Internet Draft
Document: <draft-bryant-pwe3-protocol-layer-01.txt>
Expires: August 2002

Stewart Bryant
Lloyd Wood
Mark Townsley
cisco Systems Ltd

Danny McPherson
TCB

February 2002

Protocol Layering in PWE3

Status of this Memo

Abstract

This draft proposes a unified protocol layering approach for pseudo-wire emulation edge-to-edge (PWE3). It adopts the principle that PWE3 should be a single transport type operating over a common packet-switched network (PSN) service model using, wherever possible, existing IETF protocols. The draft defines the protocol layering model for pseudo-wires (PW), guidelines for the design of a specific encapsulation type, and the service requirements of the underlying PSN tunneling mechanism.

Table of Contents

1. Introduction

This document presents a unified protocol layering approach for pseudo-wire emulation edge-to-edge (PWE3). It adopts the principle that PWE3 should be a single transport type operating over a common packet-switched network (PSN) service model using, wherever possible, existing IETF protocols. This document defines the protocol layering model for pseudo-wires (PW), guidelines for the design of a specific encapsulation type, and the service requirements of the underlying PSN tunneling mechanism.

2. Terminology

This document uses the following definition of terms. A number of these terms are further clarified by reference to Figure 1.

CE-bound — The traffic direction where PW-PDUs are received on a PW via the PSN, decapsulated to retrieve the emulated service, and then sent to the destination CE.

CE Signaling — Messages sent and received by the CEs control plane. It may be desirable or even necessary for the PE to participate in or monitor this signaling in order to effectively emulate the service.

Customer Edge (CE) — A device where one end of a service originates and terminates. The CE is not aware that it is using an emulated service rather than a native service.

Inter-working	Interactions between networks, between end systems, or between parts thereof, with the aim of providing a functional entity capable of supporting an end-to-end communication.
Inter-working Function (IWF)	A function that facilitates inter-working between two dissimilar networks. NSP may perform the IWF function.
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.

Packet Switched Network (PSN)	A network using IP or MPLS as the mechanism of packet forwarding.
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.
PE-bound	The traffic direction where 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 effectively manage the PW.
Pseudo Wire (PW)	A connection between two PEs carried over a PSN.
PW End Service (PWES)	The interface between a PE and a CE. This can be a physical interface like a T1 or Ethernet, or a virtual interface like a VC or VLAN.
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   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   Used to set up, maintain and tear down the
   Signaling   underlying PSN tunnel.

   SAR   Segmentation and reassembly.

   Tunnel   A method of transparently carrying information
                        over a network.

3. Architecture of Pseudo-wires

This section describes the PWE3 architectural model.

3.1 Network Reference Model

Figure 1 illustrates the network reference model for PWs.

```
            |<-------------- Emulated Service ---------------->|
            |                                                  |
            |          |<------- Pseudo Wire ------>|          |
            |          |                            |          |
            |          |    |<-- PSN Tunnel -->|    |          |
            | PW End   V    V                  V    V  PW End  |
            V Service  +----+                  +----+  Service V
      +-----+    |     | PE1|==================| PE2|     |    +-----+
      |     |----------|............PW1.............|----------|     |
      | CE1 |    |     |    |                  |    |     |    | CE2 |
      |     |----------|............PW2.............|----------|     |
      +-----+  ^ |     |    |==================|    |     | ^  +-----+
```

```
          ^   |       +-----+                +-----+      | |  ^
          |   |   Provider Edge 1        Provider Edge 2  |  |
          |   |                                           |  |
       Customer |                                         | Customer
       Edge 1   |                                         | Edge 2
                |                                         |
                |                                         |
          native service                            native service
```

Figure 1: PWE3 Network Reference Model

The two PEs (PE1 and PE2) need to provide one or more PWs on behalf of their client CEs (CE1 and CE2) to enable them to communicate over the PSN. A PSN tunnel is established to provide a data path for the PW that is transparent to the network core. Native data units (bits, cells or packets) presented at the PW End Service (PWES) are encapsulated in a PW-PDU and carried across the underlying network via the PSN tunnel. The PEs perform the necessary encapsulation, decapsulation, sequencing, timing and any other functions required by the PW service.

