

Network Working Group  
Internet-Draft  
Expires: April 19, 2006

S. Bryant  
Cisco Systems  
R. Perlman  
Sun Microsystems  
A. Atlas  
Google  
D. Fedyk  
Nortel Networks  
October 16, 2005

TRILL using Pseudo-Wire Emulation (PWE) Encapsulation  
draft-bryant-perlman-trill-pwe-encap-00

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/lid-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 April 19, 2006.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

A new layer of encapsulation is required with R Bridges. This layer must contain at least a time-to-live and an R Bridge identifier field. This document proposes that the reuse of the encapsulation defined by

Internet-Draft [draft-bryant-perlman-trill-pwe-encap-00](#)

October 2005

PWE3 for encapsulation of Ethernet frames over an MPLS packet switched network.

## Table of Contents

<a href="#">1.</a>	Motivation . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Forwarding Considerations . . . . .	<a href="#">4</a>
<a href="#">2.1.</a>	Forwarding Table Population . . . . .	<a href="#">5</a>
<a href="#">2.2.</a>	QoS Treatment . . . . .	<a href="#">5</a>
<a href="#">2.3.</a>	Load Balancing . . . . .	<a href="#">6</a>
<a href="#">2.4.</a>	Multicast and Broadcast Frames . . . . .	<a href="#">6</a>
<a href="#">3.</a>	Dynamic Assignment of 19-bit Nicknames . . . . .	<a href="#">7</a>
<a href="#">4.</a>	Security Considerations . . . . .	<a href="#">8</a>
<a href="#">5.</a>	References . . . . .	<a href="#">8</a>
	Authors' Addresses . . . . .	<a href="#">9</a>
	Intellectual Property and Copyright Statements . . . . .	<a href="#">10</a>

---

Internet-Draft    [draft-bryant-perlman-trill-pwe-encap-00](#)    October 2005

## 1. Motivation

The TRILL encapsulation requires a TTL and an RBridge ID, which could be the ingress or the egress depending upon the particular packet. There are four encapsulation mechanism that TRILL could use:

- a. It could design its own encapsulation from scratch.
- b. It could use an Ethernet based encapsulation.
- c. It could use an IP based encapsulation.
- d. It could use an MPLS based encapsulation.

Adding, or removing an encapsulation, or forwarding a packet based on an encapsulation is one of the most time critical operation in any networking equipment, and usually requires hardware support. The use of a new network encapsulation type is always problematic because new hardware is usually required. This is expensive to design and deploy, and frequently has a significant time and risk impact on the market acceptance of a new network architecture. The use of a new, TRILL specific, encapsulation should therefore, if possible, to be avoided.

TRILL could opt to use an Ethernet based encapsulation. The nesting of 802.x tags is a well understood technology and suitable hardware is widely deployed. However the absence of a TTL field in the header means that a controlled convergence technology needs to be used [[CCONV](#)] to avoid the collateral damage caused by microlooping packets during network convergence. Although convergence control technologies are now available, they are not well understood by the networking industry, and their use by TRILL may not be accepted by the industry.

TRILL could use an IP encapsulation, but using an IP header for this purpose has issues (see Section 5.5 in [[RBRIDGE](#)]). Such issues

include the encapsulation overhead, the complexity of providing L2 services within the L3 subnet, and the additional potential work for fragmentation and reassembly.

The simplest existing encapsulation that meets the TRILL requirement is that defined by PWE3 for the encapsulation of Ethernet frames over an MPLS packet switched network [[PWE3-ETHER](#)]. The forwarding functionality required by TRILL is very similar to that needed to implement virtual private lan service (VPLS [[VPLS](#)]). Equipment capable of encapsulating Ethernet packets for carriage over an MPLS core is widely available, and the modifications necessary to support TRILL would reside primarily in the control plane.

