Network Working Group
Request for Comments: 5087
Category: Informational

Y(J). Stein
R. Shashoua
R. Insler
M. Anavi
RAD Data Communications

December 2007

Time Division Multiplexing over IP (TDMoIP)

Status of This Memo

Abstract

Time Division Multiplexing over IP (TDMoIP) is a structure-aware method for transporting Time Division Multiplexed (TDM) signals using pseudowires (PWs). Being structure-aware, TDMoIP is able to ensure TDM structure integrity, and thus withstand network degradations better than structure-agnostic transport. Structure-aware methods can distinguish individual channels, enabling packet loss concealment and bandwidth conservation. Accesibility of TDM signaling facilitates mechanisms that exploit or manipulate signaling.

Table of Contents

1. Introduction

Telephony traffic is conventionally carried over connection-oriented synchronous or plesiochronous links (loosely called TDM circuits herein). With the proliferation of Packet Switched Networks (PSNs), transport of TDM services over PSN infrastructures has become desirable. Emulation of TDM circuits over the PSN can be carried out using pseudowires (PWs), as described in the PWE3 architecture [RFC3985]. This emulation must maintain service quality of native TDM; in particular voice quality, latency, timing, and signaling features must be similar to those of existing TDM networks, as described in the TDM PW requirements document [RFC4197].

Structure-Agnostic TDM over Packet (SAToP) [RFC4553] is a structure-agnostic protocol for transporting TDM over PSNs. The present document details TDM over IP (TDMoIP), a structure-aware method for TDM transport. In contrast to SAToP, structure-aware methods such as TDMoIP ensure the integrity of TDM structure and thus enable the PW to better withstand network degradations. Individual multiplexed channels become visible, enabling the use of per channel mechanisms for packet loss concealment and bandwidth conservation. TDM signaling also becomes accessible, facilitating mechanisms that exploit or manipulate this signaling.

Despite its name, the TDMoIP(R) protocol herein described may operate over several types of PSN, including UDP over IPv4 or IPv6, MPLS, Layer 2 Tunneling Protocol version 3 (L2TPv3) over IP, and pure Ethernet. Implementation specifics for particular PSNs are discussed in Section 4. Although the protocol should be more generally called TDMoPW and its specific implementations TDMoIP, TDMoMPLS, etc., we retain the nomenclature TDMoIP for consistency with earlier usage.

The interworking function that connects between the TDM and PSN worlds will be called a TDMoIP interworking function (IWF), and it may be situated at the provider edge (PE) or at the customer edge (CE). The IWF that encapsulates TDM and injects packets into the PSN will be called the PSN-bound interworking function, while the IWF that extracts TDM data from packets and generates traffic on a TDM network will be called the TDM-bound interworking function. Emulated TDM circuits are always point-to-point, bidirectional, and transport TDM at the same rate in both directions.

As with all PWs, TDMoIP PWs may be manually configured or set up using the PWE3 control protocol [RFC4447]. Extensions to the PWE3 control protocol required specifically for setup and maintenance of TDMoIP pseudowires are described in [TDM-CONTROL].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. TDM Structure and Structure-aware Transport

Although TDM circuits can be used to carry arbitrary bit-streams, there are standardized methods for carrying constant-length blocks of data called "structures". Familiar structures are the T1 or E1 frames [G704] of length 193 and 256 bits, respectively. By concatenation of consecutive T1 or E1 frames we can build higher level structures called superframes or multiframes. T3 and E3 frames [G704][G751] are much larger than those of T1 and E1, and even larger structures are used in the GSM Abis channel described in [TRAU]. TDM structures contain TDM data plus structure overhead; for example, the 193-bit T1 frame contains a single bit of structure overhead and 24 bytes of data, while the 32-byte E1 frame contains a byte of overhead and 31 data bytes.

Structured TDM circuits are frequently used to transport multiplexed channels. A single byte in the TDM frame (called a timeslot) is allocated to each channel. A frame of a channelized T1 carries 24 byte-sized channels, while an E1 frame consists of 31 channels. Since TDM frames are sent 8000 times per second, a single byte-sized channel carries 64 kbps.

TDM structures are universally delimited by placing an easily detectable periodic bit pattern, called the Frame Alignment Signal (FAS), in the structure overhead. The structure overhead may additionally contain error monitoring and defect indications. We will use the term "structured TDM" to refer to TDM with any level of structure imposed by an FAS. Unstructured TDM signifies a bit stream upon which no structure has been imposed, implying that all bits are available for user data.

SAToP [RFC4553] is a structure-agnostic protocol for transporting TDM using PWs. SAToP treats the TDM input as an arbitrary bit-stream, completely disregarding any structure that may exist in the TDM bit-stream. Hence, SAToP is ideal for transport of truly unstructured TDM, but is also suitable for transport of structured TDM when there is no need to protect structure integrity nor interpret or manipulate individual channels during transport. In particular, SAToP is the technique of choice for PSNs with negligible packet loss, and for applications that do not require discrimination between channels nor intervention in TDM signaling.

As described in [RFC4553], when a single SAToP packet is lost, an "all ones" pattern is played out to the TDM interface. This pattern is interpreted by the TDM end equipment as an Alarm Indication Signal (AIS), which, according to TDM standards [G826], immediately triggers a "severely errored second" event. As such events are considered highly undesirable, the suitability of SAToP is limited to extremely reliable and underutilized PSNs.

When structure-aware TDM transport is employed, it is possible to explicitly safeguard TDM structure during transport over the PSN, thus making possible to effectively conceal packet loss events. Structure-aware transport exploits at least some level of the TDM structure to enhance robustness to packet loss or other PSN shortcomings. Structure-aware TDM PWs are not required to transport structure overhead across the PSN; in particular, the FAS MAY be stripped by the PSN-bound IWF and MUST be regenerated by the TDM-bound IWF. However, structure overhead MAY be transported over the PSN, since it may contain information other than FAS.

In addition to guaranteeing maintenance of TDM synchronization, structure-aware TDM transport can also distinguish individual timeslots of channelized TDM, thus enabling sophisticated packet loss concealment at the channel level. TDM signaling also becomes visible, facilitating mechanisms that maintain or exploit this information. Finally, by taking advantage of TDM signaling and/or voice activity detection, structure-aware TDM transport makes bandwidth conservation possible.

There are three conceptually distinct methods of ensuring TDM structure integrity -- namely, structure-locking, structure-indication, and structure-reassembly. Structure-locking requires each packet to commence at the start of a TDM structure, and to contain an entire structure or integral multiples thereof. Structure-indication allows packets to contain arbitrary fragments of basic structures, but employs pointers to indicate where each structure commences. Structure-reassembly is only defined for channelized TDM; the PSN-bound IWF extracts and buffers individual channels, and the original structure is reassembled from the received constituents by the TDM-bound IWF.

All three methods of TDM structure preservation have their advantages. Structure-locking is described in [RFC5086], while the present document specifies both structure-indication (see Section 5.1) and structure-reassembly (see Section 5.2) approaches. Structure-indication is used when channels may be allocated statically, and/or when it is required to interwork with existing circuit emulation systems (CES) based on AAL1. Structure-reassembly is used when dynamic allocation of channels is desirable and/or when it is required to interwork with existing loop emulation systems (LES) based on AAL2.

Operation, administration, and maintenance (OAM) mechanisms are vital for proper TDM deployments. As aforementioned, structure-aware mechanisms may refrain from transporting structure overhead across the PSN, disrupting OAM functionality. It is beneficial to distinguish between two OAM cases, the "trail terminated" and the "trail extended" scenarios. A trail is defined to be the combination of data and associated OAM information transfer. When the TDM trail is terminated, OAM information such as error monitoring and defect indications are not transported over the PSN, and the TDM networks function as separate OAM domains. In the trail extended case, we transfer the OAM information over the PSN (although not necessarily in its native format). OAM will be discussed further in Section 6.

3. TDMoIP Encapsulation

The overall format of TDMoIP packets is shown in Figure 1.

```
+---------------------+
|     PSN Headers     |
+---------------------+
| TDMoIP Control Word |
+---------------------+
|   Adapted Payload   |
+---------------------+
```

Figure 1. Basic TDMoIP Packet Format

The PSN-specific headers are those of UDP/IP, L2TPv3/IP, MPLS or layer 2 Ethernet, and contain all information necessary for forwarding the packet from the PSN-bound IWF to the TDM-bound one. The PSN is assumed to be reliable enough and of sufficient bandwidth to enable transport of the required TDM data.

A TDMoIP IWF may simultaneously support multiple TDM PWs, and the TDMoIP IWF MUST maintain context information for each TDM PW. Distinct PWs are differentiated based on PW labels, which are carried in the PSN-specific layers. Since TDM is inherently bidirectional, the association of two PWs in opposite directions is required. The PW labels of the two directions MAY take different values.

In addition to the aforementioned headers, an OPTIONAL 12-byte RTP header may appear in order to enable explicit transfer of timing information. This usage is a purely formal reuse of the header format of [RFC3550]. RTP mechanisms, such as header extensions, contributing source (CSRC) list, padding, RTP Control Protocol (RTCP), RTP header compression, Secure RTP (SRTP), etc., are not applicable.

The RTP timestamp indicates the packet creation time in units of a common clock available to both communicating TDMoIP IWFs. When no common clock is available, or when the TDMoIP IWFs have sufficiently accurate local clocks or can derive sufficiently accurate timing without explicit timestamps, the RTP header SHOULD be omitted.

