TDM over IP
draft-ietf-pwe3-tdmoip-02.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, or will be disclosed, and any of which I become aware will be disclosed, in accordance with RFC 3668.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on January 17, 2005.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

This document describes methods for structure-aware transport of TDM traffic over packet switched networks using pseudowires.
Table of Contents

1. Introduction ........................................ 3
2. TDMoIP Encapsulation ................................ 5
3. Encapsulation Details for Specific PSNs .......... 8
   3.1 UDP/IP ........................................... 8
   3.2 MPLS ............................................. 10
   3.3 L2TPv3 ........................................... 12
   3.4 Ethernet .......................................... 13
4. TDMoIP Payload types ............................... 15
   4.1 AAL1 Format Payload .............................. 16
   4.2 AAL2 Format Payload .............................. 17
   4.3 HDLC Format Payload .............................. 18
5. TDMoIP Defect Handling .............................. 19
6. Implementation Issues .............................. 22
   6.1 Jitter and Packet Loss ........................... 22
   6.2 Timing Recovery .................................. 23
   6.3 Quality of Service ................................ 24
7. Security Considerations ............................. 24
8. IANA Considerations ................................ 25
9. Trademarks ........................................... 25
10. References .......................................... 26
   10.1 Normative References ............................. 26
   10.2 Informative References ........................... 27
11. Acknowledgments ..................................... 28
    Authors’ Addresses ................................... 29
A. Sequence Number Processing ......................... 30
B. AAL1 Review .......................................... 32
C. AAL2 Review .......................................... 36
D. Performance Monitoring Mechanisms ................. 38
   D.1 TDMoIP Connectivity Verification ................ 38
   D.2 OAM Packet Format ................................ 39
Full Copyright Statement .............................. 42
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), integration of TDM services into a unified PSN infrastructure has become desirable. Such integration requires emulation of TDM circuits within the PSN, a function that can be carried out using pseudowires (PWs), as described in the PWE3 architecture [PWE-ARCH]. This emulation must ensure QoS and voice quality similar to those of existing TDM networks as well as preserving signaling features, as described in the TDM PW requirements [TDM-REQ].

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

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 [G.704] of length 193 and 256 bits, respectively. T3 and E3 frames [G.704,G.751] 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.

TDM circuits are often used to transport multiplexed 64 kbps channels. A frame of a channelized T1 carries 24 byte-sized channels, while an E1 frame consists of 32 channels. By concatenation of consecutive T1 or E1 frames we can build higher level structures called superframes or multiframes.

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 [SAToP] is a structure-agnostic protocol for transporting TDM over 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, and 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.

When it is required or desirable to explicitly safeguard TDM structure during transport over the PSN, structure-aware TDM transport must be employed. 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 might not transport structure overhead across the PSN; in particular, the FAS MAY be stripped by the PSN-bound GW and MUST be regenerated by the TDM-bound GW. However, structure overhead MAY be transported over the PSN, since in addition to FAS it can contain maintenance information.

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 meaningful for channelized TDM; the PSN-bound GW extracts and buffers the individual channels, and the original structure is reassembled from the received constituents by the TDM-bound GW.

All three methods of TDM structure preservation have their advantages. Structure-locking is described in [CESoPSN], while the present document specifies both structure-indication (see Section 4.1) and structure-reassembly (see Section 4.2) 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). This will be discussed further in Section 5.

Despite its name, the TDMoIP(R) protocol herein described tolerates several types of PSN, including UDP over IPv4 or IPv6, MPLS, L2TPv3 over IP, or pure Ethernet. Implementation specifics for particular PSNs are discussed in Section 3. Although the protocol should be more generally called TDMoPW and its specific implementations TDMoIP, TDMoMPLS, etc. We will use the nomenclature TDMoIP for reasons of consistency with earlier usage.

2. TDMoIP Encapsulation

The overall format of TDMoIP packets is shown in the following figure.

