Using Synchronization over PSN – Does IEEE 1588^{TM} Really Make a Difference?

Yaakov (Jonathan) Stein, RAD Data Communications

and

Brian Stroehlein, TranSwitch Corporation

Special thanks to Alon Geva, DSP Algorithm Team Leader at RAD

Abstract

It has been suggested that IEEE 1588, being based on a bidirectional protocol, can detect constant delay changes (CDCs) better than the unidirectional mechanisms employed in pure adaptive timing recovery. We test this suggestion both theoretically and using the new G.8261 test suite and find to the contrary.

I. INTRODUCTION

IEEE 1588 was originally developed as a standard for precision clock synchronization of LAN-networked measurement and control systems. Recently, work has been undertaken by the telecommunications community to enhance IEEE 1588 for frequency and time distribution over metro and wide area packet-switched networks (PSNs), and specifically to enable it to be used for synchronization of circuit emulation services (CES) over PSNs, such as TDMoIP.

Once 1588 is enhanced it will impart certain advantages as compared to straightforward clock recovery based on the circuit emulation flow. The 1588 messages may be used to attain frequency lock before particular CES flows are needed, and thus enabling rapid setup of synchronized CES flows. Furthermore, as the 1588 messages are not linked to a particular flow, they need not be sent at a packet transmission creation rate that may interfere with neighboring TDMoIP flows, nor even at a constant packet transmission rate.

On the other hand, TDMoIP flows have native mechanisms for adaptive clock recovery, and IEEE 1588 adds additional traffic that may not be truly needed. In addition, the aforementioned use of 1588 creates a single point of failure, with any fault in the 1588 or its algorithms potentially affecting the timing of a very large number of TDMoIP flows.

Is the extra bandwidth and risk worthwhile? IEEE 1588 functions essentially as a pure adaptive clock recovery mechanism and thus suffers from the same performance limitations.

It has been suggested that 1588's main strength as compared with pure adaptive clock recovery is its ability to detect constant delay changes introduced by reroute or sudden changes in the loading of the network. This increased ability supposedly derives from its being able to measure the absolute delay that the protocol packets undergo while traversing the path from the remote master to the slave's receiver. G.8261 (formerly G.pactiming) specifies test scenarios for such constant delay changes, and can be used to test this suggestion.

IEEE 1588 performs timing distribution based on a bidirectional protocol. Timestamps are exchanged in both directions from a master node (usually connected to an accurate clock, often a Stratum 1 clock) to a slave node (which performs a clock recovery function) and vice versa. This bidirectional approach allows the slave node to evaluate the absolute delay that the protocol packets (i.e., timestamps) undergo when traversing a path from the remote master to the slave's receiver and therefore to deliver absolute time (referenced to a specific starting point). The IEEE 1588 protocol is designed so that every network element (e.g., a Layer 2 switch or Layer 3 router) can incorporate an IEEE 1588 master/slave pair on every port (e.g., boundary clock), thus allowing a much more controlled network environment in terms of the PDV that an IEEE 1588 timing distribution packet experiences. Achieving this goal, however, requires upgrading all network elements along the timing distribution path to support IEEE 1588.

Protocols for circuit emulation services (CES) like TDMoIP, SAToP, and CESoPSN may also be used for delivering sync over a PSN. These protocols are unidirectional: the packet flow is carried in one direction from the master to the slave only. Therefore, the common CES protocols are only suited for delivering timing (frequency) rather than absolute time.

II. "CONSTANT DELAY CHANGE" EVENTS

There are several types of common disruptive events that may cause sudden changes in the "constant delay" component of a PSN's performance. These changes may be permanent or temporary. These disruptive events include routing changes, sudden changes in network loading, temporary network overload, and temporary loss of service.

At first glance, it would seem that the bidirectional approach taken by IEEE 1588 could allow the slave to better detect constant delay changes (CDCs) associated with disruptive events as compared to a unidirectional approach. In section III we shall demonstrate that the theoretical performance of a unidirectional protocol is actually better than that of a bidirectional protocol. Since carrier-proven unidirectional systems are already available today, there is thus no need to push toward bidirectional systems solely for the sake of CDC detection.

The recently consented ITU-T G.8261 specification ("Timing and Synchronization Aspects in Packet Networks," formerly G.pactiming) has specific test cases to test the ability of a slave node to adequately detect CDCs in order to deliver synchronization of acceptable quality in the real world. In section IV we will present test results of a pure adaptive mechanism, and conclude that it is capable of passing the G.8261 tests without 1588 assistance.

III. ANALYSIS OF UNIDIRECTIONAL AND BIDIRECTIONAL PROTOCOLS

Figure 1 shows the basic system diagram where master and slave nodes are connected via a packet switched network.