If RTP is used, the fixed RTP header described in [RFC3550] MUST immediately follow the control word for all PSN types except UDP/IP, for which it MUST precede the control word. The version number MUST be set to 2, the P (padding), X (header extension), CC (CSRC count), and M (marker) fields in the RTP header MUST be set to zero, and the payload type (PT) values MUST be allocated from the range of dynamic values. The RTP sequence number MUST be identical to the sequence number in the TDMoIP control word (see below). The RTP timestamp MUST be generated in accordance with the rules established in [RFC3550]; the clock frequency MUST be an integer multiple of 8 kHz, and MUST be chosen to enable timing recovery that conforms with the appropriate standards (see Section 7.2).

The 32-bit control word MUST appear in every TDMoIP packet. Its format, in conformity with [RFC4385], is depicted in Figure 2.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  RES  |L|R| M |RES|  Length   |         Sequence Number       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2. Structure of the TDMoIP Control Word

RES (4 bits) The first nibble of the control word MUST be set to zero when the PSN is MPLS, in order to ensure that the packet does not alias an IP packet when forwarding devices perform deep packet inspection. For PSNs other than MPLS, the first nibble MAY be set to zero; however, in earlier versions of TDMoIP this field contained a format identifier that was optionally used to specify the payload format.

L Local Failure (1 bit) The L flag is set when the IWF has detected or has been informed of a TDM physical layer fault impacting the TDM data being forwarded. In the "trail extended" OAM scenario the L flag MUST be set when the IWF detects loss of signal, loss of frame synchronization, or AIS. When the L flag is set the contents of the packet may not be meaningful, and the payload MAY be suppressed in order to conserve bandwidth. Once set, if the TDM fault is rectified the L flag MUST be cleared. Use of the L flag is further explained in Section 6.

R Remote Failure (1 bit) The R flag is set when the IWF has detected or has been informed, that TDM data is not being received from the remote TDM network, indicating failure of the reverse direction of the bidirectional connection. An IWF SHOULD generate TDM Remote Defect Indicator (RDI) upon receipt of an R flag indication. In the "trail extended" OAM scenario the R flag MUST be set when the IWF detects RDI. Use of the R flag is further explained in Section 6.

M Defect Modifier (2 bits) Use of the M field is optional; when used, it supplements the meaning of the L flag.

When L is cleared (indicating valid TDM data) the M field is used as follows:

0 0 indicates no local defect modification.
0 1 reserved.
1 0 reserved.
1 1 reserved.

When L is set (invalid TDM data) the M field is used as follows:

0 0 indicates a TDM defect that should trigger conditioning or AIS generation by the TDM-bound IWF.
0 1 indicates idle TDM data that should not trigger any alarm. If the payload has been suppressed then the preconfigured idle code should be generated at egress.
1 0 indicates corrupted but potentially recoverable TDM data.
1 1 reserved.

Use of the M field is further explained in Section 6.

RES (2 bits) These bits are reserved and MUST be set to zero.

Length (6 bits) is used to indicate the length of the TDMoIP packet (control word and payload), in case padding is employed to meet minimum transmission unit requirements of the PSN. It MUST be used if the total packet length (including PSN, optional RTP, control word, and payload) is less than 64 bytes, and MUST be set to zero when not used.

Sequence number (16 bits) The TDMoIP sequence number provides the common PW sequencing function described in [RFC3985], and enables detection of lost and misordered packets. The sequence number space is a 16-bit, unsigned circular space; the initial value of the sequence number SHOULD be random (unpredictable) for security purposes, and its value is incremented modulo 2^16 separately for each PW. Pseudocode for a sequence number processing algorithm that could be used by a TDM-bound IWF is provided in Appendix A.

In order to form the TDMoIP payload, the PSN-bound IWF extracts bytes from the continuous TDM stream, filling each byte from its most significant bit. The extracted bytes are then adapted using one of two adaptation algorithms (see Section 5), and the resulting adapted payload is placed into the packet.

4. Encapsulation Details for Specific PSNs

TDMoIP PWs may exploit various PSNs, including UDP/IP (both IPv4 and IPv6), L2TPv3 over IP (with no intervening UDP), MPLS, and layer-2 Ethernet. In the following subsections, we depict the packet format for these cases.

For MPLS PSNs, the format is aligned with those specified in [Y1413] and [Y1414]. For UDP/IP PSNs, the format is aligned with those specified in [Y1453] and [Y1452]. For transport over layer 2 Ethernet the format is aligned with [MEF8].

4.1. UDP/IP

ITU-T recommendation Y.1453 [Y1453] describes structure-agnostic and structure-aware mechanisms for transporting TDM over IP networks. Similarly, ITU-T recommendation Y.1452 [Y1452] defines structure-reassembly mechanisms for this purpose. Although the terminology used here differs slightly from that of the ITU, implementations of TDMoIP for UDP/IP PSNs as described herein will interoperate with implementations designed to comply with Y.1453 subclause 9.2.2 or Y.1452 clause 10.

For UDP/IPv4, the headers as described in [RFC768] and [RFC791] are prefixed to the TDMoIP data. The format is similar for UDP/IPv6, except the IP header described in [RFC2460] is used. The TDMoIP packet structure is depicted in Figure 3.

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       | IPVER |  IHL  |    IP TOS     |          Total Length         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         Identification        |Flags|      Fragment Offset    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |  Time to Live |    Protocol   |      IP Header Checksum       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     Source IP Address                         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                  Destination IP Address                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |      Source Port Number       |    Destination Port Number    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |           UDP Length          |         UDP Checksum          |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    opt|RTV|P|X|  CC   |M|     PT      |      RTP Sequence Number      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    opt|                            Timestamp                          |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    opt|                         SSRC identifier                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |  RES  |L|R| M |RES|  Length   |         Sequence Number       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       |                        Adapted Payload                        |
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3. TDMoIP Packet Format for UDP/IP

The first five rows are the IP header, the sixth and seventh rows are the UDP header. Rows 8 through 10 are the optional RTP header. Row 11 is the TDMoIP control word.

IPVER (4 bits) is the IP version number, e.g., IPVER=4 for IPv4.

IHL (4 bits) is the length in 32-bit words of the IP header, IHL=5.

IP TOS (8 bits) is the IP type of service.

Total Length (16 bits) is the length in bytes of header and data.

Identification (16 bits) is the IP fragmentation identification field.

Flags (3 bits) are the IP control flags and MUST be set to 2 in order to avoid fragmentation.

Fragment Offset (13 bits) indicates where in the datagram the fragment belongs and is not used for TDMoIP.

Time to Live (8 bits) is the IP time to live field. Datagrams with zero in this field are to be discarded.

Protocol (8 bits) MUST be set to 0x11 (17) to signify UDP.

IP Header Checksum (16 bits) is a checksum for the IP header.

Source IP Address (32 bits) is the IP address of the source.

Destination IP Address (32 bits) is the IP address of the destination.

Source and Destination Port Numbers (16 bits each)

Either the source UDP port or destination UDP port MAY be used to multiplex and demultiplex individual PWs between nodes. Architecturally [RFC3985], this makes the UDP port act as the PW Label. PW endpoints MUST agree upon use of either the source UDP or destination UDP port as the PW Label.

UDP ports MUST be manually configured by both endpoints of the PW. The configured source or destination port (one or the other, but not both) together with both the source and destination IP addresses uniquely identify the PW. When the source UDP port is used as the PW label, the destination UDP port number MUST be set to the IANA assigned value of 0x085E (2142). All UDP port values that function as PW labels SHOULD be in the range of dynamically allocated UDP port numbers (0xC000 through 0xFFFF).

While many UDP-based protocols are able to traverse middleboxes without dire consequences, the use of UDP ports as PW labels makes middlebox traversal more difficult. Hence, it is NOT RECOMMENDED to use UDP-based PWs where port-translating middleboxes are present between PW endpoints.

UDP Length (16 bits) is the length in bytes of UDP header and data.

UDP Checksum (16 bits) is the checksum of UDP/IP header and data. If not computed it MUST be set to zero.

4.2. MPLS

ITU-T recommendation Y.1413 [Y1413] describes structure-agnostic and structure-aware mechanisms for transporting TDM over MPLS networks. Similarly, ITU-T recommendation Y.1414 [Y1413] defines structure-reassembly mechanisms for this purpose. Although the terminology used here differs slightly from that of the ITU, implementations of TDMoIP for MPLS PSNs as described herein will interoperate with implementations designed to comply with Y.1413 subclause 9.2.2 or Y.1414 clause 10.

The MPLS header as described in [RFC3032] is prefixed to the control word and TDM payload. The packet structure is depicted in Figure 4.

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |            Tunnel Label               | EXP |S|     TTL       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |              PW label                 | EXP |1|     TTL       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |  RES  |L|R| M |RES|  Length   |         Sequence Number       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    opt|RTV|P|X|  CC   |M|     PT      |      RTP Sequence Number      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    opt|                            Timestamp                          |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    opt|                         SSRC identifier                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       |                        Adapted Payload                        |
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4. TDMoIP Packet Format for MPLS

The first two rows depicted above are the MPLS header; the third is the TDMoIP control word. Fields not previously described will now be explained.