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

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 GW 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 gateway may simultaneously support multiple TDM PWs, and the TDMoIP gateway 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. In general the PW labels of these PWs will take different values.
In addition to the aforementioned headers, an OPTIONAL 12-byte RTP header may appear in order to provide a mechanism for explicit transfer of timing information in the packet. If RTP is used, the fixed RTP header described in [RTP], MUST immediately precede the control word for UDP/IP, and MUST immediately follow it for all other cases. The P (padding), X (header extension), CC (CSRC count), and M (marker) fields in the RTP header MUST be set to zero, and the 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 [RTP]; the clock frequency should be an integer multiple of 8 kHz, and MUST be chosen to enable timing recovery that conforms with the appropriate standards (see Section 6.2). When the TDMoIP gateways have sufficiently accurate local clocks or can derive sufficiently accurate timing without explicit timestamps, the RTP header SHOULD be omitted.

The 32-bit control word MUST appear in every TDMoIP packet. Its format is depicted in the following figure.

```
  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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FORMID |L|R| M |RES|  Length   |         Sequence Number       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

FORMID Format identifier (4 bits) is an OPTIONAL field that MAY be used to specify the payload format. When it is not used it MUST be set to zero by the PSN-bound GW and ignored by the TDM-bound GW. It SHOULD NOT be used when PSN forwarding mechanisms identify PWs based on the initial four bits of their packets being zero. The following values are presently defined (none of which alias an IP packet):

- 1100 AAL1 unstructured
- 1101 AAL1 structured
- 1110 AAL1 structured with CAS
- 1001 AAL2
- 1111 HDLC mode

The payload format for each of these cases will be described in Section 4.
L Local Failure (1 bit) The L flag being set indicates that the PSN-bound GW 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 GW 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 5.

R Remote Failure (1 bit) The R flag being set indicates that the PSN-bound GW 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. In the trail extended OAM scenario the R flag MUST be set when the GW detects RDI, and the GW SHOULD generate TDM RDI upon receipt of an R flag indication. Use of the R flag is further explained in Section 5.

Defect Modifier (2 bits) Use of the M field is optional, and when used 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 (indicating 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 gateway.
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 5.

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 [PWE-ARCH], 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 PW. Pseudocode for a sequence number processing algorithm that could be used by a TDM-bound GW is provided in Appendix A.

In order to form the TDMoIP payload, the PSN-bound GW 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 4), and the resulting adapted payload is placed into the packet.

3. 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 identical to that specified in [Y1413].

3.1 UDP/IP

The UDP/IP header as described in [UDP] and [IP] is prefixed to the TDMoIP data. The TDMoIP packet structure is as follows:
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. for IPv4 IPVER=4.

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 Flags=010 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 11 hex = 17 dec 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.

VER (3 bits) is the TDMoIP version number. The original version (VER=000) was experimental and should no longer be used. Presently VER=001 when RTP is not used, and VER=011 when RTP is used.

PW label (13 bits) This field is usually dedicated to the Source Port Number, but here identifies the unique data stream emanating from a given TDM circuit and sharing a common destination. This nonstandard use of a UDP port number is similar to RTP/RTCP’s use of port numbers to uniquely identify sessions, and to allocation of arbitrary UDP port numbers for VoIP sessions. Placing the PW label in the UDP header rather than the application area facilitates implementation. The value 0 is reserved; a preconfigured (default 1FFF hex = 8191 dec) label value is used for PW-layer OAM messages carried in a separate PW (see Appendix D); other PW labels in the range 1-8191 are available for use.

Destination Port Number (16 bits) MUST be set to 0x085E (2142), the user port number that has been assigned to TDMoIP by IANA.

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.

3.2 MPLS

ITU-T recommendation Y.1413 [Y1413] describes structure-agnostic and structure-aware mechanisms for transporting TDM over MPLS networks. 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.
The MPLS header as described in [MPLS] is prefixed to the control word and TDM payload. The packet 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Tunnel Label               | EXP |S| TTL           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              PW label                 | EXP |S| TTL           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FORMID |L|R| M |RES|  Length   |         Sequence Number       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
opt|RTV|P|X| CC | M| PT | RTP Sequence Number |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
opt|                          Timestamp                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
opt|                          SSRC identifier                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Adapted Payload                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 [MPLS].

