



TDM Timing

By Dr. Yaakov Stein, Chief Scientist, RAD Data Communications

This paper presents a brief overview of the theory and practice of timing in pure TDM and TDMoIP networks.

The Importance of TDM Timing

TDM signals are **isochronous**, meaning that the time between two consecutive bits is theoretically always the same. This time is called the **unit interval** (UI); for T1 signals the UI is defined to be 647 nanoseconds, and for E1 the standards dictate 488 nanoseconds. In order to maintain isochronicity and to remain within tolerances specified by recognized standards, a TDM source must employ a highly stable and accurate clock.

The stringent clock requirements are not capriciously dictated by standard bodies; rather, they are critical to the proper functioning of a high-speed TDM network. Consider a TDM receiver utilizing its own clock when converting the physical signal back into a bit-stream. If the receive clock runs at precisely the same rate as the source clock, then the receiver need only determine the optimal sampling phase. However, with any mismatch of clock rates, no matter how small, **bit slips** will eventually occur. For example, if the receive clock is slower than the source clock by one **part per million** (ppm), then the receiver will output 999,999 bits for every 1,000,000 bits sent, thus deleting one bit. Similarly, if the receive clock is faster than the source clock by one **part per billion** (ppb), the receiver will insert a spurious bit every billion bits. One bit slip every million bits may seem acceptable at first glance, but translates to a catastrophic two errors per second for a 2 Mbps E1 signal. ITU-T recommendations permit a few bit slips per day for a low-rate 64 kbps channel, but strive to prohibit bit slips entirely for higher-rate TDM signals.

Temperature changes, imperfections in materials, aging, and external influences will inevitably affect a clock's rate, whether that clock is atomic, quartz crystal, or pendulum based. Hence no clock will remain at precisely the same rate forever, and no two physical clocks will run at exactly the same rate for extended periods of time. In order to eliminate bit slips, we must ensure both that the long-term average UI of source and receive clocks are identical (any rate difference, no matter how small, will eventually accumulate up to a bit slip), and that its short-term deviations from the average are appropriately bounded.

The variation of a clock's rate over time is conventionally divided into two components, jitter and wander. **Wander** expresses slow, smooth drifting of clock rate due to

temperature changes, aging and slaving inaccuracies; while **jitter** conveys fast, erratic jumps in UI caused by phase noise phenomena and bit-stuffing mechanisms. The border between the two components is conventionally set at 10 Hz. In order to eliminate bit slips, the standards impose strict limits on tolerable jitter and wander of TDM clocks.

Timing Distribution

How are the absolute accuracy and variability (jitter and wander) limits attained? Conventional TDM networks rely on **hierarchical distribution** of timing. Somewhere in every TDM network there is at least one extremely accurate **Primary Reference Clock (PRC)** or **Primary Reference Source (PRS)**, with long-term accuracy of one part in 10^{11} as compared to UTC, the world time standard. The characteristics of this clock, known in North America (see T1.101) as a **stratum 1** clock, are described in ITU-T Recommendation G.811. The accuracy of today's atomic clocks is significantly better than that required by G.811.

The PRC is the master clock from which all other TDM clocks in the network directly or indirectly derive their timing. This hierarchy of time synchronization is essential for the proper functioning of the network as a whole. A clock that ultimately derives its rate from the PRC is said to be **traceable** to that PRC.

The distribution of clock information from the PRC towards other clocks exploits the isochronous nature of TDM signals. When a TDM signal clocked by the PRC is received, the receiver's local clock is compared to the observed UI of the received bits. If the observed rate is higher (lower) than the present rate, the receiver increases (decreases) its local clock's rate somewhat. This correction ensures that the long-term average UI will be correct but introduces jitter, since the local clock's rate was rapidly changed. In addition, wander is introduced, as it can take a long time until small timing discrepancies become apparent and can be compensated. Hence the secondary clock will be somewhat degraded relative to the primary one. In North America this is called a **stratum 2** clock, and is historically used at tandem offices. This idea of a **slave clock** deriving its timing from a **master clock** is continued in hierarchical fashion. TDM signals whose source utilizes a stratum 2 clock are received by TDM receivers that adapt *their* local clocks to track the stratum 2 clock. Such **stratum 3** clocks, historically used at local exchanges, already lead to appreciable bit slips. Finally, customer equipment such as channel banks may use **stratum 4** clocks; such clocks may lead to bit slips every few seconds.

When a slave clock loses the master clock signal it is expected to go into **holdover** mode. In this mode it attempts to maintain clock accuracy, although it no longer has access to its reference. For this reason the standard requirements are lower in holdover

mode, e.g. a Stratum 2 clock that during normal operation is required to maintain a long-term accuracy of one part in 10^{11} , is only required to keep up one part in $1.6 * 10^{-8}$ under holdover conditions. Stratum 4 clocks need not have any holdover capabilities.