The encapsulation described in [[PWE3-ETHER](#)] consists of an MPLS label stack [[RFC3032](#)] plus an OPTIONAL four byte control word. At least one MPLS label stack entry (LSE) will be present in the TRILL packet. In addition to containing the label (delivery address), the LSE also contains the TTL field required by TRILL, and a QoS field (exp bits) that may also be of use.

The control word carries some information that prevents the packet being mistaken for an IP packet in an MPLS network and incorrectly being subjected to ECMP. This functionality is not required in a TRILL network. The control word also contains a sequence number which is used to prevent the out of order delivery of PWE3 Ethernet payloads. If order preservation is required the control word **MUST** be used, otherwise a TRILL implementation **MAY** omit the PWE3 control word.

The use of the PWE3 Ethernet over MPLS encapsulation by TRILL would facilitate the integration of TRILL and MPLS networking.

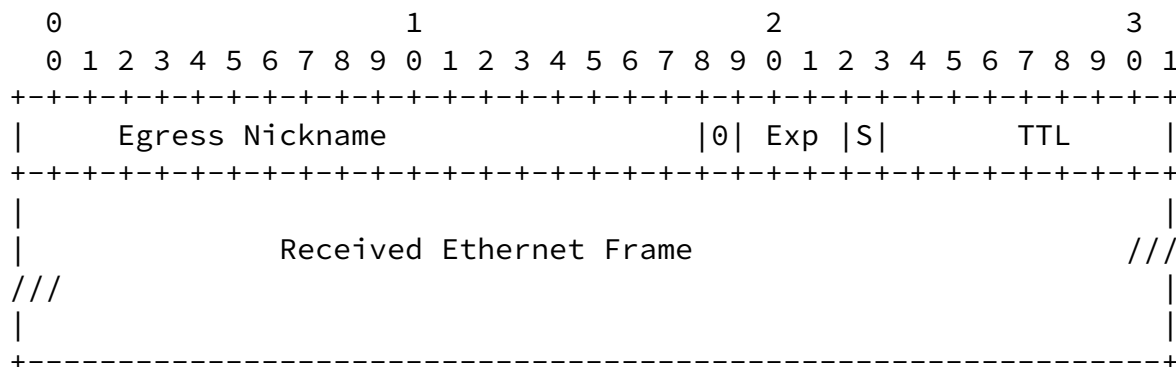
## [2.](#) Forwarding Considerations

As described in [Section 3](#), each RBridge can obtain two 19-bit nicknames. The first nickname can be used for the RBridge when unicast traffic is directed to it; it is the egress RBridge nickname. The second nickname can be used for multicast and broadcast traffic from the RBridge; it will be the ingress RBridge nickname.

An MPLS shim header contains a 20-bit label field. The same format

can be used for the TRILL shim header; the labels will be distributed via the link-state protocol used between RBridges; those labels will be unique within this RBridge network instance. The Ethertype will indicate that it is a TRILL frame; this will be used to provide the correct forwarding context for the label space. The bottom-most bit of the label field can indicate whether the top 19 bits indicate a unicast nickname or a multicast and broadcast nickname. The forwarding behavior will differ based upon this.

In the unicast case, when an Ethernet frame is received without the new TRILL ethertype, the ingress RBridge will lookup the egress RBridge, as specified in [RBRIDGE], and obtain its egress RBridge nickname. The ingress RBridge will also determine if the Ethernet frame has a priority specified as in 802.1p and will extract that 3-bit priority field. Then the original Ethernet frame will be encapsulated as follows:



Exp: Indicates Priority  
S: Bottom of Stack, 1 bit  
TTL: Time to Live, 8 bits

Figure 1: Unicast Encapsulation

Traditional bridges avoid misordering; it is an Ethernet invariant. During a traditional network convergence using a link-state protocol, it is possible for packets to be misordered. The PWE3 control word can be used for this purpose with pseudo-wires ([Section 3.7](#) in [PWE3-

ETHER]); such use might require too much hardware state due to the desired load-balancing of flows.