3.2 Native Service Processing

In some applications, there is a need to perform operations on the native data units received from the CE (both payload and control information) before it is 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 sent to the PE over one or more physical interfaces. In most cases, there are cost and operational benefits in undertaking this native service processing (NSP) within the PE. This processed data is then presented to the PW via a virtual interface within the PE. It must be emphasized that this processing uses operations that are outside the scope of the PW defined here.

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 2 illustrates the relationship between NSP and the network reference model for PWs. The PW may provide connectivity to a virtual interface with the PE equipment. The NSP function may apply any transformation operation (modification, injection, etc) on data as it passes between the physical interface to the CE and the virtual interface to the PW. It may also combine or split data between the physical interfaces to the CE and the virtual interface to one or more PWs.


```
                         PW
                      End Service
                          |
                          |<------- Pseudo Wire ------>|
                          |                            |
                          |    |<-- PSN Tunnel -->|    |
                          V    V                  V    V     PW
```

```
                  +-----+----+                  +----+ End Service
       +-----+    |NSP1 | PE1|==================| PE2|     |    +-----+
       |     |    |     |............PW1.............|----------|     |
       | CE1 |----|     |    |                  |    |     |    | CE2 |
       |     | ^  |     |............PW2.............|----------|     |
       +-----+ |  |     |    |==================|    |     | ^  +-----+
               |  +-----+----+                  +----+     | |
               |        ^                                  | |
               |        |                                  | |
               |        |<------- Emulated Service ------->| |
               |        |                                    |
               | Virtual physical                            |
               |  termination                                |
               |        ^                                    |
          CE1 native    |                                CE2 native
           service      |                                service
                        |
                   CE2 native
                    service
```

Figure 2: NSP within the PWE3 Network Reference Model

Figure 2 shows the inter-working of one PE with NSP, and a second without this functionality. This is a useful reference point because it emphasises that the functional interface between NSP 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 NSP is also supported.

The operation of a system in which the NSP functionality includes terminating the data-link, and applying network layer processing to the payload, is also supported.

Internally, a PE with NSP has the following functional structure:

```
                +-----------------------------------------+
                |                PE Device                |
        Single  +-----------------------------------------+
        PWES    |                  |    |  Single         | PW Instance
     <--------->o     NSP          O<-->+  PW IWF         X<===========>
                |                  |    |  Instance       |
                |                  |    |                 |
                +-----------------------------------------+
```

Figure 3a: A simple point-to-point service

```
                +-----------------------------------------+
                |                PE Device                |
                +-----------------------------------------+
                |                  |    |  Single         | PW Instance
                |                  O<-->+  PW IWF         X<===========>
                |                  |    |  Instance       |
                |                  |    |-----------------|
       Single   |                  O<-->+  Single         | PW Instance
       PWES     |                  |    |  PW IWF         X<===========>
     <--------->o       NSP        |    |  Instance       |
                |                  |    |-----------------|
                |                  |    |    ...          |
                |                  |    |-----------------|
                |                  O<-->+  Single         | PW Instance
                |                  |    |  PW IWF         X<===========>
                |                  |    |  Instance       |
                +-----------------------------------------+
```

Figure 3b: A point-to-multipoint service

```
               +----------------------------------------+
               |                PE Device               |
               |----------------------------------------|
      Multiple |                 |    |   Single        | PW Instance
      PWES     |                 O<..>+   PW IWF        X<===========>
    <--------->o                 |    |   Instance      |
               |                 |    |-----------------|
    <--------->o                 O<..>+   Single        | PW Instance
               |                 |    |   PW IWF        X<===========>
    <--------->o       NSP       |    |   Instance      |
               |                 |    |-----------------|
    <--------->o                 |    |    ...          |
      ...      |                 |    |-----------------|
               |                 O<..>+   Single        | PW Instance
    <--------->o                 |    |   PW IWF        X<===========>
               |                 |    |   Instance      |
               +----------------------------------------+
```

Figure 3c: A full switch/bridge/cross-connect
multipoint-multipoint service

Notation:

o   A physical CE-bound PE port

O   An NSP virtual interface to a PW IWF instance.

\+   A PW IWF instance interface to the NSP.

X   A PE PSN-bound port.