The process used by slave clocks to mimic the master clock's rate is actually more sophisticated than that described above. Due to noise and measurement inaccuracies, the receiver needs to observe the difference between its local clock and the observed TDM for some time before deciding to change its clock's rate. The task of clock recovery can thus be seen to be a kind of **averaging** process that negates the effect of the random variations and captures the average rate of transmission of the original bit stream. 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 closely approximates the original bit rate. If the incoming TDM is disrupted, and the secondary clock can no longer remain locked onto the superior source clock; it must continue in holdover mode to supply timing to yet lower-tier clocks.

ITU-T Timing Recommendations

ITU-T recommendations define a somewhat different hierarchy of clock specifications from those of T1.101. At the bottom of the pyramid are clocks conforming to G.823 (for signals belonging to the E1-hierarchy) or G.824 (for T1-hierarchy signals). These standards define the permissible output jitter and wander for two levels, the lower being traffic interface and the more stable being synchronization interface. The former may not be used as a master clock, while the latter may.

- ITU-T Recommendation G.812 defines a slave clock commonly known as a **synchronization supply unit (SSU)**. The SSU's master may be a PRC, another SSU or an SEC (see below). The SSU fulfills two objectives: it filters out jitter and relatively short-term wander, and it provides a highly accurate clock for holdover scenarios. An SSU may be designed as a **Stand Alone Synchronization Equipment (SASE)**, known in North America as a **Building Integrated Timing Source (BITS)** clock, or as part of a traffic handling network element, such as a digital cross-connect. Up to 10 SSUs may be chained without overly degrading performance; when this is done, the intermediate SSUs are called **transit SSUs (SSU-T)**, while the final one is called a **local SSU (SSU-L)**.

ITU-T Recommendation G.813 defines a slave clock known as an **SDH Equipment Clock (SEC)**. A SEC is required to have fairly good, but not excellent, timing accuracy while in holdover mode (thus allowing the use of relatively inexpensive crystal local oscillators), but stringent jitter and wander generation (in order to enable **chaining** of multiple SECs). An SSU can be employed after a chain of SECs to counteract the accumulation of timing inaccuracies.

Measuring Accuracy, Jitter and Wander

We have mentioned several times that standards rigorously constrain the accuracy, jitter and wander of TDM clocks, but have yet to explain how these physical characteristics are measured. **Frequency accuracy** is usually specified as the fractional offset from the desired value, and is only meaningful when the time period over which it is measured is specified. Frequency accuracy is the crucial measure of PRCs, and is the main measure of a timing source's stability during holdover. Accuracy is also specified by standards for cellular transmitter stations, because frequency differences can impair hand-offs between cells. However, accuracy is not usually specified for slave clocks, as their function is to follow the timing of their master, rather than remain close to an absolute timing source.

As jitter is basically an instantaneous phenomenon, its relative magnitude is an adequate measure; in contrast, wander is time-dependent, and thus its specification must take into account the time interval over which it is measured.

Jitter is conventionally measured in **U_{lpp}**, that is, the difference between maximum and minimum time intervals in units of the nominal UI. For example, for an E1 signal with a UI of 488 nanoseconds, if the maximum interval were 500 nanoseconds and the minimum 476, the jitter would be $(500-476)/488 = 0.05$ U_{lpp}.

Specification of wander is more complex than jitter, as differing time intervals must be taken into account. Two measures are commonly used, namely the MTIE and TDEV. **MTIE**(τ) (Maximum Time Interval Error) is defined as the maximum peak-to-peak time variation of the clock measured relative to an ideal timing signal, within observation time τ for all observation times of that length within the measurement period T. If there is a constant frequency offset during the time we measure MTIE, the MTIE will be linear in τ ; if there is a linear frequency drift, the MTIE(τ) will diverge. When we are interested in how well a slave clock follows its master, we measure **MRTIE** (Maximum Relative Time Interval Error), which is like MTIE but with the comparison being made to the master clock, rather than to an ideal timing source.

TDEV (Time Deviation) is similar to MTIE, but rather than specifying the maximum peak-to-peak time error over a time duration τ , we calculate the expected time variation (in nanoseconds). It can be shown that TDEV is directly related to the power spectral density of the phase deviation of the timing signal. If there is a constant frequency offset during the time we measure TDEV, the TDEV is unaffected; if there is a linear frequency drift, the TDEV will behave linearly. TDEV is superior to MTIE when the spectral content of the phase noise of a signal is of interest.