Tunnel Label (20 bits) is the MPLS label that identifies the MPLS LSP used to tunnel the TDM packets through the MPLS network. The label can be assigned either by manual provisioning or via an MPLS control protocol. While transiting the MPLS network there may be zero, one, or several tunnel label rows. For label stack usage see [RFC3032].

EXP (3 bits) experimental field, may be used to carry Diffserv classification for tunnel labels.

S (1 bit) the stacking bit indicates MPLS stack bottom. S=0 for all tunnel labels, and S=1 for the PW label.

TTL (8 bits) MPLS Time to live.

PW Label (20 bits) This label MUST be a valid MPLS label, and MAY be configured or signaled.

4.3. L2TPv3

The L2TPv3 header defined in [RFC3931] is prefixed to the TDMoIP data. The packet structure is depicted in Figure 5.

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       | IPVER |  IHL  |    IP TOS     |          Total Length         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         Identification        |Flags|      Fragment Offset    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |  Time to Live |    Protocol   |      IP Header Checksum       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     Source IP Address                         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                  Destination IP Address                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     Session ID = PW label                     |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                      cookie 1 (optional)                      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                      cookie 2 (optional)                      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |  RES  |L|R| M |RES|  Length   |         Sequence Number       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    opt|RTV|P|X|  CC   |M|     PT      |      RTP Sequence Number      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    opt|                            Timestamp                          |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    opt|                         SSRC identifier                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       |                        Adapted Payload                        |
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5. TDMoIP Packet Format for L2TPv3

Rows 6 through 8 are the L2TPv3 header. Fields not previously described will now be explained.

Protocol (8 bits) is the IP protocol field. It must be set to 0x73 (115), the user port number that has been assigned to L2TP by IANA.

Session ID (32 bits) is the locally significant L2TP session identifier, and contains the PW label. The value 0 is reserved.

Cookie (32 or 64 bits) is an optional field that contains a randomly selected value that can be used to validate association of the received frame with the expected PW.

4.4. Ethernet

Metro Ethernet Forum Implementation Agreement 8 [MEF8] describes structure-agnostic and structure-aware mechanisms for transporting TDM over Ethernet networks. Implementations of structure-indicated TDMoIP as described herein will interoperate with implementations designed to comply with MEF 8 Section 6.3.3.

The TDMoIP payload is encapsulated in an Ethernet frame by prefixing the Ethernet destination and source MAC addresses, optional VLAN header, and Ethertype, and suffixing the four-byte frame check sequence. TDMoIP implementations MUST be able to receive both industry standard (DIX) Ethernet and IEEE 802.3 [IEEE802.3] frames and SHOULD transmit Ethernet frames.

Ethernet encapsulation introduces restrictions on both minimum and maximum packet size. Whenever the entire TDMoIP packet is less than 64 bytes, padding is introduced and the true length indicated by using the Length field in the control word. In order to avoid fragmentation, the TDMoIP packet MUST be restricted to the maximum payload size. For example, the length of the Ethernet payload for a UDP/IP encapsulation of AAL1 format payload with 30 PDUs per packet is 1472 bytes, which falls below the maximal permitted payload size of 1500 bytes.

Ethernet frames MAY be used for TDMoIP transport without intervening IP or MPLS layers, however, an MPLS-style label MUST always be present. In this four-byte header S=1, and all other non-label bits are reserved (set to zero in the PSN-bound direction and ignored in the TDM-bound direction). The Ethertype SHOULD be set to 0x88D8 (35032), the value allocated for this purpose by the IEEE, but MAY be set to 0x8847 (34887), the Ethertype of MPLS. The overall frame structure is as follows:

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                       |  Destination MAC Address
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                           Destination MAC Address (cont)              |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     Source MAC Address
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           Source MAC Address  (cont)  |   VLAN Ethertype (opt)        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |VLP|C|      VLAN ID (opt)      |         Ethertype             |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |              PW label                 | RES |1|    RES        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |  RES  |L|R| M |RES|  Length   |         Sequence Number       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    opt|RTV|P|X|  CC   |M|     PT      |      RTP Sequence Number      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    opt|                            Timestamp                          |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    opt|                         SSRC identifier                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       |                        Adapted Payload                        |
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     Frame Check Sequence                      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6. TDMoIP Packet Format for Ethernet

Rows 1 through 6 are the (DIX) Ethernet header; for 802.3 there may be additional fields, depending on the value of the length field, see [IEEE802.3]. Fields not previously described will now be explained.

Destination MAC Address (48 bits) is the globally unique address of a single station that is to receive the packet. The format is defined in [IEEE802.3].

Source MAC Address (48 bits) is the globally unique address of the station that originated the packet. The format is defined in [IEEE802.3].

VLAN Ethertype (16 bits) 0x8100 in this position indicates that optional VLAN tagging specified in [IEEE802.1Q] is employed, and that the next two bytes contain the VLP, C, and VLAN ID fields. VLAN tags may be stacked, in which case the two-byte field following the VLAN ID is once again a VLAN Ethertype.

VLP (3 bits) is the VLAN priority, see [IEEE802.1Q].

C (1 bit) the "canonical format indicator" being set, indicates that route descriptors appear; see [IEEE802.1Q].

VLAN ID (12 bits) the VLAN identifier uniquely identifies the VLAN to which the frame belongs. If zero, only the VLP information is meaningful. Values 1 and FFF are reserved. The other 4093 values are valid VLAN identifiers.

Ethertype (16 bits) is the protocol identifier, as allocated by the IEEE. The Ethertype SHOULD be set to 0x88D8 (35032), but MAY be set to 0x8847 (34887).

PW Label (20 bits) This label MUST be manually configured. The remainder of this row is formatted to resemble an MPLS label.

Frame Check Sequence (32 bits) is a Cyclic Redundancy Check (CRC) error detection field, calculated per [IEEE802.3].

5. TDMoIP Payload Types

As discussed at the end of Section 3, TDMoIP transports real-time streams by first extracting bytes from the stream, and then adapting these bytes. TDMoIP offers two different adaptation algorithms, one for constant-rate real-time traffic, and one for variable-rate real-time traffic.

For unstructured TDM, or structured but unchannelized TDM, or structured channelized TDM with all channels active all the time, a constant-rate adaptation is needed. In such cases TDMoIP uses structure-indication to emulate the native TDM circuit, and the adaptation is known as "circuit emulation". However, for channelized TDM wherein the individual channels (corresponding to "loops" in telephony terminology) are frequently inactive, bandwidth may be conserved by transporting only active channels. This results in variable-rate real-time traffic, for which TDMoIP uses structure-reassembly to emulate the individual loops, and the adaptation is known as "loop emulation".

TDMoIP uses constant-rate AAL1 [AAL1,CES] for circuit emulation, while variable-rate AAL2 [AAL2] is employed for loop emulation. The AAL1 mode MUST be used for structured transport of unchannelized data and SHOULD be used for circuits with relatively constant usage. In addition, AAL1 MUST be used when the TDM-bound IWF is required to maintain a high timing accuracy (e.g., when its timing is further distributed) and SHOULD be used when high reliability is required. AAL2 SHOULD be used for channelized TDM when bandwidth needs to be conserved, and MAY be used whenever usage of voice-carrying channels is expected to be highly variable.

Additionally, a third mode is defined specifically for efficient transport of High-Level Data Link Control (HDLC)-based Common Channel Signaling (CCS) carried in TDM channels.

The AAL family of protocols is a natural choice for TDM emulation. Although originally developed to adapt various types of application data to the rigid format of ATM, the mechanisms are general solutions to the problem of transporting constant or variable-rate real-time streams over a packet network.

Since the AAL mechanisms are extensively deployed within and on the edge of the public telephony system, they have been demonstrated to reliably transfer voice-grade channels, data and telephony signaling. These mechanisms are mature and well understood, and implementations are readily available.

Finally, simplified service interworking with legacy networks is a major design goal of TDMoIP. Re-use of AAL technologies simplifies interworking with existing AAL1- and AAL2-based networks.

5.1. AAL1 Format Payload

For the prevalent cases of unchannelized TDM, or channelized TDM for which the channel allocation is static, the payload can be efficiently encoded using constant-rate AAL1 adaptation. The AAL1 format is described in [AAL1] and its use for circuit emulation over ATM in [CES]. We briefly review highlights of AAL1 technology in Appendix B. In this section we describe the use of AAL1 in the context of TDMoIP.

```
+-------------+---------------+
|control word |    AAL1 PDU   |
+-------------+---------------+
```

Figure 7a. Single AAL1 PDU per TDMoIP Packet

```
+-------------+---------------+   +----------------+
|control word |    AAL1 PDU    |---|    AAL1 PDU    |
+-------------+---------------+   +----------------+
```

Figure 7b. Multiple AAL1 PDUs per TDMoIP Packet

In AAL1 mode the TDMoIP payload consists of at least one, and perhaps many, 48-byte "AAL1 PDUs", see Figures 7a and 7b. The number of PDUs MUST be pre-configured and MUST be chosen such that the overall packet size does not exceed the maximum allowed by the PSN (e.g., 30 for UDP/IP over Ethernet). The precise number of PDUs per packet is typically chosen taking latency and bandwidth constraints into account. Using a single PDU delivers minimal latency, but incurs the highest overhead. All TDMoIP implementations MUST support between 1 and 8 PDUs per packet for E1 and T1 circuits, and between 5 and 15 PDUs per packet for E3 and T3 circuits.