This gives the encapsulated frame the same format as an Ethernet pseudo-wire [[PWE3-ETHER](#)]. The forwarding path can be exactly the same as that used for an Ethernet pseudo-wire.

### [2.1.](#) Forwarding Table Population

When an RBridge X learns a new egress nickname A, on each interface, the top 19 bits of the label are filled out with the new nickname and the bottom bit (the unicast/other) is set to 0; an insegment for that label is created (usually by adding an entry into the input label mapping (ILM) table.) A corresponding outsegment is installed for each interface that is on the shortest path tree from the RBridge X to the RBridge indicated by A. That out-segment does a label swap operation, where the label swapped to is the same constructed label. The created in-segment is connected to the created out-segments with load balancing specified; only one out-segment will be used for a particular frame.

### [2.2.](#) QoS Treatment

The encapsulation preserves the priority, if specified, of the frame without requiring intermediate RBridges to examine the encapsulated frame. The ingress RBridge extracts the priority from the 802.1p

field and stores that in the EXP field of the shim header.

When an RBridge adds the outer Ethernet frame to an TRILL encapsulated frame, the RBridge can specify an 802.1p field with a priority equal to that stored in the EXP field of the shim header. If the EXP field is 0, then no 802.1p field is necessary.

### [2.3.](#) Load Balancing

Load balancing between multiple equal cost paths is a concern for RBridges. To properly load balance TRILL encapsulated frames, an RBridge should identify TRILL encapsulated frames and implement a specific hashing algorithm for this ethertype. A specific Ethertype would be used for TRILL frames, making them trivial to identify.

The load balancing that would be provided by current mechanisms is not sufficient. Without the PWE3 control word, either the TRILL encapsulated frame would appear as non-IP and would be load balanced based on a hash of the label stack (known as LABEL ECMP [[MPLS-ECMP](#)]) or it would be mis-identified as IP and load balanced based on the bits located where IP addresses would be if the encapsulated Ethernet frame were an IP packet. The former case would provide no flow diversity, since all TRILL encapsulated frames would have the same label, corresponding to the same egress RBridge nickname. The latter case could risk packet re-ordering. Current mechanisms seeing the PWE3 control-word would use LABEL EMP and thus provide no flow diversity.

#### [2.4.](#) Multicast and Broadcast Frames

For multicast/broadcast frames, the ingress RBridge nickname indicates the spanning tree which should be used. As with the unicast case, a label is formed of the nickname field and the unicast/other field (label[19:1] = nickname[18:0] and label[0] = 1). The treatment of the TTL field and the EXP fields are the same.

When an RBridge learns of a new ingress RBridge nickname, an ILM entry corresponding to the label is created. An out-segment is created for each interface that is in the SPT rooted at the ingress RBridge. The in-segment is connected to the created out-segments with multicasting specified; subject to filtering, each frame will be sent out each out-segment. Except for the egress filtering, the above forwarding behavior is already part of MPLS; it is used to support point-to-multipoint MPLS LSPs.

Filtering may be applied based upon the frame and the outgoing interface's membership. For instance, if a frame is being broadcast along a VLAN and an interface is marked as not being connected to any

bridges or RBridges with VLAN membership, then the frame need not be sent out that interface. Similarly, if a frame is being multicasted, the RBridge could decide to filter the frame if the interface is explicitly known to not be part of the multicast tree.

### [3.](#) Dynamic Assignment of 19-bit Nicknames

We assume each RBridge has a unique 6-byte system ID, which it uses as its IS-IS ID. In order to use the compressed MPLS-like encoding of the shim header, we need to create an identifier which is 19-bits. This gives a space of half a million nicknames, large enough that there will be enough nicknames. We do, however, need a method for assigning nicknames to RBridges so that the nicknames are unique within the RBridge domain.

