In-depth Comparison of TDMoIP and CESoPSN

Yaakov (J) Stein

June 2003

Several methods of transporting TDM over IP or MPLS networks have been proposed over the past few years, but most of the debate has been between two camps. The TDMoIP camp champions *draft-anavi* while the CESoPSN camp supports *draft-vainshtein*. This document compares the two technologies.

1 Design Philosophies

Although not strictly relevant to a comparison of the two protocols as they have developed, some knowledge of the original design goals of the two development teams involved will assist the reader in understanding the divergent decisions and trade-offs made by designers of each technology.

TDMoIP was designed by RAD Data Communications, a company whose major expertise is in access networks, and was thus constructed to function even as an access method for enterprises or a means for campus users to converge networks. The RAD development team did not believe that incumbent carriers would immediately abandon their highly reliable and fully functional TDM networks in favor of IP or even MPLS ones. They *did* believe that enterprises and alternative service providers would strive to cut costs and to maximize the return on their investment in broadband data networks. In fact, the great majority of the 12,000 TDMoIP ports deployed to date are used by enterprises, campuses, and alternative service providers.

From the practical point of view, this viewpoint led to the development of devices that handled a relatively small number of ports, from a single T1 or E1 trunk, up to 16 trunks. From the protocol point of view, this viewpoint led to a strong emphasis on proper handling of fractional T1 or E1, necessitating full exploitation of TDM structure. Since enterprise customers do not usually have high quality clock sources, TDMoIP designers went to great lengths in designing adaptive clock recovery techniques that enable conformance with international standards even over degraded networks. Becuase enterprise networks are often unreliable and overloaded, TDMoIP designers considered the careful handling of packet loss to be critical. Since enterprise customers often have simple PBXs, and some users even wish to directly connect analog phones, RAD designers were concerned with robust and efficient transport of TDM signaling. Finally, RAD's installed base of circuit emulation systems operating over ATM networks led the TDMoIP design team to prefer protocols that easily interwork with ATM CES.

It is not a secret that Axerra, the company that developed CESoPSN, was founded by R&D personnel who left RAD for this purpose. Their vision was to elevate TDMoIP applications to the carrier level, providing a solution for the 'carrier's carrier'. This also explains their cooperation with proponents of SONET/SDH over packet networks.

Since the CESoPSN team focused on a different client, if is not surprising that they de-emphasized various aspects important in TDMoIP's design, while emphasizing others. Carriers require very large systems handling large numbers of trunks, constraining processing power per input trunk to be kept minimal. Furthermore, carriers generally deal in full trunks, so TDM structure is a secondary concern. On the other hand, Axerra expected their client to be in possession of a 'well engineered network', i.e. one for which packet loss and packet delay variation (PDV) are negligible. Furthermore, their clients would usually have an accurate local clock source, and so complex (and expensive) adaptive clock recovery schemes would not be needed. Also, carrier class clients in most of the world have long since abandoned CAS signaling in favor of CCS, so this responsibility too could be alleviated. Finally, having been formed at the height of the high-tech revolution, Axerra did not see any reason to require backwards compatibility or simple interworking with existing circuit emulation services.

In the following sections we will see that these design principles permeate every aspect of the two protocols, dictating quite different approaches and solutions to the various issues. The only aspect of the debate not readily understood from the design philosophies is the nomenclature. RAD, reusing technology originally developed for ATM CES chose the name TDMoIP, while Axerra, shunning ATM CES and emphasizing TDM frames trademarked the name CESoIP. The true reason for this bizarre circumstance is historical; RAD originally chose TDMoIP in order to conceal the connection with ATM, and Axerra had to resign itself to CESoIP, as TDMoIP had already been taken.

2 Encapsulation

The most visible difference between the two camps is the protocol used for encapsulation of the TDM data into packets. Both protocols chop the continuous stream of TDM data into segments, add protocol specific headers and afterward the required UDP/IP headers or MPLS label stack, but there the similarity ends. CESoPSN's payload is relatively unaltered TDM data, but requires specific payload sizes. TDMoIP has several payload modes which differ in the adaptation performed on the TDM segment before adding the headers.

2.1 Control word

Yet the first difference between draft-vainshtein and draft-anavi has nothing to do with the TDM payload format, rather it is in the control word. The control word is protocol-specific information added to the payload, and includes, at minimum, a sequence number and various flags. Early on TDMoIP the control word used by the *Martini drafts*, which was later adopted as the standard control word by the IETF PWE3 working group. CESoPSN utilizes a variant of the control word invented for the draft-malis, a protocol for transporting SONET over MPLS.

