TDM Circuit Emulation over IP Network

Meghanad Shingate
Department of Electrical Engineering
IIT Bombay
meghanads@ee.iitb.ac.in

December 4, 2009

Abstract

For rural areas, areas of low density and areas where it is not possible to deploy optical fiber backhaul due to geographical reasons, cellular backhauling is mainly done by using microwave links. These microwave links are costly and also require absolute radio frequency stability of the air interface, so network operators now looking for low cost technology. Due availability of TDM Circuit Emulation over IP (TDMoIP) technology it is possible to implement low cost wireless cellular backhaul. Also using TDMoIP it is possible to converge all type of communications voice, data and video towards full IP solution. The present IP networks offer low cost transport and flexibility for any kind of data. This gives good reason for data, voice and video services to migrate towards IP. TDM is being used from many years for reliable transport of voice, data, and video traffic and network providers had already invested huge amount in TDM based equipments so migration from TDM to IP will occur in transitional phase only, where some services continue using legacy TDM networks, while core network moves towards IP. But there are many difficulties in attaining TDM goals in TDMoIP. There are several different architectures offering different degree of robustness and quality of service. The possible architectures are presented here and their pros and cons are discussed.

1 Introduction

TDM over IP (TDMoIP) is the emulation of time-division multiplexing (TDM) over a packet switched network (PSN). Here TDM means T1, E1 signals, while PSN is based on IP. IP is packet based network, and it's traffic is unpredictable, therefore transport of TDM over IP network posses several challenges with respect to quality and security of transport, which TDM network already has. There are various issues Gateway devices have to address, such as packetization delay, packet jitter, timing and synchronization etc. Out of these achieving synchronization is most important. By using suitable clock recovery algorithms it is possible to achieve timing goals of TDM for IP networks. TDMoIP scheme can be represented as shown in figure 1. We have legacy TDM network which is producing TDM frames and send it to Interworking Function (IWF), which acts as gateway between TDM and IP network, then IWF convert this TDM frames in to IP packets and packets are injected into IP network. At far end these packets are collected by another IWF which performs reverse operation i.e. converting packets back into TDM stream along with timing and signaling information.

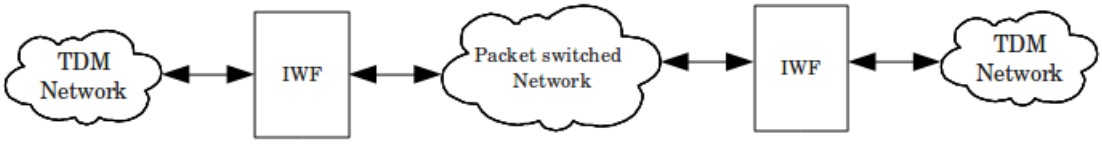

Figure 1: TDMoIP scheme

2 Traditional TDM

2.1 Importance of Timing

TDM signal is actually isochronous signal, i.e. on an average time between two adjacent bits is same. This time interval is called Unit Interval (UI); for E1 UI is 488 ns and for T1 it is 647 ns. There are some tolerances specified by standards that have to maintain in order to maintain isochronicity, so TDM have to employ highly stable clock. Consider that receiver is utilizing its own clock for converting physical signal back into bit-stream. If receiver clock runs at precisely same rate as that of transmitter, then receiver has only have to detect only sampling phase. How ever any mismatch of clock rates, no matter how small, causes bit-slips. Consider for example if receiver clock is slower than that of transmitter, say by one part per million (ppm), then receiver will output $999,999$ bits for every $1,000,000$ bits transmitted. One bit-slip per million bits seems acceptable at first glance, but causes one bit error per second for 1 Mbps E1 signal. ITU-T recommendations permit a few bit slips per day for a 64 kbps channel, but strictly prohibit bit slips entirely for higher-rate TDM signals.

Temperature changes and aging will inevitably affect a clock's rate. Hence no clock will remain at precisely accurate forever, and no two physical clocks will run at exactly the same rate for long periods of time. In order to eliminate bit slips, we must ensure both that the long-term average Unit Interval of source and receive clocks should be same, and that its short-term deviations from the average are bounded within certain limits.

