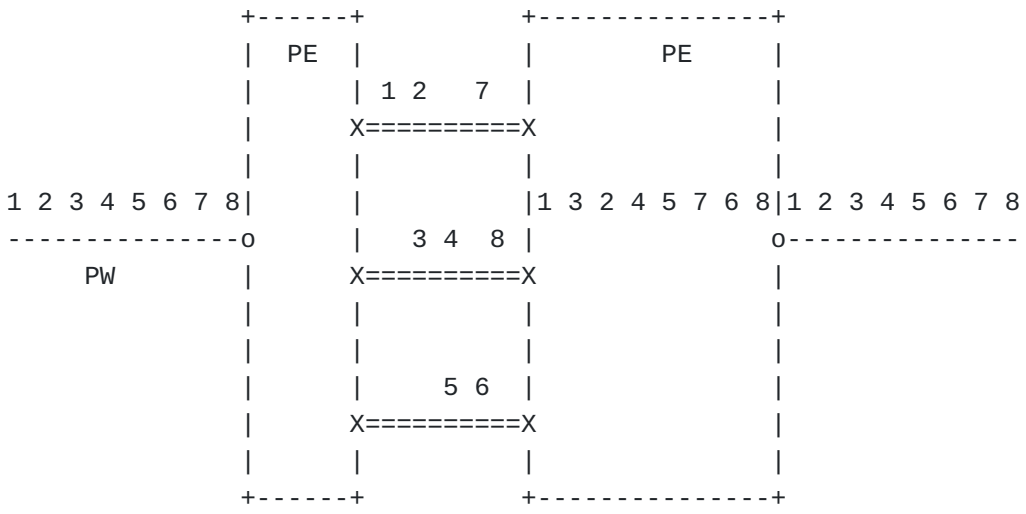

Figure 3. Use of sequence numbers to ensure correct packet ordering

In order to enable reordering, the egress PE must allocate sufficient buffer memory to sustain the largest expected differential delay. The differential delay is added to the latencies of all packets, making the effective latency equal to that of the slowest PW component.

3. PW Dynamic Bandwidth Allocation

TOC

In the simplest case, all packets to be sent over the various PW components are of the same size, and all PW components support the same data rates. For this case (but only for this case), a simple round-robin algorithm for distributing the packets onto PW components is optimal in the sense that it minimizes the probability of packet loss due to buffer exhaustion.

The simple round-robin algorithm is not optimal when the packets are not all of the same size, or when the PW components do not all support the same data rate, or both. In such cases we need to fairly distribute data bytes over the components in such fashion as to minimize the probability that a packet will be dropped due to over-run of a component's buffer. While the packet sizes are always known before transmission, the state of the buffers are usually unknown, and in some cases the supported data rates may be unknown. The following discussion will be for the edge-to-edge component case; the partial path case is similar, but requires separate consideration of the two directions.
If the packet size is not constant, and the component rates are known, but we have no further information (e.g., we do not know the size of the buffers, nor do we have feedback from the egress PE on the actual fill states) the best algorithm for an ingress PE is based on a leaky bucket scheme. In this scheme the ingress PE maintains, for each PW component, a variable Bn that approximately tracks the fill state of the egress PE's buffer for this component. The variable Bn is continually decreased at a rate equal to the data rate of the component n, but always remains non-negative. Each time a packet is sent over PW component n, its size in bytes is added to Bn. When a new packet needs to be sent, the ingress PE sends it on the PW component with minimal Bn. This algorithm can also be used when it can be assumed that the component rates are equal, or approximately so.
If in addition to packet size and PW component date rates, the ingress PE knows the buffer size used for differential compensation, a similar, but somewhat better, algorithm can be used. When deciding over which component to send the packet, rather than choosing the minimal Bn, the ingress PE chooses the maximal Bn to which the packet size can be added without overflowing the given buffer size. In practice some extra margin must be applied in order to account for PDV.
Finally, if the egress PE can send information on the actual state of its buffers back to the ingress PE, then an algorithm that uses these buffer states instead of the approximated leaky bucket ones can be employed.
Any implementation MUST support the round-robin method, and SHOULD support the first leaky bucket mode. Control protocol extensions are needed to enable communication from egress back to ingress of the additional information needed to support more optimal modes. If the rates can be accurately known the first leaky bucket mode MUST be used, and if further information is available then other mechanisms MAY be used.

---

4. Protocol Extensions

TOC

In order to set up the PW components using the PWE3 control protocol [RFC4447] (Martini, L., Rosen, E., El-Aawar, N., Smith, T., and G.

Heron, “Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP),” April 2006.) a single PWid or generalized PWid is assigned to the logical PW, and additional PWids or generalized PWids are allocated for the PW components. All PW components are assigned an identical group ID, in order to indicate their relationship, and to enable easy withdrawal of the logical PW. First the logical PW is set up using a label mapping message containing the interface parameters, and a new "bonding" sub-TLV containing the group ID. Subsequently the PW components are configured. Each PW component is assigned to a distinct transport tunnel by mechanisms not specified here.
Attachment circuit faults are signaled via PW status messages associated with the PWid or generalized PWid of the logical PW. PW component faults and capacity indicators are sent via status messages per PW component PWid or generalized PWid.
Enhancements to the PWE3 control protocol are needed in order to associate PW components with distinct labels in distinct tunnels to a single logical PW, and to communicate component capacity and status information. The format of these LDP extensions will be detailed in the next version of this draft.
Standard VCCV mechanisms [RFC5085] (Nadeau, T. and C. Pignataro, “Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires,” December 2007.) may be used independently for each PW component, and the resulting connectivity information may be used by the ingress PE in the process of distributing traffic over PW components. VCCV for the partial path scenario is for further study.