AAL1 differentiates between unstructured and structured data transfer, which correspond to structure-agnostic and structure-aware transport. For structure-agnostic transport, AAL1 provides no inherent advantage as compared to SAToP; however, there may be scenarios for which its use is desirable. For example, when it is necessary to interwork with an existing AAL1 ATM circuit emulation system, or when clock recovery based on AAL1-specific mechanisms is favored.

For structure-aware transport, [CES] defines two modes, structured and structured with Channel Associated Signaling (CAS). Structured AAL1 maintains TDM frame synchronization by embedding a pointer to the beginning of the next frame in the AAL1 PDU header. Similarly, structured AAL1 with CAS maintains TDM frame and multiframe synchronization by embedding a pointer to the beginning of the next multiframe. Furthermore, structured AAL1 with CAS contains a substructure including the CAS signaling bits.

5.2. AAL2 Format Payload

Although AAL1 may be configured to transport fractional E1 or T1 circuits, the allocation of channels to be transported must be static due to the fact that AAL1 transports constant-rate bit-streams. It is often the case that not all the channels in a TDM circuit are simultaneously active ("off-hook"), and activity status may be determined by observation of the TDM signaling channel. Moreover, even during active calls, about half the time is silence that can be identified using voice activity detection (VAD). Using the variable-rate AAL2 mode, we may dynamically allocate channels to be transported, thus conserving bandwidth.

The AAL2 format is described in [AAL2] and its use for loop emulation over ATM is explained in [SSCS,LES]. We briefly review highlights of AAL2 technology in Appendix C. In this section, we describe the use of AAL2 in the context of TDMoIP.

```
+------------+--------------+   +---------------+
|control word |    AAL2 PDU    |---|    AAL2 PDU    |
+------------+--------------+   +---------------+
```

Figure 8. Concatenation of AAL2 PDUs in a TDMoIP Packet

In AAL2 mode the TDMoIP payload consists of one or more variable-length "AAL2 PDUs", see Figure 8. Each AAL2 PDU contains 3 bytes of overhead and between 1 and 64 bytes of payload. A packet may be constructed by inserting PDUs corresponding to all active channels, by appending PDUs ready at a certain time, or by any other means. Hence, more than one PDU belonging to a single channel may appear in a packet.

[RFC3985] denotes as Native Service Processing (NSP) functions all processing of the TDM data before its use as payload. Since AAL2 is inherently variable rate, arbitrary NSP functions MAY be performed before the channel is placed in the AAL2 loop emulation payload. These include testing for on-hook/off-hook status, voice activity detection, speech compression, fax/modem/tone relay, etc.

All mechanisms described in [AAL2,SSCS,LES] may be used for TDMoIP. In particular, channel identifier (CID) encoding and use of PAD octets according to [AAL2], encoding formats defined in [SSCS], and transport of CAS and CCS signaling as described in [LES] MAY all be used in the PSN-bound direction, and MUST be supported in the TDM-bound direction. The overlap functionality and AAL-CU timer and related functionalities may not be required, and the STF (start field) is NOT used. Computation of error detection codes -- namely, the Header Error Check (HEC) in the AAL2 PDU header and the CRC in the CAS packet -- is superfluous if an appropriate error detection mechanism is provided by the PSN. In such cases, these fields MAY be set to zero.

5.3. HDLC Format Payload

The motivation for handling HDLC in TDMoIP is to efficiently transport common channel signaling (CCS) such as SS7 [SS7] or ISDN PRI signaling [ISDN-PRI], embedded in the TDM stream. This mechanism is not intended for general HDLC payloads, and assumes that the HDLC messages are always shorter than the maximum packet size.

The HDLC mode should only be used when the majority of the bandwidth of the input HDLC stream is expected to be occupied by idle flags. Otherwise, the CCS channel should be treated as an ordinary channel.

The HDLC format is intended to operate in port mode, transparently passing all HDLC data and control messages over a separate PW. The encapsulation is compatible with that of [RFC4618], however the sequence number generation and processing SHOULD be performed according to Section 3 above.

The PSN-bound IWF monitors flags until a frame is detected. The contents of the frame are collected and the Frame Check Sequence (FCS) tested. If the FCS is incorrect, the frame is discarded; otherwise, the frame is sent after initial or final flags and FCS have been discarded and zero removal has been performed. When a TDMoIP-HDLC frame is received, its FCS is recalculated, and the original HDLC frame reconstituted.

6. TDMoIP Defect Handling

Native TDM networks signify network faults by carrying indications of forward defects (AIS) and reverse defects (RDI) in the TDM bit stream. Structure-agnostic TDM transport transparently carries all such indications; however, for structure-aware mechanisms where the PSN-bound IWF may remove TDM structure overhead carrying defect indications, explicit signaling of TDM defect conditions is required.

We saw in Section 3 that defects can be indicated by setting flags in the control word. This insertion of defect reporting into the packet rather than in a separate stream mimics the behavior of native TDM OAM mechanisms that carry such indications as bit patterns embedded in the TDM stream. The flags are designed to address the urgent messaging, i.e., messages whose contents must not be significantly delayed with respect to the TDM data that they potentially impact. Mechanisms for slow OAM messaging are discussed in Appendix D.

```
 +---+   +-----+   +------+   +-----+   +------+   +-----+   +---+
 |TDM|->-|     |->-|TDMoIP|->-|     |->-|TDMoIP|->-|     |->-|TDM|
 |   |   |TDM 1|   |      |   | PSN |   |      |   |TDM 2|   |   |
 |ES1|-<-|     |-<-| IWF1 |-<-|     |-<-| IWF2 |-<-|     |-<-|ES2|
 +---+   +-----+   +------+   +-----+   +------+   +-----+   +---+
```

Figure 9. Typical TDMoIP Network Configuration

The operation of TDMoIP defect handling is best understood by considering the downstream TDM flow from TDM end system 1 (ES1) through TDM network 1, through TDMoIP IWF 1 (IWF1), through the PSN, through TDMoIP IWF 2 (IWF2), through TDM network 2, towards TDM end system 2 (ES2), as depicted in the figure. We wish not only to detect defects in TDM network 1, the PSN, and TDM network 2, but to localize such defects in order to raise alarms only in the appropriate network.

In the "trail terminated" OAM scenario, only user data is exchanged between TDM network 1 and TDM network 2. The IWF functions as a TDM trail termination function, and defects detected in TDM network 1 are not relayed to network 2, or vice versa.

In the "trail extended" OAM scenario, if there is a defect (e.g., loss of signal or loss of frame synchronization) anywhere in TDM network 1 before the ultimate link, the following TDM node will generate AIS downstream (towards TDMoIP IWF1). If a break occurs in the ultimate link, the IWF itself will detect the loss of signal. In either case, IWF1 having directly detected lack of validity of the TDM signal, or having been informed of an earlier problem, raises the local ("L") defect flag in the control word of the packets it sends across the PSN. In this way the trail is extended to TDM network 2 across the PSN.

Unlike forward defect indications that are generated by all network elements, reverse defect indications are only generated by trail termination functions. In the trail terminated scenario, IWF1 serves as a trail termination function for TDM network 1, and thus when IWF1 directly detects lack of validity of the TDM signal, or is informed of an earlier problem, it MAY generate TDM RDI towards TDM ES1. In the trail extended scenario IWF1 is not a trail termination, and hence MUST NOT generate TDM RDI, but rather, as we have seen, sets the L defect flag. As we shall see, this will cause the AIS indication to reach ES2, which is the trail termination, and which MAY generate TDM RDI.

When the L flag is set there are four possibilities for treatment of payload content. The default is for IWF1 to fill the payload with the appropriate amount of AIS (usually all-ones) data. If the AIS has been generated before the IWF this can be accomplished by copying the received TDM data; if the penultimate TDM link fails and the IWF needs to generate the AIS itself. Alternatively, with structure-aware transport of channelized TDM one SHOULD fill the payload with "trunk conditioning"; this involves placing a preconfigured "out of service" code in each individual channel (the "out of service" code may differ between voice and data channels). Trunk conditioning MUST be used when channels taken from several TDM PWs are combined by the TDM-bound IWF into a single TDM circuit. The third possibility is to suppress the payload altogether. Finally, if IWF1 believes that the TDM defect is minor or correctable (e.g., loss of multiframe synchronization, or initial phases of detection of incorrect frame sync), it MAY place the TDM data it has received into the payload field, and specify in the defect modification field ("M") that the TDM data is corrupted, but potentially recoverable.

When IWF2 receives a local defect indication without M field modification, it forwards (or generates if the payload has been suppressed) AIS or trunk conditioning towards ES2 (the choice between AIS and conditioning being preconfigured). Thus AIS has been properly delivered to ES2 emulating the TDM scenario from the TDM end system's point of view. In addition, IWF2 receiving the L flag uniquely specifies that the defect was in TDM network 1 and not in TDM network 2, thus suppressing alarms in the correctly functioning network.

If the M field indicates that the TDM has been marked as potentially recoverable, then implementation specific algorithms (not herein specified) may optionally be utilized to minimize the impact of transient defects on the overall network performance. If the M field indicates that the TDM is "idle", no alarms should be raised and IWF2 treats the payload contents as regular TDM data. If the payload has been suppressed, trunk conditioning and not AIS MUST be generated by IWF2.