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. Should be set to 2 for the PW label.

PW Label (20 bits) The value 0 is reserved; a preconfigured (default FFFFF hex = 1048575 dec) label value is used for PW-layer OAM messages carried in a separate PW (see Appendix D); other PW labels are available for use.
3.3 L2TPv3

The L2TPv3 header defined in [L2TPv3] is prefixed to the TDMoIP data. The packet 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 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)                                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| FORMID |L|R|  RES  |  Length   |         Sequence Number       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|RTV|P|X|  CC   |M|     PT      |      RTP Sequence Number      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Timestamp                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC identifier                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                Adapted Payload                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

Protocol the IP protocol field must be set to 73 hex = 115 dec, 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; a preconfigured (default FFFFFFFF hex) label value is used for PW-layer OAM messages carried in a separate PW (see Appendix D); other PW labels are available for use.
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.

3.4 Ethernet

The TDMoIP packet described in the previous subsections will frequently be further encapsulated in an Ethernet frame by prefixing the Ethernet preamble, 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, zero 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 1460 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. In this case an MPLS-style label will always be present, but if VLAN tags are sufficient to identify the PW the MPLS label MUST be set to one. The Ethertype SHOULD be set to the value allocated for CESoETH, but MAY be set to the Ethertype of MPLS. The packet structure is as follows:
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) a 8100 hex in this position indicates that optional VLAN tagging according to [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 4193 values are valid VLAN identifiers.

Ethertype (16 bits) is the protocol identifier, as allocated by the IEEE.

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

4. TDMoIP Payload types

As discussed at the end of Section 2, 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.

Since native TDM is always constant bit-rate, why is a variable rate adaptation needed? For unstructured TDM, or structured but unchannelized TDM, of structured channelized TDM with all channels active all the time, there is indeed no need. In such cases TDMoIP uses structure-indication to emulate the native TDM circuit, utilizing an adaptation known as circuit emulation. However, individual "local loops" are frequently "on-hook" and thus inactive, and bandwidth may be conserved by transporting only channels corresponding to active loops. This results in variable rate real-time traffic, for which TDMoIP uses structure-reassembly to emulate the individual loops, utilizing an adaptation 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 GW 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 HDLC-based CCS signaling 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.

4.1 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.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>control word</td>
<td>AAL1 PDU</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
</tr>
</tbody>
</table>

Single AAL1 PDU per TDMoIP packet

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>control word</td>
<td>AAL1 PDU</td>
<td>---+ AAL1 PDU</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
</tbody>
</table>

Multiple AAL1 PDUs per TDMoIP packet
In AAL1 mode the TDMoIP payload consists of between one and thirty 48-byte "AAL1 PDUs". The number of PDUs must be pre-configured and typically chosen according to latency and bandwidth constraints. Using a single PDU reduces latency to a minimum, but incurs the highest overhead, while using, for example, eight PDUs reduces the overhead percentage while increasing the latency by a factor of eight. 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 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.

4.2 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 by observation of the TDM signaling channel activity status may be determined. 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.
In AAL2 mode the TDMoIP payload consists of one or more variable-length "AAL2 PDUs". 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.

[PWE-ARCH] 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, CID encoding according to [AAL2], encoding formats defined in [SSCS], and transport of CAS and CCS signaling as described in [LES] may all be used. The overlap functionality and AAL-CU timer and related functionalities may not be required, and the STF field is NOT used. Computation of error detection codes, namely the 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 MUST be set to zero.

4.3 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 PSN-bound GW monitors flags until a frame is detected. The contents of the frame are collected and the 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 an TDMoIP- HDLC frame is received its FCS is recalculated, and the original HDLC frame reconstituted.

5. 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 transports all such indications; however, structure-aware mechanisms for which the PSN-bound GW may remove TDM structure overhead carrying defect indications, will require explicit signaling of TDM defect conditions.

We saw in Section 2 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|-<-|     |-<-|  GW1 |-<-|     |-<-|  GW2 |-<-|     |-<-|ES1|
+---+   +-----+   +------+   +-----+   +------+   +-----+   +---+
```