The draft-malis control word includes a 13 bit pointer, needed for SONET/SDH payloads, but unused for CESoPSN. In order to accommodate this SONET-specific requirement it omits the length indicator (which is not needed for SONET/SDH payloads, which are large) and uses a smaller sequence number. The CESoPSN mechanism requires no pointer in its header, and so the reason for utilizing the non-standard control word is unclear.

The control word used in the Martini drafts always has its first four bits equal to zero. This fact is exploited by MPLS switches to distinguish between pseudowires (which will have the first nibble after the label stack set equal to zero) and IP packets (which have the IP version number 4 or 6 in this position). Unfortunately, the draft-malis control word has various flags in this place, and is hence incompatible with MPLS switches that use this "MPLS PID". A kludge has been suggested whereby an additional long-word of all zeros is inserted between the bottom of the MPLS label stack and the control word, a mechanism that adds un-necessary overhead to salvage the situation.

The next difference in the control words used by TDMoIP and CESoPSN is their use of flags to indicate network problems. The CESoPSN draft defines five flags, of which one merely indicates use of an extended (64 bit) control word, which, however, is never used in CESoPSN. Another bit designates defects in reception from the packet network, while the three remaining bits map to specific TDM network faults. Of the eight possible combinations only four are legal.

The TDMoIP control word includes only two flags, representing a higher level abstraction of network faults. The L (local) bit is used to indicate a fault in the TDM network attached to the TDMoIP gateway, while activity of the R (remote) bit signifies that the gateway is not receiving packets fromthe packet network. These flags are general purpose interworking indicators and do not contain TDM-specific alarm information. Such detailed information is not required at the interworking function level, and is, in any case, carried in the TDM payload.

The control word's sequence number is used to detect lost or misordered packets. TDMoIP utilizes a 16 bit sequence number, while CESoPSN uses a smaller 14 bit one. Both of these are more than sufficient for TDM transport applications.

2.2 RTP

CESoPSN mandates the use of RTP, the IETF's real time protocol. While draft-anavi allows the inclusion of the RTP header, TDMoIP technology does not exploit it in any way. RTP is universally used for VoIP applications, and provides an excellent solution to two of VoIP's problems, namely enabling constant rate playback of non-constant bit-rate (e.g. compressed voice) signals, and absolute time synchronization of distinct real-time streams (e.g. for combining multiple speakers in conference calls). Neither of these features is required for TDM transport over packet switched networks.

The RTP header is large (at least 12 bytes) and so adds significant overhead to the packet. Although the header contains other fields, CESoPSN exploits only the four-byte timestamp, and the two-byte sequence number. The sequence number is redundant as there already is a sequence number in the control word (CESoPSN suggests equating these). CESoPSN proponents suggest that the RTP timestamp can be used for TDM clock recovery. Regrettably, it can be shown mathematically (see 'The Insufficiency of RTP for TDM Clock Recovery' by the present author) that use of the RTP timestamp does not significantly contribute to the reconstitution of a clock signal that conforms to international standards for jitter and wander allowable on TDM circuits. This may explain the consistent objection of CESoPSN proponents to specification of a hard requirement for such conformance.

More precisely, what can be shown is that of all methods of utilizing the RTP timestamp, the one that minimizes reconstructed wander does not use the timestamp at all. This conclusion is not surprising for several reasons. First, RTP is a layer four protocol, while clock recovery is a physical layer problem. Mechanisms functioning at so high a layer can not be accurate enough to conform to TDM requirements. Second, if the RTP timestamp is derived from the same clock that drives the TDM source, then the timestamp will be exactly a linear function of the sequence number, and thus redundant. Only if the timestamp is derived from an accurate clock available to both source and destination is there any hope of deriving any substantive information from it, but even for this rare case the quantization error of the RTP timestamp limits its usefulness. On the other hand, straightforward extensions of well-known DSP techniques have been shown to enable compliant clock recovery without need for any source timestamp. For this reason TDMoIP recommends omitting RTP.