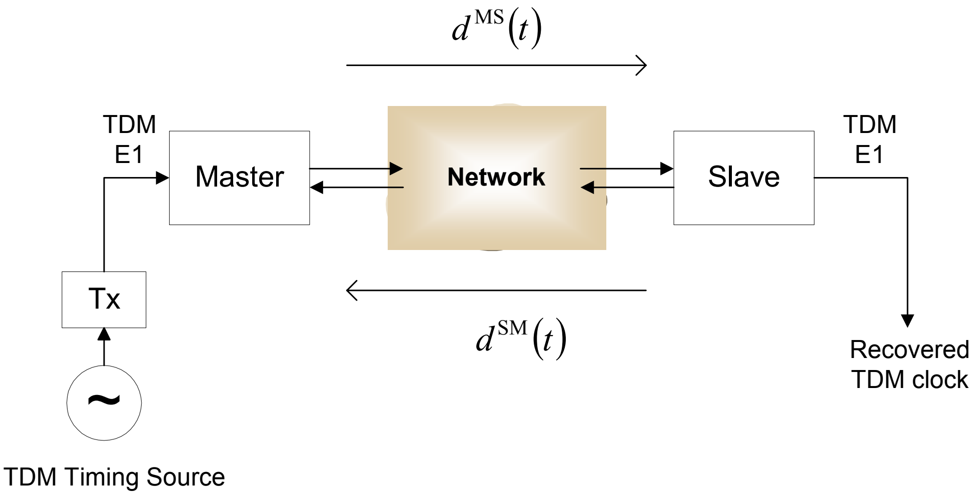

Fig. 1: Basic system diagram

A packet traversing the network from the master to slave will undergo a delay of $d^{\mathrm{MS}}(t)$, which is itself time-dependent, while packets traversing from the slave to master undergo a different delay $d^{\mathrm{SM}}(t)$, which is also time-dependent.

Figure 2 depicts a simplified version of the timestamp exchange protocol used by IEEE 1588.

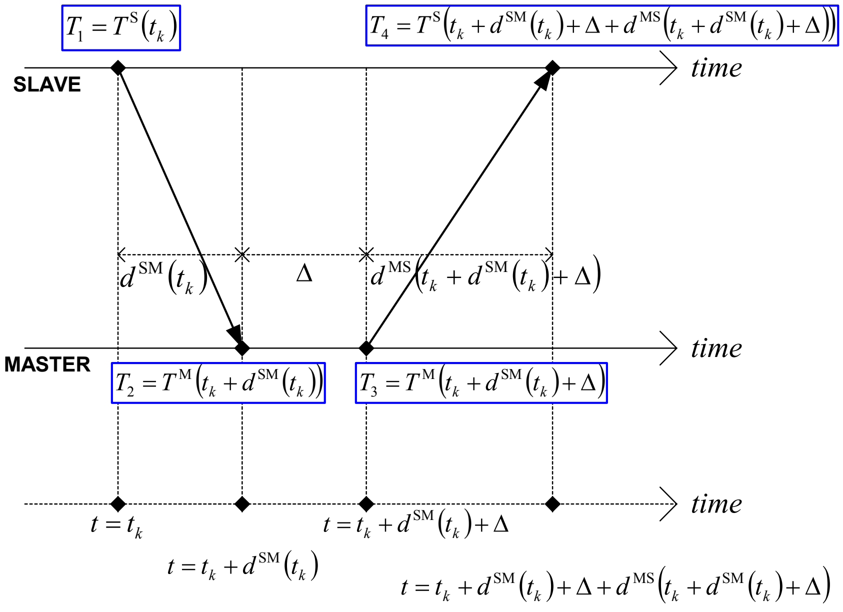
$T_1 = T^{\mathrm{S}}(t_k)$
$T_4 = T^{\mathrm{S}}(t_k + d^{\mathrm{SM}}(t_k) + \Delta + d^{\mathrm{MS}}(t_k + d^{\mathrm{SM}}(t_k) + \Delta))$
SLAVE
time
$d^{\mathrm{SM}}(t_k)$
Δ
$d^{\mathrm{MS}}(t_k + d^{\mathrm{SM}}(t_k) + \Delta)$
MASTER
time
$T_2 = T^{\mathrm{M}}(t_k + d^{\mathrm{SM}}(t_k))$
$T_3 = T^{\mathrm{M}}(t_k + d^{\mathrm{SM}}(t_k) + \Delta)$
time
$t = t_k$
$t = t_k + d^{\mathrm{SM}}(t_k) + \Delta$
$t = t_k + d^{\mathrm{SM}}(t_k)$
$t = t_k + d^{\mathrm{SM}}(t_k) + \Delta + d^{\mathrm{MS}}(t_k + d^{\mathrm{SM}}(t_k) + \Delta)$