The variation of a clock's rate over time is caused by two components, jitter and wander. Wander represents slow, smooth drifting of clock rate due to temperature changes, aging; while jitter conveys fast, jumps in Unit Interval caused by phase noise and bit-stuffing mechanisms. The border between the two components is set at 10 Hz. In order to avoid bit slips, standards impose strict limits on tolerable jitter and wander of TDM clocks.

2.2 Timing Distribution

How the absolute accuracy is achieved in TDM? In conventional TDM network we are using the hierarchical timing distribution. In every TDM network somewhere there is at least one highly accurate clock with long-term accuracy of one part in 10^{11}, called Primary Reference Clock(PRC). PRC is acts as master clock; every other clock in network is deriving its timing directly or indirectly from PRC. For entire TDM network to work as whole this hierarchical distribution is necessary. Each clock that ultimately derives its timing from PRC is called traceable to that PRC. Bellow PRC there are Synchronization Supply Units (SSU) which in turn provides clock to down tier clocks. Figure 2 shows hierarchical timing distribution scheme employed in TDM networks.

As there is distribution of clock information from PRC to all other clocks in network, it shows isochronous nature of TDM signal. when TDM signal clocked by PRC is received by receiver it compares observed time interval to its local clock interval and increases its clock rate somewhat if its clock rate is slower than that of PRC and vice versa. This correction ensures

the long-term average Unit Interval will be correct. But this change in clock introduces jitter since the clock rate is instantaneously changed. In addition to this wander is also introduced but it takes long time to become apparent so it can be compensated latter. Hence secondary clock is somewhat degraded in quality and so on.

When down tier clocks loose their connection with PRC i.e. master clock they expected to go in holdover mode. In holdover mode clock tries to maintain clock accuracy, even if it do not have access to master. For this reason the standard requirements in holdover mode are lower.

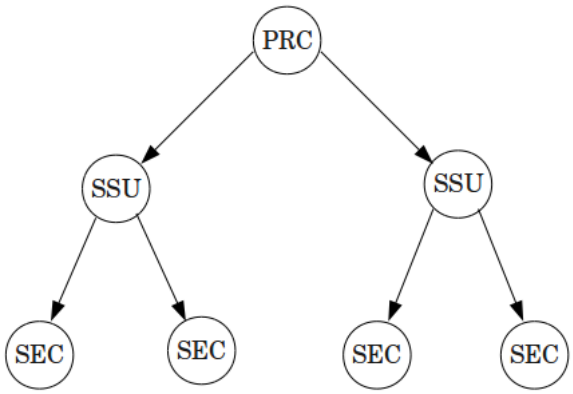

Figure 2: Hierarchical timing distribution

2.3 Jitter and Wander

Jitter and wander are defined as respectively "short-term and long-term variation in significant timing instants of digital signal from their ideal position in time domain". One way we can think of digital signal as continually varying position forward and backward in time domain with respect to the source clock. Jitter and wander both having amplitudes specifying how much the signal is shifting in phase and frequency and how quickly varying with respect to the source clock. Jitter and wander are specified in ITU-T G.810 standard. Jitter is phase variation with frequency component greater than 10Hz, and wander is less than 10 Hz.

2.3.1 Metrics for Jitter

UI (Unit Interval)

Jitter amplitude is specified in Unit Intervals (UI), where one unit interval is one data-bit duration, irrespective of data rate. Jitter amplitude is quantified in Peak-to-Peak value rather than RMS value as when peak jitter is present it causes bit error.

2.3.2 Metrics for Wander

For wander measurement we require 'wander free' reference, relative to which wander of other is measured. Obviously any PRC signal can be used as reference. Wander is measured in different units as listed bellow.

- TIE(Time Interval Error)
- MTIE(Maximum Time Interval Error)
- TDEV(Time Deviation)