2.3 Payload formats

A CESoPSN payload always corresponds to 125 μsec of TDM data, or some multiple thereof. When carrying structured TDM, the payload is forced to commence on a TDM frame boundary, while for

unstructured TDM the payload need not, and no byte or bit alignment is implied.
A problem arises when transporting unstructured T1 streams, which are of 193 bits in length. Since the payload must contain a whole number of bytes, CESoPSN requires padding the T1 frame, and recommends the use of the "G.802 Annex B" technique of mapping a T1 frame into an E1 one. In more recent draft-vainshtein revisions, CESoPSN allows the concatenation of eight consecutive T1 frames into a payload of 193 bytes so that padding is no longer required, but at the cost of a millisecond of delay.
The simplest mode of draft-anavi is the "raw mode" where the TDM segment is taken as-is to be the packet payload. Note that in this mode no bit or byte alignment is ever implied, and the payload is simply a constant length segment of TDM bits. Hence an unstructured T1 packet may contain a noninteger number of TDM octets.
The next two modes of TDMoIP are borrowed from ATM technology. Before discussing them one should note that (contrary to statements made by TDMoIP opponents) these modes do *not* put the TDM data into ATM cells. They use only the ATM Adaptation Layer (AAL) and never add the five bytes of "cell tax".
The 'AAL1 (CBR) mode' segments the TDM stream into short 48 byte cells, adds sequence numbers and structure pointers and then places some number of these cells into the packet. The structure pointers point to the commencement of the next TDM structure, where this structure may be a single TDM frame or a multiframe (see below) depending on the incoming traffic.
The 'AAL2 (VBR) mode' may be used when the incoming traffic is expected to be "bursty", i.e. not continuously full-bandwidth. It disassembles and buffers the TDM timeslots, and only sends those that are active. The activity detection may be based on TDM signaling (e.g. off-hook vs. on-hook state), or may include voice activity detection to further increase efficiency.
The final mode of draft-anavi is specifically designed to increase the efficiency of CCS signaling encoded in HDLC. In this mode active HDLC frames are detected and transported, while idle states are discarded.
The fact that TDMoIP contains several modes should be considered a major feature rather than a needless complication. While a single mode could have been defined, the alternative modes enhance functionality and data-rate efficiency. For example, we shall see below that AAL2 not only conserves data bandwidth, but can be used as a network congestion avoidance mechanism.

3 Delay vs. Bandwidth Trade-off

CESoPSN's use of integer multiples of 125 μsec of TDM data, imposes specific payload sizes. For example, for full E1 the minimum payload is 32 bytes, and all possible payloads are multiples of 32 bytes. For T3 the minimum payload is 699 bytes (which is 125 μsec of data at the T3 rate), and only a double size payload (1398 bytes) is also possible (assuming an MTU of 1500 bytes).
TDMoIP considers the payload size to be a parameter that must be carefully tuned for optimal performance, and hence intentionally breaks all connection between the TDM type and payload size. TDMoIP payloads are always multiples of 48 bytes, irrespective of whether the original TDM was T1, E1, T3, or E3.
TDMoIP implementors consider the payload size to be critically important for its optimal operation. Since the encapsulation adds a constant amount of overhead, longer packets are more bandwidth efficient. However, longer packets add more buffering latency. On the other hand short packets reduce latency, but decrease efficiency and increase the number of packets per second that need to be processed.

To better understand the trade-offs involved, a few examples are in order. In a metro-ethernet network it is desired to keep round-trip delay minimal, in order to avoid the need for echo cancellation; but in such networks bandwidth is relatively inexpensive, which means that small packet sizes are acceptable. At the other end of the spectrum, in multiple access wireless applications, there is a strong requirement to completely fill up the packet (since the user may have to wait a long time before receiving permission to transmit again), while the delay requirements are less stringent. This requires approaching the network MTU as closely as possible.
The designers of TDMoIP decided that granularity equal to the payload size of standard AAL1, namely 48 bytes, was a good compromise choice. Finer granularities do not significantly increase the ability to fill large packets, but sub-48 byte payloads *do* suffer from terribly low efficiencies and require high packet processing (packets per second) rates. Coarser granularities make it more difficult to fill large packets, and the smallest possible packet may not be small enough for some applications.
Certainly other values would have been possible (e.g. 32 or 64 bytes), but then entirely new and untested mechanisms would have to be invented to accommodate these new sizes. In particular, the extremely efficient pointer mechanism of structured AAL1 is designed specifically for 48 byte payloads, and any other size would require redesigning all of the related algorithms. In addition, only by using the same size and mechanisms as standard AAL1 is it possible to simplify interworking with existing ATM-based circuit emulation systems.