Figure 3: PE internals showing NSP

3.3 Maintenance Reference Model

Figure 4 illustrates the maintenance reference model for PWs.

```
             |<------- CE (end-to-end) Signaling ------>|
             |     |<---- PW/PE Maintenance ----->|     |
             |     |     |<-- PSN Tunnel -->|     |     |
             |     |     |    Signaling     |     |     |
             |     V     V  (out of scope)  V     V     |
             v     +-----+                  +-----+     v
       +-----+     | PE1 |=================| PE2 |     +-----+
       |     |-----|.............PW1..............|-----|     |
       | CE1 |     |     |                  |     |     | CE2 |
       |     |-----|.............PW2..............|-----|     |
       +-----+     |     |=================|     |     +-----+
                   +-----+                  +-----+
       Customer   Provider                 Provider     Customer
        Edge 1     Edge 1                   Edge 2       Edge 2
```

Figure 4: PWE3 Maintenance Reference Model

The following signaling mechanisms are required:

- o The CE (end-to-end) signaling is between the CEs. This signaling can include frame relay PVC status signaling, ATM SVC signaling, etc.

- o 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.

- o The PSN Tunnel signaling controls the underlying PSN. Examples are L2TP control protocol, or MPLS LDP. This type of signaling is not within the scope of PWE3.

3.4 Protocol Stack Reference Model

Figure 5 illustrates the protocol stack reference model for PWs.

```
    +-----------------+                                +-----------------+
    |Emulated Service|                                |Emulated Service|
    |(e.g. TDM, ATM) |<===== Emulated Service ====>|(e.g. TDM, ATM) |
    +-----------------+                                +-----------------+
    |     Payload     |                                |     Payload     |
    |  Encapsulation  |<======== Pseudo Wire =======>|  Encapsulation  |
    +-----------------+                                +-----------------+
    |   PSN Tunnel,   |<========= PSN Tunnel =======>|   PSN Tunnel,   |
    | PSN & Physical  |                                | PSN & Physical  |
    |     Layers      |                                |     Layers      |
    +--------+--------+         ______________         +--------+--------+
             |                 /              \                 |
             +===============/      PSN       \===============+
                             \                /
                              \______________/
```

Figure 5: PWE3 Protocol Stack Reference Model

The PW provides the CE with what appears to be a direct physical connection to its peer at the far end. Native data units 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.

3.5 NSP Extension to Protocol Stack Reference Model

Figure 6 illustrates how the protocol stack reference model extended

to include the provision of native service processing. This shows the correct placement of the physical interface relative to the CE.


```
      +------------------------------------+
      |    Native Service Processing       |
      +---------------+---+----------------+
      |               |   | Emulated       |
      | Service       |   | Service        |
      | Interface     |   | (TDM, ATM,     |
      | (TDM, ATM,    |   | Ethernet,      |<=== Emulated Service ==
      | Ethernet,     |   | frame relay,   |
      | frame relay,  |   | etc.)          |
      | etc.)         |   +----------------+
      |               |   |    Payload     |
      |               |   | Encapsulation  |<==== Pseudo Wire ======
      |               |   +----------------+
      |               |   |  PSN Tunnel,   |
      |               |   | PSN & Physical |<==== PSN Tunnel =======
      |               |   |    Headers     |
      +---------------+   +----------------+
      |   Physical    |   |   Physical     |
      +-------+-------+   +--------+-------+
              |                    |
              |                    |             IP or MPLS Network
              |                    |            -----     ---
```

```
              |                  |            /     \---/   \
              |                  |           /               \
              |                  |          /                 \
              v                  +=========/       PSN         |
            To CE                          \                  /
                                            \      ---       /
                                             -----/   \-----/
```

Figure 6: Protocol Stack Reference Model with NSP

4. Protocol Layering Model

The PWE3 protocol-layering model is intended to minimise the differences between PWs operating over different PSN types. The design of the protocol-layering model thus has the goals of making each PW definition independent of the underlying PSN, and maximizing the reuse of IETF protocol definitions.

4.1 Protocol Layers

The logical protocol-layering model required to support a PW is expanded to provide more detail as shown in Figure 7.

```
          +---------------------------+
          |         Payload           |
          +---------------------------+
          |      Encapsulation        | <==== May be empty
          +---------------------------+
          |        PSN Tunnel         |
          +---------------------------+
          |     PSN Convergence       | <==== May be empty
          +---------------------------+
          |           PSN             |
```

```
+---------------------------+
|       MAC/Data-link       |
+---------------------------+
|         Physical          |
+---------------------------+
```

Figure 7: Logical Protocol Layering Model

The payload is transported over the Encapsulation Layer. The Encapsulation Layer carries any information, not in the payload itself, that is required by the PW CE-bound PE interface to send the payload to the CE via the physical interface.