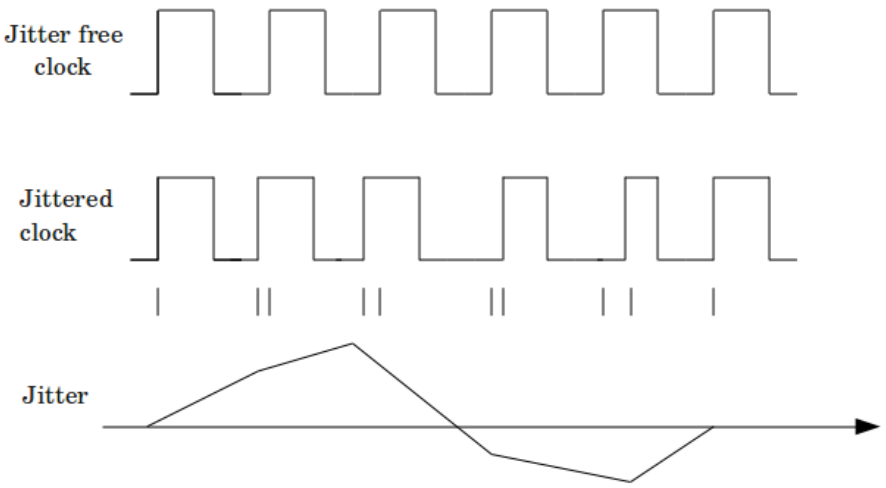

Figure 3: measurement of jitter

TIE (Time Interval Error)

It is measured in nanoseconds. It is defined as phase difference between measured signal and reference signal. First time is set to zero also phase difference is set to zero and then measurement is done for time T, so TIE is nothing but phase change in time T.

MTIE (Maximum Time Interval Error)

MTIE represents the phase transients. It is a function of a parameter τ called the Observation Interval. MTIE (τ) is the largest Peak-to-Peak TIE in any observation interval of length τ. For calculation MTIE at a certain observation interval τ from the measurement of TIE, a time window of length τ is moved across the entire duration of TIE data, and peak value is stored. The peak value is the MTIE (τ) at that particular τ. This process is repeated for each value of τ desired.

TDEV (Time Deviation)

TDEV is a measure of wander that characterizes its spectral content. It is also a function of the parameter τ called Observation Interval.

2.4 Attaining Timing Goals in TDM Networks

TDM network uses several mechanisms to attain timing requirement mentioned above. First, TDM physical layer signals are designed in such way that ensures that single bit duration can be identified easily, irrespective of data being transferred, like while data is being transferred by using coding more numbers of transitions are ensured. If there are more number of zeros or ones it is likely to be that receiver may loose the synchronization so line coding used to add sufficient number of marks(i.e. ones); so avoiding long runs without transition.

Second, TDM slave clocks receiver circuitry is designed in such manner that they can tolerate the some amount of jitter and wander at their inputs, after receiving clock they does filtering and sends clean clocks to their slaves. The instant that a TDM bit is received equals the time it was transmitted plus the propagation time over the physical channel plus a zero mean stochastic

component noise attributed to temperature, bit justification. So to eliminate this stochastic component, some sort of averaging process is required to filter this component. A Phase Locked Loop (PLL) is well suited for this task because it can lock onto the average bit rate, regenerating a clean clock signal that approximates the original bit rate.

Third, as before mentioned is hierarchical timing distribution. Every clock in TDM network is traceable to PRC. Each time it is observing the masters clock and compare it with its local clock, if any discrepancies are seen it adjusts its clock to masters clock.

3 Attaining TDM Timing Goals for TDMoIP

TDMoIP is new technology that enables TDM traffic over packet switched networks, such as IP. In this technology TDM stream is encapsulated at network ingress and then it is transported across the IP network and at far end this packets are received then original TDM signal must be reconstructed, emulating original TDM. The transport of TDM signals through packet networks requires that the signals at the output of the packet network comply with TDM timing requirements; this is crucial to enable interworking with TDM equipment. These timing requirements are independent of the data (voice or data) transported by TDM signal.

The major technical barrier is recovery of clock at receiver end. While in TDM we are transmitting the timing signal along with data, on physical layer. So far asynchronous PSN not transferring any timing information along with data, even matter is becoming worst due to packet loss (PL) and packet delay variation (PDV). So some sophisticated clock recovery mechanisms are required in order to achieve desired timing accuracy in presence of all these barriers.

One thing we can note that TDMoIP packets are injected at network ingress with constant rate as TDM source is sending data bits at constant rate. so we can calculate nominal packet delay using RTP protocol. Delay experienced by packets is this nominal delay plus some stochastic random component known as PDV. So here also we can recover the clock timing, this random component we can overcome by using "Jitter Buffer" which acts as elastic store. So we can get smoothened bit stream out of buffer. But problem is that in case of TDM we have reference clock with respect to which transmitter is sending the data, in case of TDMoIP no such reference is available, and precise rate at which data should be *'clocked out'* of the jitter buffer is unknown.