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 gateway 1 (GW1), through the PSN, through TDMoIP gateway 2 (GW2), 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 GW 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 GW1). If a break occurs in the ultimate link, the GW itself will detect the loss of signal. In either case, GW1 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, GW1 serves as a trail termination function for TDM network 1, and thus when GW1 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 GW1 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 GW1 to fill the payload with the appropriate amount of AIS (usually all-ones) data. If the AIS has been generated before the GW this can be accomplished by copying the received TDM data; if the penultimate TDM link fails and the GW needs to generate the AIS itself. Alternatively, structure-aware transport of channelized TDM MAY 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 GW into a single TDM circuit. The third possibility is to suppress the payload altogether. Finally, if GW1 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 (ÅMÅ) that the TDM data is corrupted, but potentially recoverable.
When GW2 receives a local defect indication without ÂMÂ-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, GW2 receiving the ÂLÂ indication 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 GW2 treats the payload contents as regular TDM data. If the payload has been suppressed, trunk conditioning and not AIS MUST be generated by GW2.

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, GW2 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 GW1. GW1, 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 GW1 sends packets toward GW2, but these are not received. GW2 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 GW2 detects RDI it MUST raise the "R" flag indication. When GW1 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 4.3, 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 GWs commence sending AIS (and not trunk conditioning) to their respective TDM networks. The GWs then engage in connectivity testing using TDMoIP OAM as described in Appendix D until connectivity is restored.

6. Implementation Issues

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

6.1 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 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 packets in the jitter buffer and MUST take appropriate action when anomalies are detected. When missing packet(s) are detected the mechanism MUST output filler packet(s) in order to retain TDM timing. Packets with incorrect serial numbers or other detectable header errors MUST be discarded. Packets arriving in incorrect order SHOULD be swapped. Processing of filler packets 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 packets may be sufficient to maintain the TDM timing, for voice traffic it may lead to gaps or artifacts that result in choppy, annoying or even unintelligible speech, see [TDM-PLC]. An implementation MAY blindly insert a preconfigured constant value in place of any lost speech samples, and this value SHOULD be chosen to minimize the perceptual effect.
Alternatively one MAY replay the previously received packet. Since a TDMoIP packet is usually declared lost following the reception of the next packet, when computational resources are available, implementations SHOULD conceal the packet loss event by properly estimating the missing speech sample values.

6.2 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 10E-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 three methods of overcoming this difficulty. In the first method timing information is provided by some means independent of the PSN. This timing may be provided to the TDM end system or to the GWs. In a second method a common clock is assumed available to both gateways, and the relationship between the TDM source clock and this clock is encoded in the packet. This encoding may be take the form of RTP timestamps or may utilizing the 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 [TDM-REQ] 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 GW, 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 GW 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 required.

6.3 Quality of Service

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.

If the PSN is Diffserv-enabled then an EF-PHB (expedited forwarding) class-based PDB SHOULD be used, in order to provide a low latency and minimal jitter service. It is suggested that the transport LSP be somewhat overprovisioned.

If the MPLS network is Intserv enabled, then GS (Guaranteed Service) with the appropriate bandwidth reservation SHOULD be used in order to provide a bandwidth BW guarantee equal or greater than that of the aggregate TDM traffic. The delay introduced by the MPLS network SHOULD be measured prior to traffic flow, to ensure its compliance with latency requirements.

7. 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.