If needed, this layer also provides support for real-time processing, and sequencing.

The PSN Tunnel Layer provides the ability to deliver multiple PWs over a single PSN tunnel.

The PSN header, MAC/Data-link and Physical Layer definitions are outside the scope of this framework.

The PSN Convergence Layer provides the enhancements needed to make the PSN conform to the assumed PSN service requirement. This layer therefore 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 can be any PSN type defined by the IETF. These are currently IPv4, IPv6 and MPLS.

4.2 Domain of PWE3

PWE3 defines the Encapsulation Layer, the method of carrying various payload types, and the interface to the PSN Tunnel Layer. It is expected that the other layers will be provided by tunneling methods such as L2TP or MPLS over the PSN.

4.3 Payload Types

The payload is classified into the following generic types of native data unit:

- o Bit-stream
- o Structured bit-stream
- o Cell
- o Packet

Within these generic types there are specific service types. For example:

Generic Payload Type	PW Service
Bit-stream	SONET, TDM (e.g. DS1, DS3, E1).
Structured bit-stream	SONET, TDM.
Cell	ATM.
Packet	Ethernet (all types), HDLC, frame relay, ATM AAL5 PDU.

4.3.1. 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. The Encapsulation Layer submits an identical number of bits for transport in each PW-PDU.

This service will require sequencing and real-time support.

4.3.2. Structured bit-stream

A 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:

- o Some part of the original (unstructured) bit stream is stripped by, for example, the PSN-bound direction of the NSP block. For example, in Structured SONET the section and line overhead (and, possibly, more) may be stripped.

- o 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 Encapsulation Layer may also perform silence/idle suppression or similar a 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 (i.e. structured SONET). Note that "short" structures are undistinguishable from cells and may benefit from the use of cell encapsulations.

This service will require sequencing and real-time support.

4.3.3. Cell Payload

A cell payload is created by capturing, transporting and replaying groups of bits 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 53-octet cells carrying ATM AAL2, and the larger 188-octet DVB Transport Stream packets.

To reduce PSN tunnel header overhead, multiple cells may be concatenated into a single payload. The Encapsulation Layer may consider the payload complete on the expiry of a timer, or when a fixed number of cells have been received. The benefit of concatenating multiple PDUs should be weighed against the resulting larger penalty incurred by packet loss. In some cases, it may be appropriate for the Encapsulation Layer to perform a silence suppression or a similar compression.

The generic cell payload service will normally need sequence number support, and may also need real-time support. The cell generic payload service would not normally require fragmentation.

The Encapsulation Layer may apply some form of compression to some of these sub-types.

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. That is an NSP function, and the selection would therefore be made before the packet was presented to the PW Encapsulation Layer.

4.3.4. Packet Payload

A packet payload is one that operates by capturing, transporting and replaying groups of bits of varying sizes that are presented on the wire. The delineation of the packet boundaries is encapsulation-specific. Common examples of packet payloads are HDLC and Ethernet PDUs. 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. This is likely to be the case when a user is providing the service and attaching to the service provider via an Ethernet, or where nested pseudo-wires are involved. The pseudo-wire would in these cases require the use of the fragmentation support of the underlying PSN or PSN Convergence Layer.

A packet payload may need sequencing and real-time support.

In some instances 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, on the basis of the VLAN identifier. This is an NSP function, and this selection would therefore be made before the packet was presented to the PW Encapsulation Layer.

4.3.5. Principle of Minimum Intervention

To minimise the scope of information, and to improve the efficiency of data flow through the Encapsulation Layer, the payload should, where possible, be transported as received without modification.

5. PW Encapsulation

The PW Encapsulation Layer provides the necessary infrastructure to adapt the specific payload type being transported over the PW to the

PSN Tunneling Layer that is used to carry the PW over the PSN.

The PW Encapsulation Layer consists of three sub-layers:

- o Payload Convergence
- o Sequencing
- o Timing

The Payload Convergence Sub-layer is highly tailored to the specific payload type, but, by grouping a number of target payload types into a generic class, and then providing a single convergence sub-layer type common to the group, we achieve a reduction in the number of payload convergence sub-layer types. The provision of per-packet signalling and other out-of-band information (other than sequencing or timing) is undertaken by this layer.