In some cases timing is derived at end-points, for example by placing GPS receivers at both sides. Another alternative is attempt to derive the clock timing from TDM traffic only, it is possible because TDM device is producing bits at constant rate, just we receive bits in packets that suffers from random packet delay. So this PDV can be considered as zero mean random process, our job is again same some kind of averaging that eliminates effect of PDV and captures average rate of bit stream. In TDM case it was PLL that we used for averaging, but in TDMoIP case jitter and wander are of higher magnitude so conventional PLL will not work well.

One conventional mechanism is to adapt local clock depending on the receivers jitter buffer level. To understand working of this mechanism lets consider first that there is no PDV so we get data in jitter buffer at constant rate i.e. buffer is gets filled linearly, if we read at some constant rate but less than transmitters rate then buffer level starts increasing, so this rise in buffer level can be detected and by increasing reading rate somewhat we can settle the buffer level at constant, and that is the transmitter frequency. If there is PDV buffer level no longer will be rise or fall linearly, rather it will be fluctuating at its mean position. So by using PLL that locks to average rate so buffer level settle within certain fluctuations, i.e. buffer level settle down to certain precise frequency alignment between receiver and senders clock.

But this PLL to work well there are many faults, like, PLL should observe sequence of buffer levels for long period before it locks to source clock, hence convergence time is more. Second, buffer level may be settling down far from desired position at jitter buffer center, thus making it vulnerable to overflow and underflow. Also size of jitter buffer should be such that it will

compromise between jitter-wander and added latency due to buffer size. Low resolution of jitter buffer leads to high wander generation.

To attain TDM timing goals, several different TDMoIP architectures exist offering different degrees of robustness, service quality guarantees and management. There are two main architectures, unstructured and structured TDMoIP[1], which are discussed bellow.

3.1 TDMoIP Architectures

3.1.1 Unstructured TDM over IP

It is also called as structure agnostic TDMoIP. In unstructured TDMoIP, TDM bit stream is transparently encapsulated and transported across the IP network. Here IP network is viewed as the point-to-point links. Main advantage of this architecture is that circuit emulation inerworking function can be implemented without understanding any TDM services and signaling, so that it does not require any change in TDM network nodes.

The disadvantage is that it introduces uncontrollable delay since packets must be packetized and de-packetized at each TDMoIP network node.

Figure 4: Unstructured TDM(TDM=bit stream)

3.1.2 Structured TDM over IP

Structured mode refers to the case where TDM channels are individually transported across the IP network or grouped depending on their destination. Advantage of these scheme is that only active channels are encapsulated and transported and send across IP network, so ensuring very efficient utilization of network resources. TDMoIP packetization and de-packetization is done only once, so reducing end-to-end delay considerably. As we can handle each channel individually, it allows channel management based on priority and quality of service requirements. The disadvantage is that TDMoIP interworking function need to understand the signaling of TDM, also additional packet processing is required. But as we allowing prioritized channel management it may happen number of small packets in channel is becoming large, if we are grouping the channels having same destination then along with high priority signal low priority signals also getting high priority. This can be implemented by giving some intelligence to TDMoIP hops.

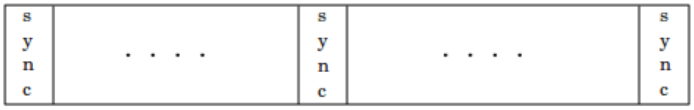

Figure 5: Framed TDM

Figure 6: Channelized TDM

In deployment of TDMoIP there are many challenges that we have to face as listed bellow.

- Packetization.
- Clock recovery and synchronization.
- Attenuate the packet delay variation.
- Compensate for packet loss and out-of-sequence packets.

Out of these clock recovery and synchronization is most important. Four principal timing distribution scenarios may be identified, that differ in the availability and placement of timing sources.

4 Timing Distribution Scenarios

ITU-T Rec.Y.1413 describes following timing distribution scenarios[5].

4.1 Reference Clock Available at TDM End Systems

Figure 7 depicts scenario wherein TDM end systems share a reference clock (by some means). Primary reference clock may be available at both end systems, due to accuracy of PRC two clock can be considered identical. While transmitting data to end systems IWFs (Inter working function) can derive clock from end systems by slaving their clock circuitry to end systems clock.