The second case is when the defect is in TDM network 2. Such defects cause AIS generation towards ES2, which may respond by sending TDM RDI in the reverse direction. In the trail terminated scenario this RDI is restricted to network 2. In the trail extended scenario, IWF2 upon observing this RDI inserted into valid TDM data, MUST indicate this by setting the "R" flag in packets sent back across the PSN towards IWF1. IWF1, upon receiving this indication, generates RDI towards ES1, thus emulating a single conventional TDM network.

The final possibility is that of a unidirectional defect in the PSN. In such a case, TDMoIP IWF1 sends packets toward IWF2, but these are not received. IWF2 MUST inform the PSN's management system of this problem, and furthermore generate TDM AIS towards ES2. ES2 may respond with TDM RDI, and as before, in the trail extended scenario, when IWF2 detects RDI it MUST raise the "R" flag indication. When IWF1 receives packets with the "R" flag set it has been informed of a reverse defect, and MUST generate TDM RDI towards ES1.

In all cases, if any of the above defects persist for a preconfigured period (default value of 2.5 seconds) a service failure is declared. Since TDM PWs are inherently bidirectional, a persistent defect in either directional results in a bidirectional service failure. In addition, if signaling is sent over a distinct PW as per Section 5.3, both PWs are considered to have failed when persistent defects are detected in either.

When failure is declared the PW MUST be withdrawn, and both TDMoIP IWFs commence sending AIS (and not trunk conditioning) to their respective TDM networks. The IWFs then engage in connectivity testing using native methods or TDMoIP OAM as described in Appendix D until connectivity is restored.

7. Implementation Issues

General requirements for transport of TDM over pseudo-wires are detailed in [RFC4197]. In the following subsections we review additional aspects essential to successful TDMoIP implementation.

7.1. Jitter and Packet Loss

In order to compensate for packet delay variation that exists in any PSN, a jitter buffer MUST be provided. A jitter buffer is a block of memory into which the data from the PSN is written at its variable arrival rate, and data is read out and sent to the destination TDM equipment at a constant rate. Use of a jitter buffer partially hides the fact that a PSN has been traversed rather than a conventional synchronous TDM network, except for the additional latency. Customary practice is to operate with the jitter buffer approximately half full, thus minimizing the probability of its overflow or underflow. Hence, the additional delay equals half the jitter buffer size. The length of the jitter buffer SHOULD be configurable and MAY be dynamic (i.e., grow and shrink in length according to the statistics of the Packet Delay Variation (PDV)).

In order to handle (infrequent) packet loss and misordering, a packet sequence integrity mechanism MUST be provided. This mechanism MUST track the serial numbers of arriving packets and MUST take appropriate action when anomalies are detected. When lost packet(s) are detected, the mechanism MUST output filler data in order to retain TDM timing. Packets arriving in incorrect order SHOULD be reordered. Lost packet processing SHOULD ensure that proper FAS is sent to the TDM network. An example sequence number processing algorithm is provided in Appendix A.

While the insertion of arbitrary filler data may be sufficient to maintain the TDM timing, for telephony traffic it may lead to audio gaps or artifacts that result in choppy, annoying or even unintelligible audio. An implementation MAY blindly insert a preconfigured constant value in place of any lost samples, and this value SHOULD be chosen to minimize the perceptual effect. Alternatively one MAY replay the previously received packet. When computational resources are available, implementations SHOULD conceal the packet loss event by properly estimating missing sample values in such fashion as to minimize the perceptual error.

7.2. Timing Recovery

TDM networks are inherently synchronous; somewhere in the network there will always be at least one extremely accurate primary reference clock, with long-term accuracy of one part in 1E-11. This node provides reference timing to secondary nodes with somewhat lower accuracy, and these in turn distribute timing information further. This hierarchy of time synchronization is essential for the proper functioning of the network as a whole; for details see [G823][G824].

Packets in PSNs reach their destination with delay that has a random component, known as packet delay variation (PDV). When emulating TDM on a PSN, extracting data from the jitter buffer at a constant rate overcomes much of the high frequency component of this randomness ("jitter"). The rate at which we extract data from the jitter buffer is determined by the destination clock, and were this to be precisely matched to the source clock proper timing would be maintained. Unfortunately, the source clock information is not disseminated through a PSN, and the destination clock frequency will only nominally equal the source clock frequency, leading to low frequency ("wander") timing inaccuracies.

In broadest terms, there are four methods of overcoming this difficulty. In the first and second methods timing information is provided by some means independent of the PSN. This timing may be provided to the TDM end systems (method 1) or to the IWFs (method 2). In a third method, a common clock is assumed available to both IWFs, and the relationship between the TDM source clock and this clock is encoded in the packet. This encoding may take the form of RTP timestamps or may utilize the synchronous residual timestamp (SRTS) bits in the AAL1 overhead. In the final method (adaptive clock recovery) the timing must be deduced solely based on the packet arrival times. Example scenarios are detailed in [RFC4197] and in [Y1413].

Adaptive clock recovery utilizes only observable characteristics of the packets arriving from the PSN, such as the precise time of arrival of the packet at the TDM-bound IWF, or the fill-level of the jitter buffer as a function of time. Due to the packet delay variation in the PSN, filtering processes that combat the statistical nature of the observable characteristics must be employed. Frequency Locked Loops (FLL) and Phase Locked Loops (PLL) are well suited for this task.

Whatever timing recovery mechanism is employed, the output of the TDM-bound IWF MUST conform to the jitter and wander specifications of TDM traffic interfaces, as defined in [G823][G824]. For some applications, more stringent jitter and wander tolerances MAY be imposed.

7.3. Congestion Control

As explained in [RFC3985], the underlying PSN may be subject to congestion. Unless appropriate precautions are taken, undiminished demand of bandwidth by TDMoIP can contribute to network congestion that may impact network control protocols.

The AAL1 mode of TDMoIP is an inelastic constant bit-rate (CBR) flow and cannot respond to congestion in a TCP-friendly manner prescribed by [RFC2914], although the percentage of total bandwidth they consume remains constant. The AAL2 mode of TDMoIP is variable bit-rate (VBR), and it is often possible to reduce the bandwidth consumed by employing mechanisms that are beyond the scope of this document.

Whenever possible, TDMoIP SHOULD be carried across traffic-engineered PSNs that provide either bandwidth reservation and admission control or forwarding prioritization and boundary traffic conditioning mechanisms. IntServ-enabled domains supporting Guaranteed Service (GS) [RFC2212] and Diffserv-enabled domains [RFC2475] supporting Expedited Forwarding (EF) [RFC3246] provide examples of such PSNs. Such mechanisms will negate, to some degree, the effect of TDMoIP on neighboring streams. In order to facilitate boundary traffic conditioning of TDMoIP traffic over IP PSNs, the TDMoIP packets SHOULD NOT use the Diffserv Code Point (DSCP) value reserved for the Default Per-Hop Behavior (PHB) [RFC2474].

When TDMoIP is run over a PSN providing best-effort service, packet loss SHOULD be monitored in order to detect congestion. If congestion is detected and bandwidth reduction is possible, then such reduction SHOULD be enacted. If bandwidth reduction is not possible, then the TDMoIP PW SHOULD shut down bi-directionally for some period of time as described in Section 6.5 of [RFC3985].

Note that:

1. In AAL1 mode TDMoIP can inherently provide packet loss measurement since the expected rate of packet arrival is fixed and known.

2. The results of the packet loss measurement may not be a reliable indication of presence or absence of severe congestion if the PSN provides enhanced delivery. For example, if TDMoIP traffic takes precedence over other traffic, severe congestion may not significantly affect TDMoIP packet loss.

3. The TDM services emulated by TDMoIP have high availability objectives (see [G826]) that MUST be taken into account when deciding on temporary shutdown.

This specification does not define exact criteria for detecting severe congestion or specific methods for TDMoIP shutdown or subsequent re-start. However, the following considerations may be used as guidelines for implementing the shutdown mechanism:

1. If the TDMoIP PW has been set up using the PWE3 control protocol [RFC4447], the regular PW teardown procedures of these protocols SHOULD be used.

2. If one of the TDMoIP IWFs stops transmission of packets for a sufficiently long period, its peer (observing 100% packet loss) will necessarily detect "severe congestion" and also stop transmission, thus achieving bi-directional PW shutdown.

TDMoIP does not provide mechanisms to ensure timely delivery or provide other quality-of-service guarantees; hence it is required that the lower-layer services do so. Layer 2 priority can be bestowed upon a TDMoIP stream by using the VLAN priority field, MPLS priority can be provided by using EXP bits, and layer 3 priority is controllable by using TOS. Switches and routers which the TDMoIP stream must traverse should be configured to respect these priorities.

8. Security Considerations

TDMoIP does not enhance or detract from the security performance of the underlying PSN, rather it relies upon the PSN’s mechanisms for encryption, integrity, and authentication whenever required. The level of security provided may be less than that of a native TDM service.

When the PSN is MPLS, PW-specific security mechanisms MAY be required, while for IP-based PSNs, IPsec [RFC4301] MAY be used. TDMoIP using L2TPv3 is subject to the security considerations discussed in Section 8 of [RFC3931].

TDMoIP shares susceptibility to a number of pseudowire-layer attacks (see [RFC3985]) and implementations SHOULD use whatever mechanisms for confidentiality, integrity, and authentication are developed for general PWs. These methods are beyond the scope of this document.