The Sequencing Layer and the Timing Layer provide a generic services to the Payload Convergence Layer for all payload types.

5.1 Payload Convergence Layer

5.1.1. Encapsulation

The primary task of the Payload Convergence Layer is the encapsulation of the payload in PDUs. The native data units to be encapsulated may or may not contain L2 or L1 header information. 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 PSN tunnel header is not considered as part of the PW header.

It should be noted that not all such information needs to 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 destination PE during PW set-up.

5.1.2. Bearer Channel Types

The PW Encapsulation Layer and its associated signaling require one

or more of the following types of channel from its underlying PSN Tunnel and PSN Layers:

1. A reliable control channel for signaling line events, status indications, and, in some exceptional cases, CE-CE events which must be translated and sent reliably between PEs.

   For example, this capability is needed in [PPPoL2TP], because PPP negotiation has to be split between the two ends of the

   tunnel. PWE3 may also 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 reflected via DSCP/EXP bits for priority during transit. It may also use a bit in the tunnel header itself to indicate that packets received at the PE should be processed with higher quality of service.

3. A sequenced channel for data traffic that is intolerant to packet reordering (one classification for use could be for any non-IP traffic).

4. An un-sequenced channel for data traffic insensitive to packet order.

These channels should be carried "in band" with one another to as much of a degree as is reasonably possible on a PSN.

In some cases there is a need to synchronize some CE events with the data carried over a PW. This is especially the case with TDM circuits (e.g., on-hook/off-hook events in PSTN switches).

Bearer channel types not needed by the supported PWs need not be included in 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. Specification of a QoS framework common to all PW Service types needs further investigation.

5.2 Payload independent PW Encapsulation Layers

Two PWE3 Encapsulation Sub-layers provide a common service to all payload types: Sequencing and Timing. These services are optional and are only used if needed by a particular PW instance. If the service is not needed, the associated header may be omitted in order to conserve processing and network resources.

There will be instances where a specific payload type will be required to be transported with or without sequence and/or real-time support. For example, an invariant of frame relay transport is the preservation of packet order. However, where the frame relay service is itself only being used to carry IP, it may be desirable to relax that constraint in return for reduced per-packet processing cost.

The guiding principle is that, where possible an existing IETF protocol should be used to provide these services. Where a suitable protocol is not available, the existing protocol should be extended 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 requirements.

5.2.1. Sequencing

The sequencing function provides three services: frame ordering, frame duplication detection and frame loss detection. These are invariant properties of a physical wire. Support for sequencing depends on the payload type, and may be omitted if not needed.

The size of the sequence number space depends on the speed of the emulated service, and the maximum time of the transient conditions in the PSN. A sequence number space greater than 2^16-1 may therefore be needed.

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 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 out-of-sequence detection. Alternatively it can be noted that sequencing is a sub-set of the problem of delivering timed packets, and that a single combined mechanism such as [RTP] may be employed.

There are two possible misordering strategies:

- o Drop miss-ordered PW PDUs.

- o Try to sort PW PDUs into the correct order.

The choice of strategy will depend on:

- o How critical the loss of packets is to the operation of the PW (e.g. the acceptable bit error rate).

- o The speeds of the PW and PSN.

- o The acceptable delay (since delay must be introduced to reorder)

- o The 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 such services, some mechanism must be provided to ensure that duplicated frames will not be delivered to the destination CE. The mechanism may or may not be the same as the mechanism 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 received PW PDUs.

In some instances, a destination PE will have to assume that a PW PDU is lost, if it fails to arrive within a certain time. 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 that they were designed to travel over, and it can be necessary for the emulated service to duplicate these network characteristics as closely as possible, e.g. in delivering traffic with the same jitter, bit-rate and timing characteristics as it was sent.

In such cases, it is necessary for the receiving PE to play out the native traffic as it was received at the sending PE. This relies on timing information sent between the two PEs.

The Timing Sub-layer must therefore support two timing functions: clock recovery and timed payload delivery. A particular payload type may require either or both of these services.

5.2.1.1 Clock Recovery

Clock recovery is the extraction of output transmission bit timing information from the delivered packet stream, and requires a phase-locking mechanism. A physical wire provides this naturally, but it is a relatively complex task to extract this from a highly jittered source such as packet stream. It is therefore desirable that an existing real-time protocol such as [RTP] be used for this purpose, unless it can be shown that this is unsuitable for a particular payload type.