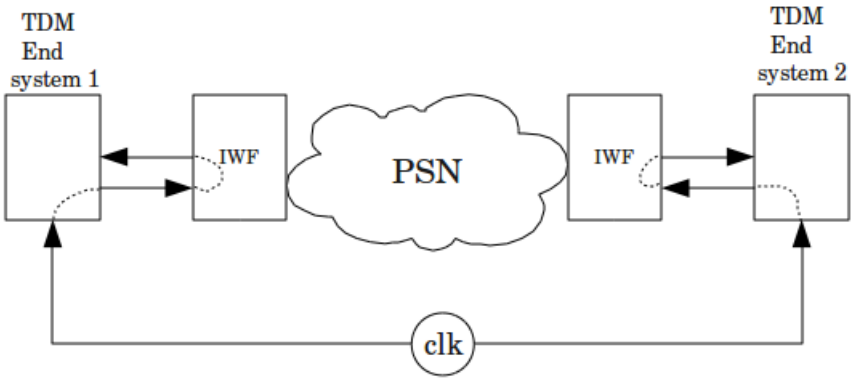

Figure 7: Reference clock available at end systems

4.2 Reference Clock Available at IWFs

Figure 8 depicts the scenario where common clock is available i.e. PRC. This method refers to the fully synchronous network operation as PRC is available at network end systems. TDM end systems while transmitting towards IWFs have to slave their clock circuitry to IWFs clock. It is seen that this method does not preserve service timing, as IWFs using PRC.

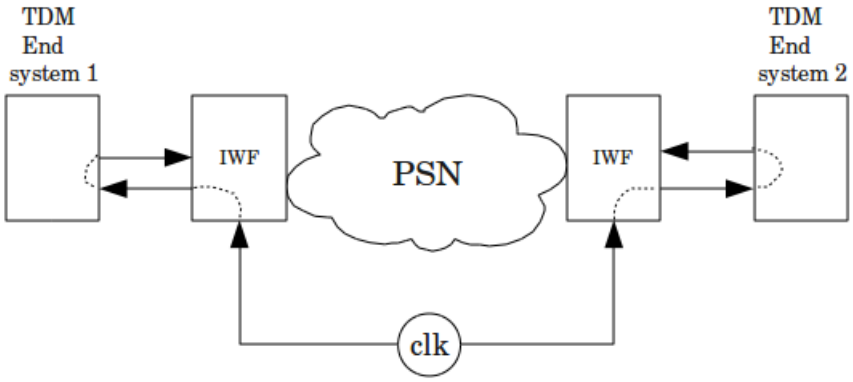

Figure 8: Reference clock available at IWFs

4.3 Differential Method

In this method IWFs are traceable to PRC and TDM end systems are using their local clock for transmitting data. Difference between end system clock and PRC is encoded in some manner e.g. using Synchronous Residual Time Stamp (SRTS) method or Real Time Protocol(RTP) at IWFs and transmitted across the IP network. Here it is require that end system-2 should slave its clock circuitry to that of end system-2. At far end this service clock is get recovered using this timing message. It should be highlighted that here service timing is preserved (see figure 9).

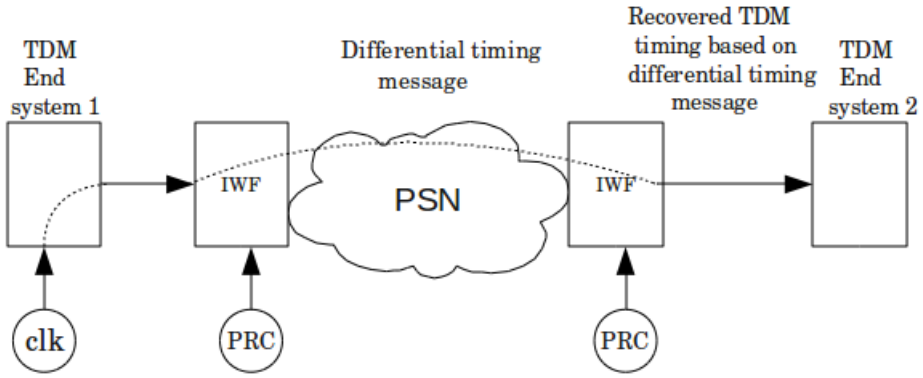

Figure 9: Differential Method

4.4 Adaptive Clock Recovery

In this method end system-1 sending data at rate determined by its local clock and end system-2 have to slave its clock circuitry to that of end system-1. IWF-1 slaves its clock circuitry to that of end system-1 while transmitting towards end system-1. At far end after receiving packets at IWF-2 this clock is recovered adaptively using TDM traffic characteristics at network egress. So while transmitting towards end system-2, IWF-2 slave its clock circuitry to this recovered clock rate and end system-2 slave its clock circuitry to that of IWF-2 (see figure 10).