TDMoIP does not provide protection against malicious users utilizing snooping or packet injection during setup or operation. However, random initialization of sequence numbers makes known-plaintext attacks on link encryption methods more difficult.

PW labels SHOULD be selected in an unpredictable manner rather than sequentially or otherwise in order to deter session hijacking. When using L2TPv3, randomly selected cookies MAY be used to validate circuit origin. Sequence numbers SHOULD be randomly initialized in order to increase the difficulty of decrypting based on packet headers.
8. IANA Considerations

When used with UDP/IP the destination port number MUST be set to 0x085E (2142), the user port number which has been assigned by IANA to TDMoIP.

9. Trademarks

TDMoIP is a registered trademark of RAD Data Communications. RAD Data Communications grants the IETF a perpetual license to reproduce this trademark solely in connection with the reproduction, distribution or publication of this contribution and derivative works thereof, in accordance with RFC 3667.
10. References

10.1 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

[CONNECT] RFC 2678 IPPM Metrics for Measuring Connectivity

[DELAY] RFC 2679 A One-way Delay Metric for IPPM

[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

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


[IPPM] RFC 2330 Framework for IP Performance Metrics


[LES] ATM forum specification atm-vmoa-0145 (LES) Voice and Multimedia over ATM - Loop Emulation Service Using AAL2
[L2TPv3] draft-ietf-l2tpext-12tp-base-10.txt (08/03) Layer Two Tunneling Protocol (L2TPv3), J. Lau et al., work in progress

[MPLS] RFC 3032 MPLS Label Stack encoding

[RTP] RFC 3550 RTP: Transport Protocol for Real-Time Applications

[SAToP] draft-ietf-pwe3-satop-00.txt (09/03) Structure-Agnostic TDM over Packet (SAToP), A. Vainshtein and Y. Stein, work in progress

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

[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

[UDP] RFC 768 (STD0006) User Datagram Protocol (UDP)

[VCCV] draft-ietf-pwe3-vccv-03.txt (06/04) Pseudo Wire Virtual Circuit Connectivity Verification, T. Nadeau and R. Aggarwal, work in progress

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

10.2 Informative References

[CESoPSN] draft-ietf-cesopsn-00.txt (01/04), TDM Circuit Emulation Service over Packet Switched Network, A. Vainshtein et al, work in progress

[ICMP] RFC 792 Internet Control Message Protocol. (09/81)

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

[PWE3-ARCH] draft-ietf-pwe3-arch-07.txt (3/04), PWE3 Architecture, Stewart Bryant et al, work in progress

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

[TDM-PLC] draft-stein-pwe3-tdm-packetloss-01.txt (10/03), The Effect of Packet Loss on Voice Quality for TDM over Pseudowires, Y(J) Stein and I. Druker, work in progress
11. Acknowledgments

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

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 having previously having been assumed lost, the TDM-bound GW 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 sequence numbers (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 of packet recoverable (Note 1).

Note 1: this is only required for the optional re-ordering.
Note 2: this number is always in the range 
\(-2^{15} \ldots +2^{15} - 1\).
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
     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
         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
         calculate L = ( (played-received) mod 2^16 ) - 2^15
         if 0 < L <= R
            replace packet previously marked as lost with actual data
         else
            discard packet
Note: by choosing R=0 we always discard the late packet
Appendix B. AAL1 Review

The first byte of the 48-byte AAL1 PDU always contains an error-protected three-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 this byte can only take on eight different values, and 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 namely 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 5/8 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 taken;

<table>
<thead>
<tr>
<th>byte</th>
<th>1 2 3 4 5 6 7 8 9 10 --- 41 42 43 44 45 46 47</th>
</tr>
</thead>
<tbody>
<tr>
<td>channel</td>
<td>2 3 5 7 11 2 3 5 7 11 --- 2 3 5 7 11 2 3</td>
</tr>
</tbody>
</table>

the next PDU commences where the present PDU left off

<table>
<thead>
<tr>
<th>byte</th>
<th>1 2 3 4 5 6 7 8 9 10 --- 41 42 43 44 45 46 47</th>
</tr>
</thead>
<tbody>
<tr>
<td>channel</td>
<td>5 7 11 2 3 5 7 11 2 3 --- 5 7 11 2 3 5 7</td>
</tr>
</tbody>
</table>