3.1 Bandwidth Efficiency

Due to the fundamental differences between them, direct comparison of the data rates of TDMoIP and CESoPSN is dependent on various factors. It is important to compare situations with similar delay, as increasing latency will increase efficiency for both methods.
Let's start with the simple case wherein we wish to transport a full unstructured T1 over an IP network with relatively low delay. Since CESoPSN must pad the payload, each TDM frame requires 25 bytes; since TDMoIP packs 47 TDM bytes into 48 payload bytes, each packet will contain 47N payload bytes. Hence a CESoPSN packet containing two TDM frames (2*193=386 bits) is comparable in delay to a TDMoIP packet with a single cells (47*8=376 bits).
Both TDMoIP and CESoPSN will prepend twenty bytes of IP header, eight bytes of UDP header, and four bytes of control word. CESoPSN adds a further twelve bytes of RTP header. TDMoIP will then insert 48 bytes of payload for a grand total of $20 + 8 + 4 + 48 = 80$ bytes $= 640$ bits, of which $47 * 8 = 376$ bits are traffic, for an efficiency of $376/640 = 58.75\%$. CESoPSN inserts $2 * 25 = 50$ bytes of payload, making $20 + 8 + 12 + 4 + 50 = 94$ bytes $= 752$ bits, of which 386 bits are true traffic, for an efficiency of $386/752 = 51.33\%$. So in this case TDMoIP beats CESoPSN by over 7%.
In fact, while TDMoIP adds 1 byte of overhead every 47 (about 2%), CESoPSN adds 7 bits every 193 (3.6%) in addition to a constant overhead of 12 bytes, so TDMoIP is always more efficient for T1 links. The situation for MPLS is similar, but CESoPSN fares even worse due to its having to add an extra four 'kludge' bytes of zeros before the control word.
In figure 1 we depict the efficiency percentage as a function of the delay in milliseconds for a full T1, using draft-anavi's raw mode (solid lines), TDMoIP AAL1 mode (dotted lines) and CESoPSN (dashed lines) for both IP and MLPS networks.
For E1 links, where the byte padding is not required, the difference will be smaller, but TDMoIP will still have the advantage for small packet sizes due to not requiring the 12 byte RTP overhead. At some point (above 2.3 milliseconds for IP networks and 2.8 milliseconds for MPLS networks) CESoPSN overtakes TDMoIP, but even for giant packets (close to the 1500 byte limit) it never surpasses it by more than about a percent.

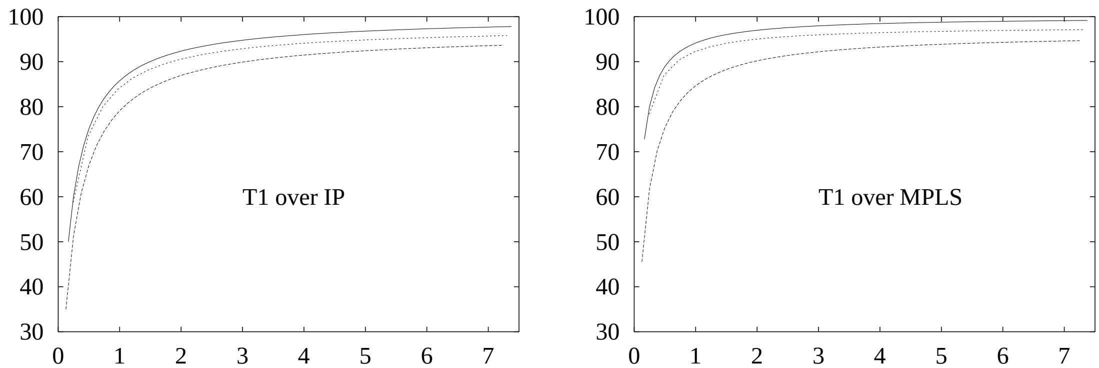

Figure 1: Efficiency of raw mode, AAL1 mode and CESoPSN for full T1 over IP and MPLS networks.

In figure 2 we depict the efficiency percentage as a function of the delay in milliseconds for a full E1, using raw mode (solid lines), AAL1 mode (dotted lines) and CESoPSN (dashed lines) for IP and MLPS networks.