Attaining Timing Goals in TDM Networks

TDM networks utilize several mechanisms in order to guarantee that their timing conforms to the requisite standards. First, the TDM physical layer signals are designed

to ensure that the basic single-bit duration is readily identified, irrespective of the data being transferred. For example, data is scrambled in order to eliminate long runs without transitions, and lines codes such as AMI and HDB3 are employed in order to introduce additional transitions.

Second, TDM slave clocks attenuate jitter and wander that accumulate on their inputs. They do this by using sophisticated circuitry in the Line Interface Unit (LIU). The precise instant that a TDM bit is received equals the time it was transmitted plus the nominal propagation time over the physical channel plus a zero mean stochastic component attributable to temperature, oscillator noise, regenerator jitter, justification effects, etc. In order to eliminate the stochastic component, some sort of averaging process must be carried out to capture the average rate of transmission of the original bit stream. 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, the aforementioned hierarchy of TDM clocks forms a synchronization network, whereby every clock in the TDM network is traceable to the PRC. The various synchronization network elements have carefully designed jitter/wander tolerance, transfer, generation and output characteristics. **Tolerance** refers to the ability of an element to tolerate inaccuracies at its input, while **transfer** describes the cleaning up of input inaccuracies. **Generation** expresses the intrinsic jitter and wander contributed by the element itself, and the **output** jitter/wander are the result of the input deficiencies and all of the above.

Attaining TDM Timing Goals for TDMoIP

TDMoIP is a relatively new technology that enables transport of TDM traffic over **Packet Switched Networks** (PSNs), such as Ethernet, IP and MPLS. The TDM bit-stream is segmented and encapsulated, packets containing TDM payload traverse the PSN, and at the far end the TDM signal must be reconstructed, emulating the original TDM transport.

The major technical barrier to TDM emulation is clock recovery. While isochronous TDM networks inherently deliver timing along with the data, and even ATM networks provide a physical layer clock reference, asynchronous PSNs do not transfer any timing information whatsoever.

As will be explained shortly, matters are made worse due to **Packet Loss (PL)** and **Packet Delay Variation (PDV)**. In order to reconstruct TDM timing, sophisticated clock recovery mechanisms are required in order to achieve the desired timing accuracy in the presence of packet delay variation and packet loss.

TDMoIP packets injected into a PSN at a constant rate reach their destination with delay that has a random component, known as PDV. When emulating TDM on such a network, it is possible to overcome this randomness by placing the TDM into a 'jitter

buffer' from which the data can be read out a constant rate for delivery to TDM end-user equipment. The problem is that the time reference of the TDM source is no longer available, and the precise rate at which the data is to be 'clocked out' of the jitter buffer is hence unknown.

In certain cases, timing may be derived from accurate clocks at both endpoints, for example, if the TDMoIP replaces a link in an otherwise isochronous network, or if atomic clocks or GPS receivers are available at both sides. However, often the only alternative is to attempt to recover the clock based only on the TDMoIP traffic. This is possible since the source TDM device is producing bits at a constant rate determined by its local clock. We receive these bits in packets that suffer PDV that can be considered a zero-mean random process. The task of clock recovery can thus, once again, be seen to be a kind of averaging process that negates the effect of the random PDV and captures the average rate of transmission of the original bit stream. As in the pure TDM case, a PLL is well-suited for this task, but now the jitter and wander are orders of magnitude higher.

One conventional means of clock recovery is based on adapting a local clock based on the level of the receiver's jitter buffer. To understand the operation of the conventional mechanism, let us assume for the moment that there is no PDV but that the local clock is initially lower in frequency than the source clock. The writing into the jitter buffer occurs faster than it is emptied and thus the fill-level starts to rise. This rise is detected and compensated by increasing the frequency of the local clock. When in addition to clock discrepancy there *is* PDV, the jitter buffer level no longer smoothly rises or falls, but rather fluctuates wildly about its slowly changing average level. By using a PLL that locks onto the average rate, any frequency discrepancy between the source and destination clocks is eventually compensated, and the receiver's jitter buffer will settle on the level corresponding to precise frequency alignment between the two clocks.

This conventional PLL has several faults. First, the PLL must observe the sequence of level positions for a long period before it can lock onto the source clock, and hence the scheme exhibits lengthy convergence time. Second, the jitter buffer level may settle down far from its desired position at the buffer center, thus making it vulnerable to overflow and underflow conditions. Alternatively, the jitter buffer size may be increased to lower the probability of underflow/overflow, but such a size increase inevitably brings about an increase in the added latency. Finally, the low resolution of the jitter buffer level leads to unnecessarily high wander generation.

By using advanced clock recovery algorithms, recovered TDM clocks can be made to comply with ITU-T G.823 and G.824 specifications for many PSN configurations. However, frequent congestion and/or reroute events may make it physically impossible for pure adaptive clock recovery to conform to these specifications.