Random initialization of sequence numbers, in both the control word and the optional RTP header, makes known-plaintext attacks on encrypted TDMoIP more difficult. Encryption of PWs is beyond the scope of this document.

PW labels SHOULD be selected in an unpredictable manner rather than sequentially or otherwise in order to deter session hijacking. When using L2TPv3, a cryptographically random [RFC4086] Cookie SHOULD be used to protect against off-path packet insertion attacks, and a 64-bit Cookie is RECOMMENDED for protection against brute-force, blind, insertion attacks.

Although TDMoIP MAY employ an RTP header when explicit transfer of timing information is required, SRTP (see [RFC3711]) mechanisms are not applicable.

9. IANA Considerations

For MPLS PSNs, PW Types for TDMoIP PWs are allocated in [RFC4446].

For UDP/IP PSNs, when the source port is used as PW label, the destination port number MUST be set to 0x085E (2142), the user port number assigned by IANA to TDMoIP.

10. Applicability Statement

It must be recognized that the emulation provided by TDMoIP may be imperfect, and the service may differ from the native TDM circuit in the following ways.

The end-to-end delay of a TDM circuit emulated using TDMoIP may exceed that of a native TDM circuit.

When using adaptive clock recovery, the timing performance of the emulated TDM circuit depends on characteristics of the PSN, and thus may be inferior to that of a native TDM circuit.

If the TDM structure overhead is not transported over the PSN, then non-FAS data in the overhead will be lost.

When packets are lost in the PSN, TDMoIP mechanisms ensure that frame synchronization will be maintained. When packet loss events are properly concealed, the effect on telephony channels will be perceptually minimized. However, the bit error rate will be degraded as compared to the native service.

Data in inactive channels is not transported in AAL2 mode, and thus this data will differ from that of the native service.

Native TDM connections are point-to-point, while PSNs are shared infrastructures. Hence, the level of security of the emulated service may be less than that of the native service.

11. Acknowledgments

The authors would like to thank Hugo Silberman, Shimon HaLevy, Tuvia Segal, and Eitan Schwartz of RAD Data Communications for their invaluable contributions to the technology described herein.

Appendix A. Sequence Number Processing (Informative)

The sequence number field in the control word enables detection of lost and misordered packets. Here we give pseudocode for an example algorithm in order to clarify the issues involved. These issues are implementation specific and no single explanation can capture all the possibilities.

In order to simplify the description, modulo arithmetic is consistently used in lieu of ad-hoc treatment of the cyclicity. All differences between indexes are explicitly converted to the range [-2^15 ... +2^15 - 1] to ensure that simple checking of the difference’s sign correctly predicts the packet arrival order.

Furthermore, we introduce the notion of a playout buffer in order to unambiguously define packet lateness. When a packet arrives after previously having been assumed lost, the TDM-bound IWF may discard it, and continue to treat it as lost. Alternatively, if the filler data that had been inserted in its place has not yet been played out, the option remains to insert the true data into the playout buffer. Of course, the filler data may be generated upon initial detection of a missing packet or upon playout. This description is stated in terms of a packet-oriented playout buffer rather than a TDM byte oriented one; however, this is not a true requirement for re-ordering implementations since the latter could be used along with pointers to packet commencement points.

Having introduced the playout buffer we explicitly treat over-run and under-run of this buffer. Over-run occurs when packets arrive so quickly that they can not be stored for playout. This is usually an indication of gross timing inaccuracy or misconfiguration, and we can do little but discard such early packets. Under-run is usually a sign of network starvation, resulting from congestion or network failure.

The external variables used by the pseudocode are:

received:	sequence number of packet received
played:	sequence number of the packet being played out (Note 1)
over-run:	is the playout buffer full? (Note 3)
under-run:	has the playout buffer been exhausted? (Note 3)

The internal variables used by the pseudocode are:

expected: sequence number we expect to receive next
D: difference between expected and received (Note 2)
L: difference between sequence numbers of packet being played out and that just received (Notes 1 and 2)

In addition, the algorithm requires one parameter:

R: maximum lateness for a packet to be recoverable (Note 1).

Note 1: this is only required for the optional re-ordering
Note 2: this number is always in the range -2^15 ... +2^15 - 1
Note 3: the playout buffer is emptied by the TDM playout process, which runs asynchronously to the packet arrival processing, and which is not herein specified

Sequence Number Processing Algorithm

```
Upon receipt of a packet
  if received = expected
    { treat packet as in-order }
    if not over-run then
      place packet contents into playout buffer
    else
      discard packet contents
    set expected = (received + 1) mod 2^16
  else
    calculate D = ( (expected-received) mod 2^16 ) - 2^15
    if D > 0 then
      { packets expected, expected+1, ... received-1 are lost }
      while not over-run
        place filler (all-ones or interpolation) into playout buffer
        if not over-run then
          place packet contents into playout buffer
        else
          discard packet contents
        set expected = (received + 1) mod 2^16
    else  { late packet arrived }
      declare "received" to be a late packet
      do NOT update "expected"
      either
        discard packet
      or
        if not under-run then
          calculate L = ( (played-received) mod 2^16 ) - 2^15
          if 0 < L <= R then
            replace data from packet previously marked as lost
          else
            discard packet
Note: by choosing R=0 we always discard the late packet
```

Appendix B. AAL1 Review (Informative)

The first byte of the 48-byte AAL1 PDU always contains an error-protected 3-bit sequence number.

```
 1 2 3 4 5 6 7 8
+-+-+-+-+-+-+-+-+-----------------------
|C| SN  | CRC |P| 47 bytes of payload
+-+-+-+-+-+-+-+-+-----------------------
```

C (1 bit) convergence sublayer indication, its use here is limited to indication of the existence of a pointer (see below); C=0 means no pointer, C=1 means a pointer is present.

SN (3 bits) The AAL1 sequence number increments from PDU to PDU.

CRC (3 bits) is a 3-bit error cyclic redundancy code on C and SN.

P (1 bit) even byte parity.

As can be readily inferred, incrementing the sequence number forms an eight-PDU sequence number cycle, the importance of which will become clear shortly.

The structure of the remaining 47 bytes in the AAL1 PDU depends on the PDU type, of which there are three, corresponding to the three types of AAL1 circuit emulation service defined in [CES]. These are known as unstructured circuit emulation, structured circuit emulation, and structured circuit emulation with CAS.

The simplest PDU is the unstructured one, which is used for transparent transfer of whole circuits (T1,E1,T3,E3). Although AAL1 provides no inherent advantage as compared to SAToP for unstructured transport, in certain cases AAL1 may be required or desirable. For example, when it is necessary to interwork with an existing AAL1-based network, or when clock recovery based on AAL1-specific mechanisms is favored.

For unstructured AAL1, the 47 bytes after the sequence number byte contain the full 376 bits from the TDM bit stream. No frame synchronization is supplied or implied, and framing is the sole responsibility of the end-user equipment. Hence, the unstructured mode can be used to carry data, and for circuits with nonstandard frame synchronization. For the T1 case the raw frame consists of 193 bits, and hence 1 183/193 T1 frames fit into each AAL1 PDU. The E1 frame consists of 256 bits, and so 1 15/32 E1 frames fit into each PDU.

When the TDM circuit is channelized according to [G704], and in particular when it is desired to fractional E1 or T1, it is advantageous to use one of the structured AAL1 circuit emulation services. Structured AAL1 views the data not merely as a bit stream, but as a bundle of channels. Furthermore, when CAS signaling is used it can be formatted so that it can be readily detected and manipulated.

In the structured circuit emulation mode without CAS, N bytes from the N channels to be transported are first arranged in order of channel number. Thus if channels 2, 3, 5, 7 and 11 are to be transported, the corresponding five bytes are placed in the PDU immediately after the sequence number byte. This placement is repeated until all 47 bytes in the PDU are filled.

byte	1	2	3	4	5	6	7	8	9	10	---	41	42	43	44	45	46	47
channel	2	3	5	7	11	2	3	5	7	11	---	2	3	5	7	11	2	3

The next PDU commences where the present PDU left off.

byte	1	2	3	4	5	6	7	8	9	10	---	41	42	43	44	45	46	47
channel	5	7	11	2	3	5	7	11	2	3	---	5	7	11	2	3	5	7

And so forth. The set of channels 2,3,5,7,11 is the basic structure and the point where one structure ends and the next commences is the structure boundary.

The problem with this arrangement is the lack of explicit indication of the byte identities. As can be seen in the above example, each AAL1 PDU starts with a different channel, so a single lost packet will result in misidentifying channels from that point onwards, without possibility of recovery. The solution to this deficiency is the periodic introduction of a pointer to the next structure boundary. This pointer need not be used too frequently, as the channel identifications are uniquely inferable unless packets are lost.

The particular method used in AAL1 is to insert a pointer once every sequence number cycle of eight PDUs. The pointer is seven bits and protected by an even parity MSB (most significant bit), and so occupies a single byte. Since seven bits are sufficient to represent offsets larger than 47, we can limit the placement of the pointer byte to PDUs with even sequence numbers. Unlike most AAL1 PDUs that contain 47 TDM bytes, PDUs that contain a pointer (P-format PDUs) have the following format.

```
 0                   1
 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-----------------------
|C| SN  | CRC |P|E|   pointer   | 46 bytes of payload
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-----------------------
```

where

C (1 bit) convergence sublayer indication, C=1 for P-format PDUs.