The adaptive clock recovery function utilizes only observable characteristics of the packets arriving through the IP network, such as the precise time of arrival of the packet to the IWF, and the fill-level of the jitter buffer as a function of time. Due to the packet delay variation in

the IP network, some kind of averaging is to be done at IWFs. Phase Locked Loops (PLL) are well suited for this task.

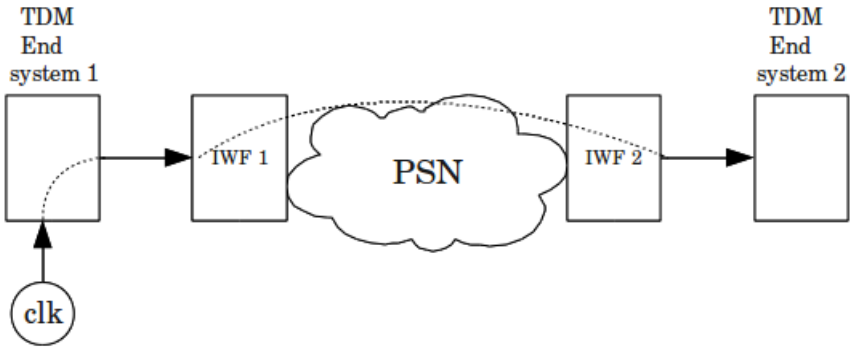

Figure 10: Adaptive clock recovery

4.4.1 One Method for Adaptive Clock Recovery[3]

In this method we are using one term *Instantaneous Packet Delay Variation (IPDV)*. Packet Delay Variation (PDV) is variance of the packet delay and IPDV is defined as delay difference between consecutive packets.

$$IPDV(n) = d_n - d_{n-1} \tag{1}$$

Where d_n is n^{th} packet delay. In this method for buffer level measerement we are using two time scales[3]. One is larger for estimate clock frequency and second smaller to collect statistics. Larger interval m is integer multiple of smaller interval n i.e. $k(m) = k_m \tau_n$, where k_m is positive integer and $\tau_n = 1/f_2(n)$ is duration of smaller scale and $f_2(n)$ is initial estimate of clock frequency at receiver. Here we are assuming that PDV as zero mean random process. So we have to choose k_m such that for larger interval $k(m)$ average PDV is close to zero, then sample the buffer. By measuring buffer level difference from m to $m+1$ time instants, we are estimating the clock frequency.

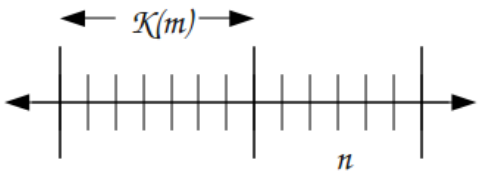

Figure 11: Two time scale model

Frequency Difference Estimation Let f_1 is actual source frequency, and $f_2(n)$ is initial estimate of clock frequency at receiving end gateway of TDMoIP traffic. So clock frequency received at receiver can be written as $f_1 + \omega(n)$, where $\omega(n)$ is noise term, which stands for PDV. So frequency difference is,

frequency difference $= f_1 + \omega(n) - f_2(n) = \Delta f(n) + \omega(n)$

where $\Delta f(n) = f_1 - f_2(n)$ is actual frequency difference. Let $\theta(n)$ be buffer level, then change in buffer level is,

$$\theta(n+1) - \theta(n) = (\Delta f(n) + \omega(n))\tau_n \tag{2}$$

$$\Delta\theta(n) = \Delta f(n) + \eta(n) \tag{3}$$

where $\tau_n = 1/f_2(n)$ is sampling interval, $\eta(n) = \omega(n)\tau_n$ is noise term (packet jitter) of buffer in terms of packets, which is assumed to be zero mean random process. Since $f_2(n)$ is updated only for larger intervals, so within $[n, n+k(m)]$ interval $f_2(n)$ and $\Delta f(n)$ does not change, so above equation can be written as,