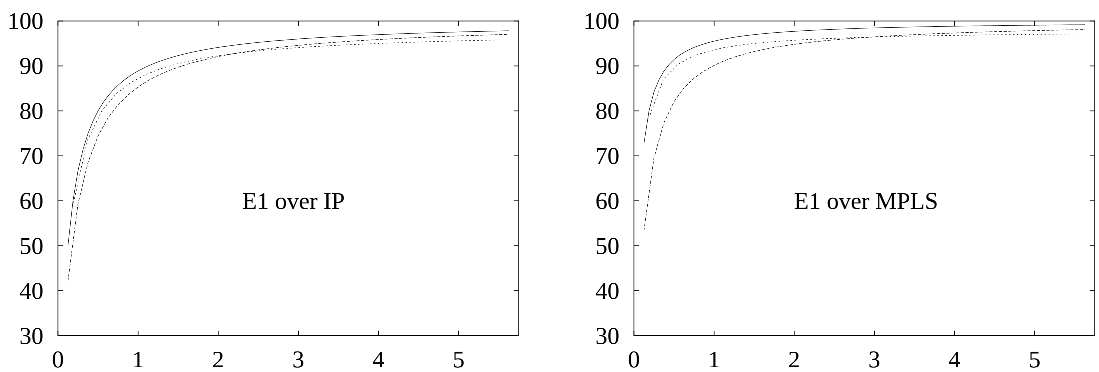

Figure 2: Efficiency of raw mode, AAL1 mode, and CESoPSN for full E1 over IP and MPLS networks.

3.2 Delay

In draft-vainshtein we find the following statement: *Fixed end-to-end delay: CESoPSN implementations SHOULD provide the same end-to-end delay between any given pair of PEs regardless of the bit-rate of the emulated service.* In addition, in the editorial team that drafted the TDM requirements document for the PWE3 WG, CESoPSN proponents spoke at length of the absolute requirement that any two links carrying different TDM data should have the same latency.

Although a softened wording of this 'requirement' did eventually make its way into the TDM requirements document, no reasonable explanation of this requirement was ever proffered. It seems that CESoPSN proponents decided to take one of CESoPSN least savory features, namely its packet size inflexibility, and to turn this into a 'requirement' that they obey while TDMoIP does not.
The only true requirement regarding delay is the required flexibility, and the ability to trade-off delay for efficiency. As we have seen, by breaking the link between TDM frames and packet sizes, TDMoIP allows a flexible trade-off.

4 Computational Complexity (MIPS)

CESoPSN proponents are fond of reminding us that the CESoPSN encapsulation requires less processing power than the 'SAR engine' used by TDMoIP in the AAL1 mode. The corollary they allege is that CESoPSN may be implemented purely in software while TDMoIP requires assistance from special hardware. Were this to be completely true it would indeed be a strong point in favor of CESoPSN.
Let's look a bit more closely at the computation required to implement the AAL1 functionality. On the transmit end we must add a sequence number every 48 bytes. This sequence number is error protected, but since it may take only eight possible values, the entire byte that needs to be inserted may be readily precomputed and recalled from an array. For structured mode, there is the additional task of inserting a structure pointer once in the sequence of eight cells. To compute this pointer one must subtract the position of the first byte in the pointer's cell from the position of the first byte in the next structure. Hence the entire computational burden is seen to be a few trivial operations.
In the receive direction one is required to check the cell sequence number and normally one ignores the pointer, since it must only be used when information has been lost. So here too the computational burden is seen to be minimal. In fact, pure software implementations of TDMoIP exist, and are comparable in cost per port to CESoPSN implementations.
More importantly, what percentage of the total MIPS budget is consumed by these AAL1 functions? One should not forget that there are several extremely MIPS intensive tasks that must be carried out in any TDM transport implementation. For example, clock recovery is a number-crunching DSP task compared to which the entire encapsulation routine is negligible. Hence in the common scenario where clock recovery is required the question of encapsulation complexity is moot. Similarly, in networks with packet loss, some method of packet loss concealment should be used to conceal the effect of lost packets on perceived audio quality. Such packet loss concealment algorithms consume computational resources that must be considered. In actual TDMoIP implementations it has been estimated that the AAL1 mechanisms consume no more than 5% of the processing budget.
So while CESoPSN's computational demands are indeed somewhat lighter than TDMoIP's, in full-featured implementations this difference is negligible.

5 Handling Packet Loss

Were packet networks to be as reliable as TDM networks, all methods of transporting TDM would function equally well. In particular, sending arbitrary segments of TDM data (as is done in draft-anavi's "raw mode") would be the optimal solution. Unfortunately, packet networks suffer from packet loss, packet misordering and packet delay variation. These problems give rise to challenges that must be met by the transport protocol. In this section we will consider the effect of packet loss.
CESoPSN's solution to the packet loss problem is to insist on the transport of whole TDM frames, or what is known in CESoPSN jargon as "the ability to interpret every single packet". When a

packet is lost no resynchronization of the TDM data stream is required, as each packet contains the same structure TDM data in the same place. Of course one can not simply drop packets and play the next packet, as the TDM timing would be destroyed. So CESoPSN recommends to compensate by insertion of a packet containing all ones, thus simulating TDM fault mechanisms.