5.2.1.2 Timed delivery

Timed delivery is the delivery of non-contiguous PW PDUs to the PW output interface with a constant phase-shift relative to the input interface. The timing of the delivery may be relative to a clock derived from the packet stream via clock recovery, or via an external clock.

5.3 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 of Figure 7 should, where possible, use existing IETF standards and common work in progress. Where such protocols do not exist, the goal should be to call for the design of components that have the wider application within the IETF.

6. PSN Tunnel Layer

PWE3 places three service requirements on the underlying PSN:

- o Multiplexing
- o Segmentation and Reassembly
- o Length and Delivery

6.1 Multiplexing

The purpose of the PSN Tunnel Layer is to allow multiple PWs to originate and terminate at a single interface address within a PE. This minimizes complexity and conserves resources.

If a service in its native form is capable of grouping multiple circuits into a "trunk", e.g. multiple ATM VCs in a VP, multiple Ethernet VLANs in a port, or Multiple DS0 services within a T1 or E1, then a single PW may connect two end-trunks.

6.2 Segmentation and Reassembly

It is desirable to avoid the processing and storage overhead of packet segmentation and reassembly (SAR). One way to do this is to set the MTU of the links between the CEs and the corresponding PEs to a value smaller than (PW_Path_MTU - PW_header - PSN_tunnel_header),

if that is possible. If segmentation cannot be completely avoided at an encapsulating PE (because, for example, the length of a packet after encapsulation would exceed the PW_Path_MTU), the PDU may be dropped. In this case, the management plane of the encapsulating PE

may be notified. Alternatively the SAR mechanism in the underlying PSN may be used.

If the length of a L2/L1 frame, restored from a PW PDU, exceeds the MTU of the destination PWES, it must be dropped. In this case, the management plane of the destination PE may be notified.

6.3 Length and Delivery

PDU length and delivery is the function of the PSN Layer. Where a length service is not provided by the underlying PSN, this becomes a requirement of the PSN Convergence Layer.

The three PSN types within the scope of the IETF are IPv4, IPv6 and MPLS. IPv4 and IPv6 both provide the necessary switching, length and fragmentation services needed to support all IETF specified Transport protocols. When the PSN is IPv4 or IPv6, no PSN Convergence Layer is needed.

MPLS provides a switching service, but does not provide length or fragmentation information. When MPLS is used as the PSN, a suitable convergence layer providing length and fragmentation services is needed. The definition of this length and fragmentation service is outside the scope of PWE3, and should be undertaken by the MPLS WG.

7. Control Plane

This section describes PWE3 control plane services.

7.1 Set-up 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.

Set-up or teardown of a PW can be triggered by a CLI command from the management plane of a PE, or by signaling (i.e., set-up or teardown) of a PWES, e.g., an ATM SVC.

During the set-up process, the PEs need to exchange some information (i.e., learn each others' capabilities). The tunneling control

protocol may be extended to provide mechanisms to 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 explicitly allowed.

7.2 Status Monitoring

Some native services have mechanisms for status monitoring. For example, ATM supports OAM for this purpose. For such 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 is bi-directional, the corresponding emulated service can only be signaled up when the associated PWs, and PSN tunnels if any, are functional in both directions.

Because the two CEs of an emulated service are not adjacent, a failure may occur at a place such that one or both physical links between the CEs and PEs remain up. For example in Figure 1, 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 such native services, the corresponding PWE3 service must provide a failure notification mechanism.

Similarly, if a native service has notification mechanisms so that when a network failure is fixed, all the affected services will change status from "Down" to "Up", 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 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. If a misconnection occurs it can breach the integrity of the system. If a payload mismatch occurs it can disrupt the customer network. In both

instances, there are security concerns.

The services of the underlying tunneling mechanism, and its associated control protocol, can be used to mitigate this.

This area needs further study.

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 impact the working condition of an emulated service. For some payload types, packet loss, corruption, and out-of-order delivery can be mapped to a bit error 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 notification, if any.

7.3.5. Collective Status Notification

Status of a group of emulated services 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 services that go through that link will fail. It is likely that there exists a group of emulated services which all terminate at a remote CE. (There can be multiple such CEs). 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 some mechanism for notifying status changes of a group of emulated circuits. One possible approach 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 identical group ID. In status notification, that group ID can be used to refer all the emulated services in that group.