$$\Delta\theta(n) = \frac{\Delta f(m)}{f_2(m)} + \eta(n) \tag{4}$$

Here, intuitively $\eta(n)$ and $IPDV(n)$ are effect of same cause, packet jitter. So if for larger interval $IPDV(n)$ become zero $\eta(n)$ also reduces to zero. Thus run time average of $IPDV(n)$ is calculated and when it is become less than some ε, say for k_m^* small intervals, then we sample buffer to estimate the frequency difference. So taking expectation of Eqn.(4) we get,

$$\frac{1}{k(m)} \sum_{i=n}^{n+k(m)-1} \Delta\theta(i) - \frac{\Delta f(m)}{f_2(m)} = \frac{1}{k(m)} \sum_{i=n}^{n+k(m)-1} \eta(i) \tag{5}$$

Define right hand side of above Eqn.(5) as,

$$X(k(m)) = \frac{1}{k_m^*} \sum_{i=n}^{n+k_m^*-1} \eta(i) \tag{6}$$

$$k_m^* = argmin_{k(m)}(X(k(m))) \tag{7}$$

above equation can be considered as optimization problem, i.e. minimum value of $X(k(m))$ is taken to be less than ε for $k(m) = k_m^*$. Thus for k_m^* small intervals Eqn.(5) can be written as,

$$\frac{1}{k_m^*} \sum_{i=n}^{n+k_m^*-1} \Delta\theta(i) = \frac{\Delta\hat{f}(m)}{f_2(m)} \tag{8}$$

where $\Delta\hat{f}(m)$ is estimate of $\Delta f(m)$. Also sum of smaller changes of buffer level can be written as difference in buffer level for larger interval. So we can write Eqn.(5) as,

$$\theta(m+1) - \theta(m) = k_m^* \frac{\Delta\hat{f}(m)}{f_2(m)} \tag{9}$$

Recovered Frequency So recovered frequency can be written as,

$$f_2(m) = f_2(m-1) + \alpha\Delta\hat{f}(m) \tag{10}$$

Let estimation error be $\Delta\tilde{f}(m)$, then $\Delta\tilde{f}(m) = \Delta\hat{f}(m) - \Delta f(m)$. So Eqn.(10) becomes,

$$f_2(m) = (1-\alpha)f_2(m-1) + \alpha f_1 + \alpha\Delta\tilde{f}(m-1) \tag{11}$$

Hence, by choosing appropriate value for α such that $0 < \alpha < 1$ equation (10) we can used for frequency difference estimation. It is also seen from Eqn.(11) that for larger values of α, error component may start dominating leads to instability.

5 Packet Loss

Packet loss occurs when one or more packets traveling across IP network failed to reach destination. Packet loss can be caused by a number of factors, including signal degradation over the network medium, over-saturated network links, corrupted packets rejected in-transit, faulty networking hardware, faulty network drivers or normal routing routines. But in order to maintain timing SOMETHING must be output towards TDM interface when packet is lost. There are many Packet loss concealment methods. Packet loss concealment (PLC) is a technique to mask the effects of Packet loss in Voice communications. Because the voice signal is sent as packets on IP network, they may travel different routes to get to destination. At the receiver a packet might arrive very late, corrupted or simply might not arrive. One of the situations in which the last situation could happen is where a packet is rejected by a server which has a full buffer and cannot accept any more data. Mainly there are two packet loss concealment techniques, namely, Replay of previous sample and Interpolation. Following graph shows the effect of different PLC techniques on voice quality in mean opinion score[2].