---

5. Partial Path PW Bonding

TOC

When only a portion of the PW's path suffers from bandwidth constriction, the partial path bonding scenario depicted in Figure 2 is used. As for the regular bonding case, the ingress PE decomposes the input into multiple PW components, and performs the same algorithm to decide into which component to send a given packet. For those portions of the network where a single tunnel can support the entire service bandwidth, the PW components may all be all placed in the same transport tunnel. For constricted bandwidth segments, each PW component must be placed in a distinct tunnel. The distinct transport tunnels are merged into the single tunnel using label merging, per section 3.26.2 of [RFC3031] (Rosen, E., Viswanathan, A., and R. Callon, “Multiprotocol Label Switching Architecture,” January 2001.).
Another case of practical interest is when the bandwidth is restricted in a non-MPLS access network, and the PE terminating the MPLS can not inverse multiplex the traffic onto low capacity links based on PW labels alone. This case arises for a DSLAM terminating MPLS (or a PE terminating MPLS upstream from the DSLAM) and forwarding to customers solely based on Ethernet MAC address (and possibly VLAN ID). For such a

case a double PW encapsulation may be used. Through the core network we tunnel an Ethernet PW, which itself carries the bonded PW components (which may be of any type supported by PWE encapsulations), see Figure 4.

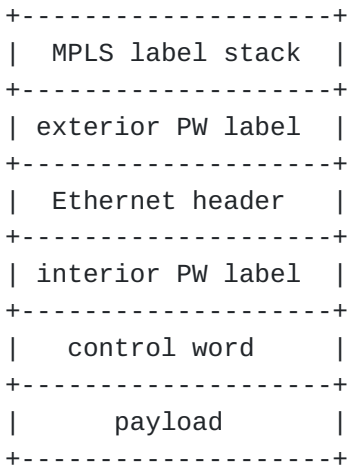

Figure 4. packet format for DSL partial path scenario

The DSLAM (or PE immediately upstream from the DSLAM) terminates the MPLS and exterior PW protocols, thus exposing the Ethernet header. Under the Ethernet header there MAY be an MPLS header (which the CE negotiates with the immediately upstream PE), and there MUST be an interior PW label (which the CE negotiates with the remote CE or PE). Based purely on the Ethernet addressing the DSLAM distributes the traffic over multiple DSL links following the partition crafted by the ingress PE. All of these DSL links terminate on a single CE device which terminates the Ethernet, exposes the interior PW labels and sequence numbers in the control word. Using these sequence numbers the CE can thus piece together the original traffic stream.

---

6. Applicability

TOC

PW bonding is a useful mechanism when the bandwidth of available physical links is insufficient to carry the user traffic, but several links can be dedicated. Unlike load balancing and equal cost multipath mechanisms, PW bonding makes no assumption that the PW traffic can be decomposed into distinguishable flows. It is fully applicable for non-IP or encrypted traffic. By using mechanisms described above, PW bonding can approach full utilization of the aggregate link bandwidth. PW bonding involves delay compensation and packet reordering, and thus requires allocation of sufficient memory at the egress PE. The amount of memory needed is proportional to the link speed and to the difference in propagation delay between the fastest and slowest links.

Thus PW bonding is most applicable when the link speeds are low (e.g., supported by DSL lines), and the delay differences are small.
Only the PEs need to know that the PW components are not full PWs (the only difference being the sequence number processing). Thus PW bonding requires changes only to the PEs and does not require any changes to the intervening PSN.

---

7. Security Considerations

TOC

PW bonding does not introduce security considerations above those present for regular PWs. In particular, attacks based on sequence number manipulation are of concern. For partial path cases where CE devices participate in the PWE signaling, authentication is required.

---

8. IANA Considerations

TOC

Required extensions to the PWE3 control protocol, including the sub-TLV type code for the PW component label, and new PW status codes, will be detailed in the next version of this draft.

---

9. Acknowledgments

TOC

The authors would like to thank Gabriel Zigelboim for fruitful discussions on optimal dynamic allocation mechanisms.

---

10. References

TOC

10.1. Normative References

TOC

[RFC3031]	Rosen, E., Viswanathan, A., and R. Callon, “Multiprotocol Label Switching Architecture,” RFC 3031, January 2001 (TXT).
[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 (TXT).
[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 (TXT).
[RFC5085]	Nadeau, T. and C. Pignataro, “Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires,” RFC 5085, December 2007 (TXT).

10.2. Informative References

TOC

[I-D.bryant-filsfils-fat-pw]	Bryant, S., Filsfils, C., Drafz, U., Kompella, V., Regan, J., and S. Amante, “Flow Aware Transport of MPLS Pseudowires,” draft-bryant-filsfils-fat-pw-03 (work in progress), March 2009 (TXT).

Authors' Addresses

TOC

	Yaakov (Jonathan) Stein
	RAD Data Communications
	24 Raoul Wallenberg St., Bldg C
	Tel Aviv 69719
	ISRAEL
Phone:	+972 3 645-5389
Email:	yaakov_s@rad.com
	Itai Mendelsohn
	RAD Data Communications
	24 Raoul Wallenberg St., Bldg C
	Tel Aviv 69719
	ISRAEL
Phone:	+972 3 645-5761
Email:	itai_m@rad.com
	Ron Insler
	RAD Data Communications
	24 Raoul Wallenberg St., Bldg C
	Tel Aviv 69719
	ISRAEL
Phone:	+972 3 645-5445
Email:	ron_i@rad.com

Full Copyright Statement

TOC

Copyright © The IETF Trust (2008).
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, THE IETF TRUST 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.