We will assign a new type value to be carried in LSPs. The TLV will carry the nickname the LSP source wishes to use. The TLV will be:

```

+-----+-----+-----+
| type | length | value=19 bit nickname |
+-----+-----+-----+

```

Figure 2: Nickname TLV

Each RBridge chooses its own nickname. However, each RBridge is also responsible for ensuring that its nickname is unique. If R1 chooses nickname x, and R1 discovers, through receipt of R2's LSP, that R2 has also chosen x, then the RBridge with the lower system ID keeps the nickname, and the other one must choose a new nickname.

If two RBridge domains merge, then there might be a lot of nickname collisions for a short time, but as soon as each side receives the link state packets of the other, the RBridges that need to change nicknames will quickly become aware of this, and choose new nicknames that do not, to the best of their ability, collide with any existing nicknames.

To minimize the probability of nickname collisions, each RBridge chooses its nickname randomly from the set of assigned nicknames. Alternatively, we could use some sort of hash algorithm (such as the bottom 19 bits of the MD5 of the RBridge's system ID), to choose the first nickname, and then if there is a collision, go to the next 19 bits of the MD5, and so on, until all 128 bits of the MD5 hash are exhausted, in which case the RBridge hashes its own system ID again, this time together with the constant "1".

There is no reason for all RBridges to use the same algorithm for

choosing nicknames. Picking them at random, or using a hash, are an



attempt to avoid collisions when the network starts up, but that is only an optimization. Even if all RBridges used the same algorithm, say as a worst case, they all start with "1" and count up sequentially until they find an uncontested nickname, the network will eventually stabilize. And once it is stable, nicknames should remain stable even as routers go up or down.

To minimize the probability of a new RBridge usurping a nickname already in use, an RBridge should wait to acquire the link state database from a neighbor before it announces its own nickname.

#### [4.](#) Security Considerations

The security implications of selecting this format have not yet been considered.

#### [5.](#) References

- [CCONV] Bryant, S. and M. Shand, "Applicability of Loop-free Convergence", [draft-bryant-shand-lf-conv-frmwk-00.txt](#) (work in progress), June 2005.
- [MPLS-ECMP] Swallow, G., Bryant, S., and L. Andersson, "Avoiding Equal Cost Multipath Treatment in MPLS Networks", [draft-ietf-mpls-ecmp-bcp-01.txt](#) (work in progress), July 2005.
- [PWE3-ETHER] Martini, L., Rosen, E., and G. Heron, "Encapsulation Methods for Transport of Ethernet Over MPLS Networks", [draft-ietf-pwe3-ethernet-encap-10.txt](#) (work in progress), June 2005.
- [RBRIDGE] Perlman, R., Touch, J., and A. Yegin, "RBridges: Transparent Routing", [draft-perlman-rbridge-03.txt](#) (work in progress), May 2005.
- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", [RFC 3032](#), January 2001.
- [VPLS] Lasserre, M. and V. Kompella, "Virtual Private LAN Services over MPLS", [draft-ietf-l2vpn-ldp-07.txt](#) (work in progress), July 2005.

Authors' Addresses

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

Email: [stbryant@cisco.com](mailto:stbryant@cisco.com)

Radia Perlman  
Sun Microsystems

Email: [Radia.Perlman@sun.com](mailto:Radia.Perlman@sun.com)

Alia K. Atlas  
Google  
1600 Amphitheatre Parkway  
Mountain View, CA 94043  
USA

Email: [akatlas@alum.mit.edu](mailto:akatlas@alum.mit.edu)

Don Fedyk  
Nortel Networks  
600 Technology Park  
Billerica, MA 01821  
USA

Phone: +1 978 288 3041

Email: [dwfedyk@nortelnetworks.com](mailto:dwfedyk@nortelnetworks.com)

---

Internet-Draft    [draft-bryant-perlman-trill-pwe-encap-00](#)    October 2005

## Intellectual Property Statement

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](mailto:ietf-ipr@ietf.org).

## Disclaimer of Validity

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.

## Copyright Statement

Copyright (C) The Internet Society (2005). 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.

#### Acknowledgment

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