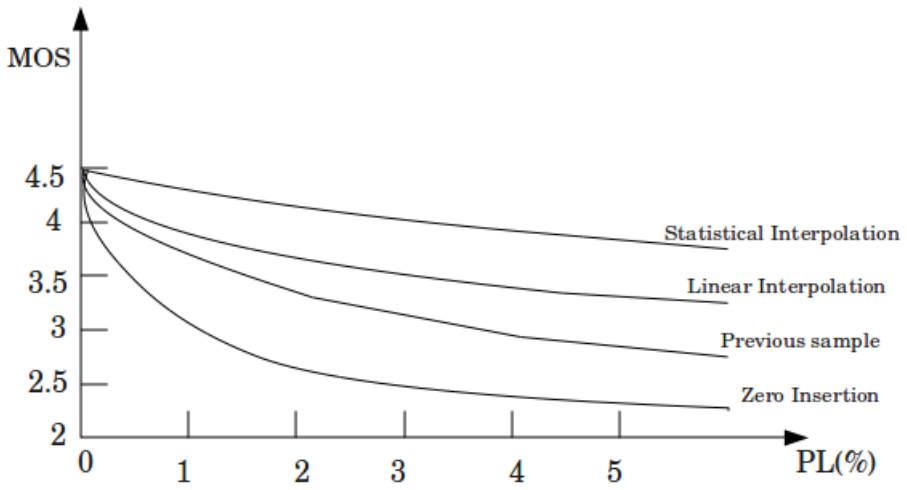

Figure 12: Effect of packet loss on voice quality[2]

6 Out of Sequence Packets

As there are many parallel paths between the routers in PSN networks so for avoid congestion in IP network packets may be routed in different paths for load balancing so experiences different delays. In TDMoIP technology there are two ways that we can handle packet loss. First is if packets are out of order and present in jitter buffer we can re-order them, and second is, if packets are out of order and packets are not present in jitter buffer then we can handle it as packet loss.

7 Application

As fore mentioned we are looking for low cost wireless cellular backhauling. Figure 13 shows the particular application of TDMoIP technology.

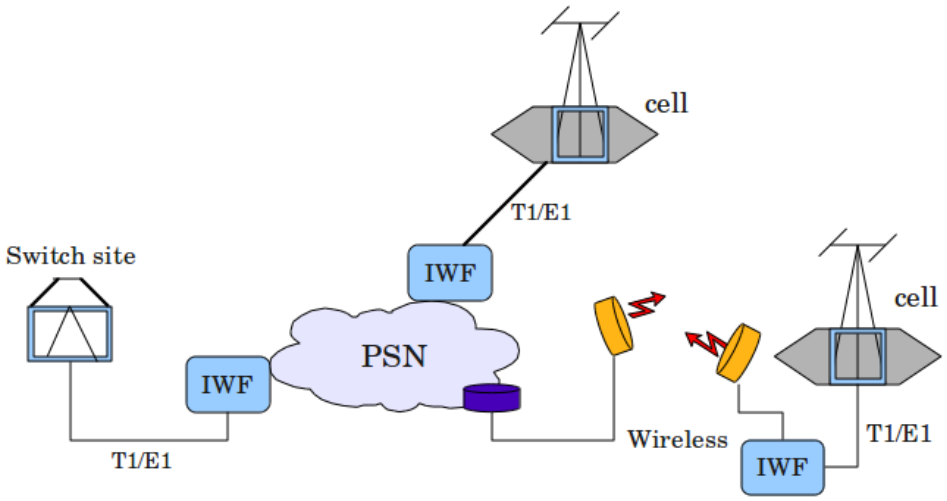

Figure 13: Wireless cellular backhauling

8 Conclusion

From above discussion we infer that it is possible to attain TDM goals over IP by suitable clock recovery algorithms without much changing the underlying TDM infrastructure. But, it is also found that, performance reduces when PDV is very large. By introducing TDMoIP, convergence of all type of communication networks, i.e. voice, video and data, to unique medium is possible.

References

[1] Kure,O.E., Sorteberg, I., Oevsthus, K. *"Architecture for TDM circuit emulation over IP in tactical networks."* in Military Communications Conference, 2003. MILCOM 2003. IEEE.

[2] T. Johnson, *"TDM Circuit Emulation over Packet (CEP-TDM),"* Internet Draft 'draft-johnson-pwe3-tdm-00.txt', Exp: August 2003.

[3] Prem kumar Nonia, *"Feedback System for End-to-End QoS provisioning and clock recovery for TDMoIP traffic,"* IIT Madras, June 2009.

[4] Stefano Bregni, *"Synchronization of digital telecommunication networks,"* 2002, John Wiley and Sons, Ltd

[5] ITU-T Recommendation G.8261/Y.1361 *"Timing and synchronization aspects in packet networks."*