This method of packet loss compensation requires only limited intervention, in particular one needn't identify and process individual time slots. This greatly simplifies the implementation - in fact one needn't guarantee that the first bit in the packet correspond to the first bit in an actual TDM frame, as long as the packet contains the proper number of bits. However, we shall see shortly that this limited intervention technique is sufficient only for very low packet loss percentages. This is consistent with the design philosophy of CESoPSN which assumes a 'well engineered' network with minimal packet loss. CESoPSN proponents have stated on several occasions that they expect their protocol to be employed for networks with less than 0.2% packet loss.

A major limitation of this solution has to do with TDM signals that carry CAS signaling. As will be discussed in the next section, in such cases the CESoPSN packet loses the characteristic of individual packet interpretability, as the single TDM frame is no longer the relevant structure.

Unlike CESoPSN packets, successive TDMoIP packets do not contain the same TDM timeslots in the same positions, and thus the processing of packet loss is necessarily more complex. When packet loss is detected TDMoIP in structured AAL1 mode looks for the structure pointer and then jumps to the beginning of the next structure (whether that structure be an individual frame, or a CAS multiframe). The proper number of TDM bytes to be inserted is calculated and 'interpolation' data introduced.

Packet loss in voice traffic can cause in gaps or artifacts that result in choppy, annoying or even unintelligible speech. The precise effect of packet loss on voice quality, and the development of packet loss concealment algorithms have been the subject of detailed study in the VoIP community. Their results can be summarized as follows: 1) One percent packet loss causes perceived voice quality to drop from toll-quality to cell-phone quality. 2) Above two percent, packet loss is the dominant cause of voice quality deterioration, compressed and uncompressed speech becoming comparable in quality. 3) Packet length is not a significant factor (at least for lengths typically employed in VoIP). 4) By using appropriate packet loss concealment algorithms (PLC) five percent packet loss of uncompressed speech can be comparable or better than cell-phone quality.

Unfortunately, these results are not directly applicable to TDMoIP because VoIP packets typically contain between 80 samples (10 milliseconds) and 240 samples (30 milliseconds) of the speech signal, while TDMoIP packets may contain only a single sample, or perhaps a very small number of samples.

The simplest PLC to implement is to blindly insert a constant value in place of any lost speech samples. CESoPSN's insertion of all ones is equivalent to this technique with a specific constant value (and only this value and all-zeros are equivalent to a consistent constant value). Since we can assume that the input signal is zero-mean (i.e. contains no DC component) minimal distortion is attained when this constant is chosen to be zero. This is in fact precisely what happens when a G.711 μ-law codec receives a word containing all ones, but unfortunately is not true for A-law.

A slightly more sophisticated technique is to replace the missing sample with the previous one. Even in the single sample case it is decidedly better than replacement by zero due to the typical low-pass quality of speech signals, and to the fact that during intervals with significant high frequency content (e.g. fricatives) the error is less noticeable.

For TDMoIP, a packet is usually declared lost following the reception of the next packet, hence the both the sample prior to the missing one, and that following it are available. This enables us to estimate the missing sample value by interpolation, the simplest type of which is linear interpolation, whereby the missing sample is replaced by the average of the two surrounding values. This serves to conceal the packet loss event.

More sophisticated methods of packet concealment are based on statistical models. Standardized speech compression algorithms have had integral packet loss concealment methods for some time, and more recently the ITU-T has standardized a packet loss concealment method for uncompressed speech. For TDMoIP a new interpolation method has been developed, which has been shown to maintain perceived voice quality at packet loss rates that exceed five percent.