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, 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 number. Unlike most AAL1 PDUs that contain 47 TDM bytes, PDUs that contain a pointer (P-format PDUs) have the following format.
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 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

\[
\begin{array}{cccccccccccccccccccccccc}
2 & 3 & 5 & 2 & 3 & 5 & 2 & 3 & 5 & 2 & 3 & 5 & 2 & 3 & 5 & 2 & 3 & 5 & 2 & 3 & 5 & 2 & 3 & 5 \\
2 & 3 & 5 & 2 & 3 & 5 & 2 & 3 & 5 & 2 & 3 & 5 & 2 & 3 & 5 & 2 & 3 & 5 & 2 & 3 & 5 & 2 & 3 & 5
\end{array}
\]

[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

The basic AAL2 PDU is:

```
+-------+-------+-------+
| CID   | LI     | UUI    |
+-------+-------+-------+
|       |   6   |   5   |
+-------+-------+-------+
|   8   |   6   |   5   |
| ALREADY ALLOCATED | ALREADY ALLOCATED | ALREADY ALLOCATED |
|        | ALREADY ALLOCATED | ALREADY ALLOCATED |
|        | ALREADY ALLOCATED | ALREADY ALLOCATED |
```

CID (8 bits) channel identifier is an identifier that must be unique for the PW. The values below 8 are reserved and so there are 248 possible channels. The mapping of CID values to channels is beyond the scope of the TDMoIP protocol and must be configured manually or via network management.

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 a type 3 packet (in the notation of [AAL2]) in order to ensure error protection. The signaling is sent with the same CID as the corresponding voice channel. Signaling is sent whenever the state of the ABCD bits changes, and is sent with triple redundancy, i.e. sent three times spaced 5 milliseconds apart. In addition, the entire set of the signaling bits is sent periodically to ensure reliability.

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

Stein, et al.
Expires January 17, 2005
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 for periodic refresh of the CAS information.

Timestamp (14 bits) The timestamp is 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

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 [IPPM], and its suitability for carrying TDM services.

TDMoIP GWs 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 [ICMP] 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. One, internal to the PW and based on inband VCCV [VCCV], and another that runs in a separate PW. The latter is particularly efficient when a large number of TDM PWs are placed in a single PSN tunnel.

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 GW could send a continuous stream of packets towards the destination without knowing whether the TDM-bound GW 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 GW is first brought up, when the TDM-bound GW fails or is disconnected from the PSN, or the PW is broken. After an aging time the destination gateway disappears from the routing tables, 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, as defined in [CONNECT].

Connectivity is tested by periodically sending OAM messages from the source GW to the destination GW, 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 GW is ready to receive packets before starting to send them. Since TDMoIP gateways 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 gateway 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 FORMID=0001, and must be separated from TDM data packets before sequence number processing.

When using a separate OAM PW, all OAM messages MUST use the PW label preconfigured to indicate OAM (the default value is the highest label available). All PSN layer parameters (for example, MPLS label, IP addresses, TOS, EXP bits, and VLAN ID) MUST remain those of the PW being investigated.

The format of an OAM PW message packet is depicted in the following figure. Note that PSN-specific layers are identical to those used to carry the TDMoIP data, with the exception that the PW label MUST be set to the preconfigured value instead of the usual PW identifier.
FORMID, 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 gateways. 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 gateway.

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

Source Transmit Timestamp (32 bits) represents the time the PSN-bound GW 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 gateway received the query message.

Destination Transmit Timestamp (32 bits) represents the time the destination gateway transmitted the reply message.
Full Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

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

Acknowledgement

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