SN (3 bits) is an even AAL1 sequence number.

CRC (3 bits) is a 3-bit error cyclic redundancy code on C and SN.

P (1 bit) even byte parity LSB (least significant bit) for sequence number byte.

E (1 bit) even byte parity MSB for pointer byte.

pointer (7 bits) pointer to next structure boundary.

Since P-format PDUs have 46 bytes of payload and the next PDU has 47 bytes, viewed as a single entity the pointer needs to indicate one of 93 bytes. If P=0 it is understood that the structure commences with the following byte (i.e., the first byte in the payload belongs to the lowest numbered channel). P=93 means that the last byte of the second PDU is the final byte of the structure, and the following PDU commences with a new structure. The special value P=127 indicates that there is no structure boundary to be indicated (needed when extremely large structures are being transported).

The P-format PDU is always placed at the first possible position in the sequence number cycle that a structure boundary occurs, and can only occur once per cycle.

The only difference between the structured circuit emulation format and structured circuit emulation with CAS is the definition of the structure. Whereas in structured circuit emulation the structure is composed of the N channels, in structured circuit emulation with CAS the structure encompasses the superframe consisting of multiple repetitions of the N channels and then the CAS signaling bits. The CAS bits are tightly packed into bytes and the final byte is padded with zeros if required.

For example, for E1 circuits the CAS signaling bits are updated once per superframe of 16 frames. Hence, the structure for N*64 derived from an E1 with CAS signaling consists of 16 repetitions of N bytes, followed by N sets of the four ABCD bits, and finally four zero bits if N is odd. For example, the structure for channels 2,3 and 5 will be as follows:

```
    2 3 5 2 3 5 2 3 5 2 3 5 2 3 5 2 3 5 2 3 5 2 3 5 2 3 5 2 3 5 2 3 5
    2 3 5 2 3 5 2 3 5 2 3 5 2 3 5 [ABCD2 ABCD3] [ABCD5 0000]
```

Similarly for T1 ESF circuits the superframe is 24 frames, and the structure consists of 24 repetitions of N bytes, followed by the ABCD bits as before. For the T1 case the signaling bits will in general appear twice, in their regular (bit-robbed) positions and at the end of the structure.

Appendix C. AAL2 Review (Informative)

The basic AAL2 PDU is:

```
     |    Byte  1    |    Byte  2    |    Byte  3    |
      0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+------------
     |      CID      |     LI    |  UUI    |   HEC   |   PAYLOAD
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+------------
```

CID (8 bits) channel identifier is an identifier that must be unique for the PW. The values 0-7 are reserved for special purposes, (and if interworking with VoDSL is required, so are values 8 through 15 as specified in [LES]), thus leaving 248 (240) CIDs per PW. The mapping of CID values to channels MAY be manually configured manually or signaled.

LI (6 bits) length indicator is one less than the length of the payload in bytes. Note that the payload is limited to 64 bytes.

UUI (5 bits) user-to-user indication is the higher layer (application) identifier and counter. For voice data, the UUI will always be in the range 0-15, and SHOULD be incremented modulo 16 each time a channel buffer is sent. The receiver MAY monitor this sequence. UUI is set to 24 for CAS signaling packets.

HEC (5 bits) the header error control

Payload - voice
A block of length indicated by LI of voice samples are placed as-is into the AAL2 packet.