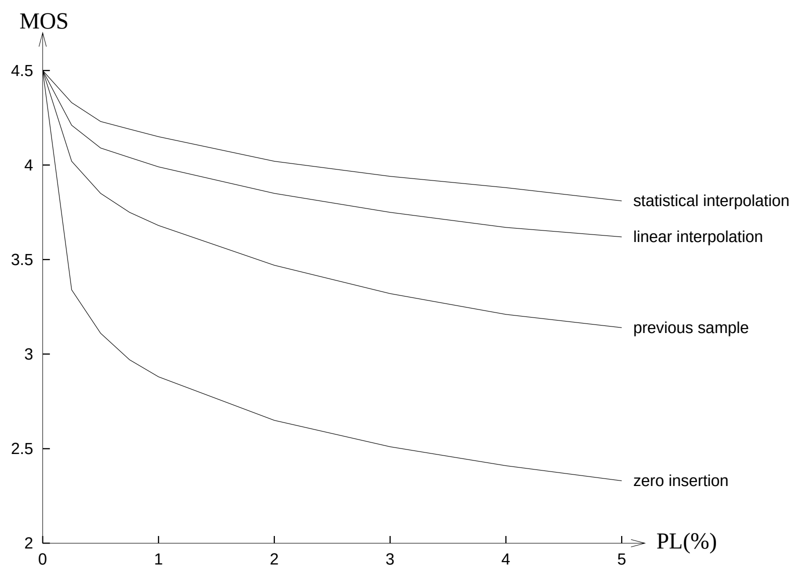

Figure 3: Effect of packet loss on perceived voice quality, using various interpolation techniques.

In figure 3 we depict the results of a controlled experiment to determine the effect of various percentages of uniform packet loss on perceived voice quality. For all cases the MOS resulting from the use of zero insertion is less than that obtained by replacing with the previous sample, which in turn is less than that of linear interpolation, which is slightly less than that obtained by statistical interpolation. In order to appreciate the vertical scale, consider MOS of 4.0 or above to be toll quality voice as heard on a standard telephone, while 3.5 is about the quality of a cellular phone.
The effect of bursty packet noise is more pronounced, but does not change the ordering or relative differences between the interpolation schemes.
The bottom line is that the explicit data handling for packet loss events required by TDMoIP is an advantage, not a burden. In low computational complexity implementations simple interpolation schemes may be employed. For networks that may suffer from high packet loss, more sophisticated algorithms offer higher performance. CESoPSN's philosophy of sending all ones is indeed of lower complexity, but is only acceptable for networks with small fractions of one percent packet loss.

6 Handling TDM Signaling

TDM networks employ CAS or CCS signaling, and these signals must be transported over the IP or MPLS networks for the TDM end-systems to properly function. CAS, or Channel Associated Signaling, is carried in the same T1 or E1 frame as the voice signals, but not in the speech band.

VoIP systems detect the CAS bits, interpret them according to the appropriate protocol, send them through the IP network using some messaging protocol and finally regenerate and recombine them at the far end. Both TDMoIP and CESoPSN transparently transport CAS signaling.

Since CAS signaling is slower than the TDM traffic in a timeslot, one needn't update all the CAS bits every TDM frame. Hence CAS systems cycle through all the signaling bits only after some number of TDM frames, defining a new structure known as a multiframe or superframe. Common multiframes are be 12, 16, or 24 frames in length, corresponding to 1.5, 2 and 3 milliseconds in duration.

As we have mentioned above, when there is CAS signaling (or any other reason to preserve that multiframe structure) the AAL1 pointer indicates the beginning of the next multiframe. Until the time comes for playing out of that data, one can replay the previous data from the proper multiframe phase. This ensures that all CAS signaling will remain intact; for example, phones that were originally on-hook will remain on-hook, while those off-hook will remain off-hook. In addition, such replaying induces 'previous sample' packet loss concealment.

When CESoPSN transmits all ones towards the destination TDM device, it relies on that device to detect the fault condition and compensate using its own fault recovery mechanisms. TDM switches and even PBXs are designed to be able to withstand a certain interval of fault condition, but such conditions are expected to be extremely rare events. By allowing common packet loss events to induce TDM fault conditions, CESoPSN relentlessly tries the patience of the TDM switches. In such switches extended periods of TDM alarm may cause resynchronization procedures to kick-in, or even mandatory shut-down (typically 10 seconds before resynchronization is attempted).

In order to ameliorate this situation, the successive CESoPSN drafts invented new methods for handling CAS signaling. In early versions CAS was not treated, thereafter it was to be delivered once in a millisecond by a special packet. In the latest draft (version 5) CAS has once again been taken out and declared to be for further study.

ISDN signaling and SS7 are examples of CCS, or Common Channel Signaling, which are often found occupying a TDM timeslot. The special HDLC mode of TDMoIP is designed for efficient and reliable transfer of such CCS signaling. CESoPSN specifically states that CCS signaling is beyond its scope.

7 Packet Delay Variation and Clock Recovery

As we have mentioned before, CESoPSN assumes a 'well engineered' network with limited PDV, although precisely how much has never been articulated. What Axerra has stated is that they utilize some method of minimizing jitter buffer size, a statement consistent with the assumption of low PDV.

TDMoIP proposes the use of adaptive clock recovery, without the need for explicit timestamps (e.g. from RTP). TDMoIP implementations have been shown to converge to the correct clock frequency within seconds, and to conform to all G.823/G.824 jitter and wander specifications, even when PDV is significant.