This should be a mechanism provided by the underlying tunneling

protocol.

7.4 Keep-alive

If a native service has a keep-alive mechanism, the corresponding emulated service needs to use a mechanism to propagate this across the PW. One strategy is to transparently transport keep-alive messages over the PW. Another strategy is to piggy-back them on the

tunnel signaling mechanism. The principle of minimum intervention implies that the former strategy is the preferred approach.

7.5 Handling Control Messages of the Native Services

Some native services use control messages for maintaining the circuits. These control messages may be in-band, e.g. Ethernet flow control or ATM performance management, or out-of-band, e.g. the signaling VC of an ATM VP.

From 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.

If control messages are passed through, it may be desirable to send them using a reliable channel provided by the PSN tunnel layer. See Bearer Channel Types.

8. IANA considerations

There are no IANA considerations for this document.

9. Security Considerations

PWE3 provides no means of protecting the contents or delivery of the native data units. PWE3 may, however, leverage security mechanisms provided by the PSN Tunnel Layer. This section addresses the PWE3 vulnerabilities, and the mechanisms available to protect the native services.

Vulnerabilities exist at the tunnel end-point, the PW Encapsulation Layer, and the payload of the native service.

The security aspects of PWE3 need further study.

9.1 Tunnel End-Point Security

Protection mechanisms must be considered for the tunnel end-point in order to avoid denial-of-service attacks to the native service, and to prevent spoofing of the native data units. Exploitation of vulnerabilities from within the PSN may be directed to the tunnel end-point such that PSN tunnel services are disrupted. Controlling PSN access to the tunnel end-point may protect against this.

By restricting Tunnel End-point access to legitimate remote PE sources of traffic destined for the Tunnel End-point, the PE may reject traffic that interferes with the PSN tunnel services.

9.2 Validation of PW Encapsulation

Protection mechanisms must address the spoofing of tunneled PW data. The validation of traffic addressed to the tunnel end-point is paramount in ensuring integrity of PW encapsulation. Security protocols such as IPSec may be used by the PSN Tunnel Layer in order to maintain the integrity of the PW by authenticating data between the PE Tunnel End-points. IPSec may provide authentication, integrity, non-repudiation, and confidentiality of data transferred between two PE. It cannot provide the equivalent services to the native service.

Based on the type of data being transferred, the PW may indicate to the PSN Tunnel Layer that enhanced security services are required. The PSN Tunnel 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 Tunnel End-point may use peer authentication for every PSN packet to prevent spoofed native data units from being sent to the destination CE.

9.3 End to End Security

Protection of the PW encapsulated data stream between PE should not be considered equivalent to end-to-end security since the CE-PE interface and the PE processing element remains unprotected. PW service emulation does not preclude the application of additional security mechanisms such as IPSec that are implemented end-to-end. Likewise, end-to-end security mechanisms applied in the native service do not protect the PSN tunnel services provided by the PE for PW encapsulation.

Acknowledgments

We thank Sasha Vainshtein for his work on Native Service Processing and advice on bit-stream over PW services. We thank Scott Wainner and Stephen Casner for their comments and contributions.

References

Internet-drafts are works in progress available from <http://www.ietf.org/internet-drafts/>

[PPPoL2TP] PPP Tunneling Using Layer Two Tunneling Protocol, J Lau et al. <draft-ietf-l2tpext-l2tp-ppp-01.txt>, work in progress.

[RTP] RFC-1889: A Transport Protocol for Real-Time Applications, H. Schulzrinne et al.

Authors' Addresses

Stewart Bryant
Cisco Systems,
4, The Square,

Stockley Park,
Uxbridge UB11 1BL, Email: stbryant@cisco.com
United Kingdom. Phone: +44-20-8756-8828

Danny McPherson Email: danny@tcb.net
TCB. Phone: +1-303-470-9257

Lloyd Wood
Cisco Systems,
4, The Square,
Stockley Park,
Uxbridge UB11 1BL, Email: lwood@cisco.com
United Kingdom. Phone:+44-20-8734-4236

W. Mark Townsley
Cisco Systems,
7025 Kit Creek Road,
PO Box 14987,
Research Triangle Park, Email: mark@townsley.net
NC 27709

Full copyright statement