Payload - CAS signaling
For CAS signaling the payload is formatted as an AAL2 "fully protected" (type 3) packet (see [AAL2]) in order to ensure error protection. The signaling is sent with the same CID as the corresponding voice channel. Signaling MUST be sent whenever the state of the ABCD bits changes, and SHOULD be sent with triple redundancy, i.e., sent three times spaced 5 milliseconds apart. In addition, the entire set of the signaling bits SHOULD be sent periodically to ensure reliability.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|RED|       timestamp             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  RES  | ABCD  |    type   | CRC
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    CRC (cont)  |
+-+-+-+-+-+-+-+-+
```

RED (2 bits) is the triple redundancy counter. For the first packet it takes the value 00, for the second 01 and for the third 10. RED=11 means non-redundant information, and is used when triple redundancy is not employed, and for periodic refresh messages.

Timestamp (14 bits) The timestamp is optional and in particular is not needed if RTP is employed. If not used, the timestamp MUST be set to zero. When used with triple redundancy, it MUST be the same for all three redundant transmissions.

RES (4 bits) is reserved and MUST be set to zero.

ABCD (4 bits) are the CAS signaling bits.

type (6 bits) for CAS signaling this is 000011.

CRC-10 (10 bits) is a 10-bit CRC error detection code.

Appendix D. Performance Monitoring Mechanisms (Informative)

PWs require OAM mechanisms to monitor performance measures that impact the emulated service. Performance measures, such as packet loss ratio and packet delay variation, may be used to set various parameters and thresholds; for TDMoIP PWs adaptive timing recovery and packet loss concealment algorithms may benefit from such information. In addition, OAM mechanisms may be used to collect statistics relating to the underlying PSN [RFC2330], and its suitability for carrying TDM services.

TDMoIP IWFs may benefit from knowledge of PSN performance metrics, such as round trip time (RTT), packet delay variation (PDV) and packet loss ratio (PLR). These measurements are conventionally performed by a separate flow of packets designed for this purpose, e.g., ICMP packets [RFC792] or MPLS LSP ping packets [RFC4379] with multiple timestamps. For AAL1 mode, TDMoIP sends packets across the PSN at a constant rate, and hence no additional OAM flow is required for measurement of PDV or PLR. However, separate OAM flows are required for RTT measurement, for AAL2 mode PWs, for measurement of parameters at setup, for monitoring of inactive backup PWs, and for low-rate monitoring of PSNs after PWs have been withdrawn due to service failures.

If the underlying PSN has appropriate maintenance mechanisms that provide connectivity verification, RTT, PDV, and PLR measurements that correlate well with those of the PW, then these mechanisms SHOULD be used. If such mechanisms are not available, either of two similar OAM signaling mechanisms may be used. The first is internal to the PW and based on inband VCCV [RFC5085], and the second is defined only for UDP/IP PSNs, and is based on a separate PW. The latter is particularly efficient for a large number of fate-sharing TDM PWs.

D.1. TDMoIP Connectivity Verification

In most conventional IP applications a server sends some finite amount of information over the network after explicit request from a client. With TDMoIP PWs the PSN-bound IWF could send a continuous stream of packets towards the destination without knowing whether the TDM-bound IWF is ready to accept them. For layer-2 networks, this may lead to flooding of the PSN with stray packets.

This problem may occur when a TDMoIP IWF is first brought up, when the TDM-bound IWF fails or is disconnected from the PSN, or the PW is broken. After an aging time the destination IWF becomes unknown, and intermediate switches may flood the network with the TDMoIP packets in an attempt to find a new path.

The solution to this problem is to significantly reduce the number of TDMoIP packets transmitted per second when PW failure is detected, and to return to full rate only when the PW is available. The detection of failure and restoration is made possible by the periodic exchange of one-way connectivity-verification messages.

Connectivity is tested by periodically sending OAM messages from the source IWF to the destination IWF, and having the destination reply to each message. The connectivity verification mechanism SHOULD be used during setup and configuration. Without OAM signaling, one must ensure that the destination IWF is ready to receive packets before starting to send them. Since TDMoIP IWFs operate full-duplex, both would need to be set up and properly configured simultaneously if flooding is to be avoided. When using connectivity verification, a configured IWF may wait until it detects its peer before transmitting at full rate. In addition, configuration errors may be readily discovered by using the service specific field of the OAM PW packets.

In addition to one-way connectivity, OAM signaling mechanisms can be used to request and report on various PSN metrics, such as one-way delay, round trip delay, packet delay variation, etc. They may also be used for remote diagnostics, and for unsolicited reporting of potential problems (e.g., dying gasp messages).

D.2. OAM Packet Format

When using inband performance monitoring, additional packets are sent using the same PW label. These packets are identified by having their first nibble equal to 0001, and must be separated from TDM data packets before further processing of the control word.

When using a separate OAM PW, all OAM messages MUST use the PW label preconfigured to indicate OAM. All PSN layer parameters MUST remain those of the PW being monitored.

The format of an inband OAM PW message packet for UDP/IP PSNs is based on [RFC2679]. The PSN-specific layers are identical to those defined in Section 4.1 with the PW label set to the value preconfigured or assigned for PW OAM.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         PSN-specific layers  (with preconfigured PW label)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 0|L|R| M |RES| Length    |     OAM Sequence Number       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| OAM Msg Type  | OAM Msg Code  | Service specific information  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Forward PW label        |      Reverse PW label         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Source Transmit Timestamp                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Destination Receive Timestamp                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Destination Transmit Timestamp                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

L, R, and M are identical to those of the PW being tested.

Length is the length in bytes of the OAM message packet.

OAM Sequence Number (16 bits) is used to uniquely identify the message. Its value is unrelated to the sequence number of the TDMoIP data packets for the PW in question. It is incremented in query messages, and replicated without change in replies.

OAM Msg Type (8 bits) indicates the function of the message. At present the following are defined:

0 for one-way connectivity query message
8 for one-way connectivity reply message.

OAM Msg Code (8 bits) is used to carry information related to the message, and its interpretation depends on the message type. For type 0 (connectivity query) messages the following codes are defined:

0 validate connection.
1 do not validate connection

for type 8 (connectivity reply) messages the available codes are:

0 acknowledge valid query
1 invalid query (configuration mismatch).

Service specific information (16 bits) is a field that can be used to exchange configuration information between IWFs. If it is not used, this field MUST contain zero. Its interpretation depends on the payload type. At present, the following is defined for AAL1 payloads.

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Number of TSs | Number of SFs |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Number of TSs (8 bits) is the number of channels being transported, e.g., 24 for full T1.

Number of SFs (8 bits) is the number of 48-byte AAL1 PDUs per packet, e.g., 8 when packing 8 PDUs per packet.

Forward PW label (16 bits) is the PW label used for TDMoIP traffic from the source to destination IWF.

Reverse PW label (16 bits) is the PW label used for TDMoIP traffic from the destination to source IWF.

Source Transmit Timestamp (32 bits) represents the time the PSN-bound IWF transmitted the query message. This field and the following ones only appear if delay is being measured. All time units are derived from a clock of preconfigured frequency, the default being 100 microseconds.

Destination Receive Timestamp (32 bits) represents the time the destination IWF received the query message.

Destination Transmit Timestamp (32 bits) represents the time the destination IWF transmitted the reply message.

Appendix E. Capabilities, Configuration and Statistics (Informative)

Every TDMoIP IWF will support some number of physical TDM connections, certain types of PSN, and some subset of the modes defined above. The following capabilities SHOULD be able to be queried by the management system:

AAL1 capable

AAL2 capable (and AAL2 parameters, e.g., support for VAD and compression)

HDLC capable

Supported PSN types (UDP/IPv4, UDP/IPv6, L2TPv3/IPv4, L2TPv3/IPv6, MPLS, Ethernet)

OAM support (none, separate PW, VCCV) and capabilities (CV, delay measurement, etc.)

maximum packet size supported.

For every TDM PW the following parameters MUST be provisioned or signaled:

PW label (for UDP and Ethernet the label MUST be manually configured)

TDM type (E1, T1, E3, T3, fractional E1, fractional T1)

  for fractional links: number of timeslots

TDMoIP mode (AAL1, AAL2, HDLC)

for AAL1 mode:

  AAL1 type (unstructured, structured, structured with CAS)

  number of AAL1 PDUs per packet

for AAL2 mode:

  CID mapping

  creation time of full minicell (units of 125 microsecond) size of jitter buffer (in 32-bit words)

clock recovery method (local, loop-back timing, adaptive, common clock)

use of RTP (if used: frequency of common clock, PT and SSRC values).

During operation, the following statistics and impairment indications SHOULD be collected for each TDM PW, and can be queried by the management system.

average round-trip delay

packet delay variation (maximum delay - minimum delay)

number of potentially lost packets

indication of misordered packets (successfully reordered or dropped)

for AAL1 mode PWs:

indication of malformed PDUs (incorrect CRC, bad C, P or E)

indication of cells with pointer mismatch

number of seconds with jitter buffer over-run events

number of seconds with jitter buffer under-run events

for AAL2 mode PWs:

number of malformed minicells (incorrect HEC)

indication of misordered minicells (unexpected UUI)

indication of stray minicells (CID unknown, illegal UUI)

indication of mis-sized minicells (unexpected LI)

for each CID: number of seconds with jitter buffer over-run events for HDLC mode PWs:

number of discarded frames from TDM (e.g., CRC error, illegal packet size)

number of seconds with jitter buffer over-run events.

During operation, the following statistics MAY be collected for each TDM PW.

number of packets sent to PSN

number of packets received from PSN

number of seconds during which packets were received with L flag set

number of seconds during which packets were received with R flag set.

References

Normative References

[AAL1] ITU-T Recommendation I.363.1 (08/96) - B-ISDN ATM Adaptation Layer (AAL) specification: Type 1

[AAL2] ITU-T Recommendation I.363.2 (11/00) - B-ISDN ATM Adaptation Layer (AAL) specification: Type 2

[CES] ATM forum specification atm-vtoa-0078 (CES 2.0) Circuit Emulation Service Interoperability Specification Ver. 2.0

[G704] ITU-T Recommendation G.704 (10/98) - Synchronous frame structures used at 1544, 6312, 2048, 8448 and 44736 kbit/s hierarchical levels

[G751] ITU-T Recommendation G.751 (11/88) - Digital multiplex equipments operating at the third order bit rate of 34368 kbit/s and the fourth order bit rate of 139264 kbit/s and using positive justification

[G823] ITU-T Recommendation G.823 (03/00) - The control of jitter and wander within digital networks which are based on the 2048 Kbit/s hierarchy

[G824] ITU-T Recommendation G.824 (03/00) - The control of jitter and wander within digital networks which are based on the 1544 Kbit/s hierarchy

[G826] ITU-T Recommendation G.826 (12/02) - End-to-end error performance parameters and objectives for international, constant bit-rate digital paths and connections

[IEEE802.1Q] IEEE 802.1Q, IEEE Standards for Local and Metropolitan Area Networks -- Virtual Bridged Local Area Networks (2003)

[IEEE802.3] IEEE 802.3, IEEE Standard Local and Metropolitan Area Networks - Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications (2002)

[LES] ATM forum specification atm-vmoa-0145 (LES) Voice and Multimedia over ATM - Loop Emulation Service Using AAL2

[MEF8] Metro Ethernet Forum, "Implementation Agreement for the Emulation of PDH Circuits over Metro Ethernet Networks", October 2004.

[RFC768] Postel, J., "User Datagram Protocol (UDP)", STD 6, RFC 768, August 1980.

[RFC791] Postel, J., "Internet Protocol (IP)", STD 5, RFC 791, September 1981.

[RFC2119] Bradner, S., "Key Words in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.

[RFC3931] Lau, J., Townsley, M., Goyret, I., "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V., "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC4446] Martini, L., "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)", BCP 116, RFC 4446, April 2006.

[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.

[RFC4553] Vainshtein A., and Stein YJ., "Structure-Agnostic TDM over Packet (SAToP)", RFC 4553, June 2006.

[RFC4618] Martini L., Rosen E., Heron G., and Malis A., "Encapsulation Methods for Transport of PPP/High-Level Data Link Control (HDLC) over MPLS Networks", RFC 4618, September 2006.

[RFC5085] Nadeau, T., Ed., and C. Pignataro, Ed., "Pseudowire Virtual Circuit Connectivity Verification: A Control Channel for Pseudowires", RFC 5085, December 2007.

[SSCS] ITU-T Recommendation I.366.2 (11/00) - AAL type 2 service specific convergence sublayer for narrow-band services.

[Y1413] ITU-T Recommendation Y.1413 (03/04) - TDM-MPLS network interworking - User plane interworking

[Y1414] ITU-T Recommendation Y.1414 (07/04) - Voice services - MPLS network interworking.

[Y1452] ITU-T Recommendation Y.1452 (03/06) - Voice trunking over IP networks.

[Y1453] ITU-T Recommendation Y.1453 (03/06) - TDM-IP interworking - User plane interworking.

Informative References

[ISDN-PRI] ITU-T Recommendation Q.931 (05/98) - ISDN user-network interface layer 3 specification for basic call control.

[RFC792] Postel J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.

[RFC2212] Shenker, S., Partridge, C., and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, September 1997.

[RFC2330] Paxson, V., Almes, G., Mahdavi, J., Mathis M., "Framework for IP Performance Metrics", RFC 2330, May 1998.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Service", RFC 2475, December 1998.

[RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

[RFC3246] Davie, B., Charny, A., Bennet, J.C., Benson, K., Le Boudec, J., Courtney, W., Davari, S., Firoiu, V., and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", RFC 3246, March 2002.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC3985] Bryant, S. and P. Pate, "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, March 2005.

[RFC4086] Eastlake, D., 3rd, Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC4197] Riegel, M., "Requirements for Edge-to-Edge Emulation of Time Division Multiplexed (TDM) Circuits over Packet Switching Networks", RFC 4197, October 2005.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4379] Kompella, K. and Swallow, G., "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.

[RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, February 2006.

[RFC5086] Vainshtein, A., Ed., Sasson, I., Metz, E., Frost, T., and P. Pate, "Structure-Aware Time Division Multiplexed (TDM) Circuit Emulation Service over Packet Switched Network (CESoPSN)", RFC 5086, December 2007.

[SS7] ITU-T Recommendation Q.700 (03/93) - Introduction to CCITT Signalling System No. 7.

[TDM-CONTROL] Vainshtein, A. and Y(J) Stein, "Control Protocol Extensions for Setup of TDM Pseudowires in MPLS Networks", Work in Progress, November 2007.

[TRAU] GSM 08.60 (10/01) - Digital cellular telecommunications system (Phase 2+); Inband control of remote transcoders and rate adaptors for Enhanced Full Rate (EFR) and full rate traffic channels.

Authors' Addresses

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

Ronen Shashoua
RAD Data Communications
24 Raoul Wallenberg St., Bldg C
Tel Aviv 69719
ISRAEL

Phone: +972 3 645-5447
EMail: ronen_s@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

Motty (Mordechai) Anavi
RAD Data Communications
900 Corporate Drive
Mahwah, NJ 07430
USA

Phone: +1 201 529-1100 Ext. 213
EMail: motty@radusa.com

Full Copyright Statement

Intellectual Property