One question frequently raised is why TDMoIP in AAL1 mode does not always use SRTS, a clock recovery mechanism defined in ATM-AAL1. To understand why, one needs to know something about SRTS. Synchronous Residual Time Stamp was defined for ATM networks, which (unlike IP and MPLS networks) define the physical layer. ATM networks have a physical layer clock upon which both ends of the link agree, and SRTS encodes the difference between the source TDM's clock and this ATM physical layer clock. The receiver, having access to the ATM clock, can then easily reconstruct the required TDM clock by applying the SRTS differences.

SRTS can certainly be used in any draft-anavi implementations that have a physical layer clock to reference. Unfortunately, not all IP or MPLS networks have end-to-end physical clock integrity. When TDMoIP is used over a single hop network, for example an optical link or a copper ethernet link for which the transmitting rate can be locked, SRTS can, and should, be used. The SRTS mechanism is automatically available in TDMoIP, but can not be used in CESoPSN.

8 Network Congestion Avoidance

When a network route used by a constant bandwidth service becomes congested, the only recourses are to reroute the service and avoid the congested links, or to withdraw the service. In many cases the former option is not available, due to the service provider lacking the required control over underlying network. The remaining option, that of service removal, is obviously an unacceptable alternative for TDM services. Not only does such service withdrawal impact a large number of users, its impression on users accustomed to "five nines" availability would be highly detrimental to the service's (and the service provider's) image.

The situation is all the more deplorable since congestion is not expected to be the norm. Assuming that there is sufficient bandwidth for the TDM flow to statistically co-exist with the other traffic, congestion will be solely due to temporary load peaks. If such a peak leads to withdrawal of the TDM tunnel, the loops will be disconnected, the TDM equipment will suffer red alarm, a nd the reconnection time before other users may be served will be significant.

TDMoIP's AAL2 mode may be used to mitigate problems deriving from network congestion. AAL2 may be combined with native service processing such as silence suppression and voice compression, and so such a tunnel may consume less bandwidth a priori, and hence its effect on neighboring flows is minimized. Moreover, when congestion *is* detected, there are several bandwidth conserving options available that facilitate significant further reduction. For example, speech compression could be enabled which can reduce bandwidth by a factor of ten; voice activity detection and comfort noise generation typically contribute a further factor of two, reducing the bandwidth to 5 percent of the original. At this level there is little reason to withdraw the tunnel, and the service can be maintained. The compression may indeed degrade the quality perceivable to users, but this would undoubtedly be considered much more acceptable than service outages. Once the congestion clears the original service emulation characteristics can be restored.

Since CESoPSN does not provide an AAL2-like mode, momentary congestion must lead to service withdrawal.

9 Interworking with Existing Circuit Emulation Systems

CESoPSN proponents have stated that they consider ATM AAL1 services to be limited in extent, and hence doubt the need to interwork with them. In addition, they speak of the need to upgrade such devices to CESoPSN devices, i.e. they suggest that operators and users of existing ATM AAL1 services perform forklift upgrades to their technology.

TDMoIP provides an easy migration path for operators and users presently using ATM CES, since the interfaces remain the same, as do the general service parameters. In addition, TDMoIP was designed to easily interwork with existing ATM-based circuit emulation systems, so that an operator can run any combination of IP, MPLS and ATM based CES services.

For example, an operator with an ATM network can now add an IP or MPLS based one and extend over it CES traffic originating in the ATM network. To do this the operator needs only to terminate the ATM layer and extract the AAL1 traffic, which is then encapsulated according to the TDMoIP layering.

In order to use CESoPSN in this application, the ATM network's AAL1 layer would need to be terminated, the TDM stream reconstructed, and then new packetization performed. Such double encapsulation leads to degradation of timing quality and to numerous problems due to packet or cell loss.

10 Summary

	TDMoIP	**CESoPSN**
packet size	flexible tuning	locked to TDM frame
control word	standard (Martini compatible)	nonstandard
payload	adapted TDM	(padded) TDM
computational complexity	higher	lower
bandwidth efficiency	usually higher	usually lower
clock recovery	adaptive - standards compliant	RTP-based - noncompliant
packet loss	optimal interpolation	AIS - relies on end equipment
CAS signaling	full treatment	untreated
CCS signaling	efficient transport	untreated
bandwidth conservation	by AAL2 mechanism	untreated
congestion avoidance	by AAL2 mechanism	service withdrawal
ATM CES interworking	simple	complex