Fig. 2: IEEE 1588 timestamp exchange

At time instant $t = t_k$ the slave creates a timestamp of its local time representation $T_1 = T^{\mathrm{S}}(t_k)$, puts it in a packet and sends it to the master. The packet undergoes a time varying delay (packet delay variation, or PDV) denoted by $d^{\mathrm{SM}}(t_k)$. The master receives the packet exactly at time $t = t_k + d^{\mathrm{SM}}(t_k)$ and issues its own timestamp (according to its own local time representation) $T_2 = T^{\mathrm{M}}(t_k + d^{\mathrm{SM}}(t_k))$. The master waits for a time (Δ seconds), then at time instant $t = t_k + d^{\mathrm{SM}}(t_k) + \Delta$ it issues another timestamp (again according to its local time representation) $T_3 = T^{\mathrm{M}}(t_k + d^{\mathrm{SM}}(t_k) + \Delta)$ and sends both T_2 and T_3 back to the slave. This last exchanged packet undergoes a time varying delay of $d^{\mathrm{MS}}(t_k + d^{\mathrm{SM}}(t_k) + \Delta)$ and arrives at the slave at time instant $t = t_k + d^{\mathrm{SM}}(t_k) + \Delta + d^{\mathrm{MS}}(t_k + d^{\mathrm{SM}}(t_k) + \Delta)$. The slave then issues a fourth timestamp: $T_4 = T^{\mathrm{S}}(t_k + d^{\mathrm{SM}}(t_k) + \Delta + d^{\mathrm{MS}}(t_k + d^{\mathrm{SM}}(t_k) + \Delta))$.

In order to simplify the calculation we can make a few common assumptions at this point:

1) The slave's PLL has achieved lock so there is a (almost) zero frequency error between the master and the slave. Hence, the relationship between the local time representation of the slave and that of the master is $T^{\mathrm{S}}(t) = T^{\mathrm{M}}(t + \varepsilon)$, where ε is a constant uncompensated phase (time) error between the master and the slave.
2) The master's clock is locked to a primary reference clock (PRC or Stratum 1 clock). Hence the time representation of the master can be simplified to: $T^{\mathrm{M}}(t) \cong t$.

Given these assumptions we can rewrite the four timestamps of Figure 2 in a simpler manner:

$$
\begin{aligned}
T_1(k) &= t_k + \varepsilon \\
T_2(k) &= t_k + d^{\mathrm{SM}}(k) \\
T_3(k) &= t_k + d^{\mathrm{SM}}(k) + \Delta \\
T_4(k) &= t_k + d^{\mathrm{SM}}(k) + \Delta + d^{\mathrm{MS}}(k) + \varepsilon
\end{aligned}
$$

Where k is the packet's sequence number. At this last step, we also set $d^{\mathrm{SM}}(t_k), d^{\mathrm{MS}}(t_k + d^{\mathrm{SM}}(t_k) + \Delta)$ to be $d^{\mathrm{SM}}(k), d^{\mathrm{MS}}(k)$ respectively.

In order to calculate the current master-slave end-to-end delay the slave performs the following calculation:

$$\tilde{d}^{\mathrm{MS}}_{\mathrm{2way}}(k) = \frac{T_4(k) - T_3(k) + T_2(k) - T_1(k)}{2} = \frac{d^{\mathrm{SM}}(k) + d^{\mathrm{MS}}(k)}{2}$$

In general $d^{\mathrm{SM}}(k), d^{\mathrm{MS}}(k)$ would be characterized as uncorrelated stochastic processes having the following means and standard deviations:

$$\mu^{\mathrm{SM}} = E\left[d^{\mathrm{SM}}(k)\right]$$

$$\sigma^{\mathrm{SM}} = \sqrt{E\left[\left(d^{\mathrm{SM}}(k)\right)^2\right] - \left(\mu^{\mathrm{SM}}\right)^2}$$

$$\mu^{\mathrm{MS}} = E\left[d^{\mathrm{MS}}(k)\right]$$

$$\sigma^{\mathrm{MS}} = \sqrt{E\left[\left(d^{\mathrm{MS}}(k)\right)^2\right] - \left(\mu^{\mathrm{MS}}\right)^2}$$

Where E is the expectation value operator. Hence, the total amount of noise power in the slave's end-to-end estimate $\tilde{d}^{\mathrm{MS}}_{\mathrm{2way}}(k)$ would be $(\sigma^{\mathrm{SM}^2} + \sigma^{\mathrm{MS}^2})/2$ and, as a consequence, the detection of CDCs would have to overcome this noise level.

On the other hand, a unidirectional timing distribution protocol (e.g. that used by pure adaptive clock recovery) would use a unidirectional delay estimate for CDC detection:

$$\tilde{d}^{\mathrm{MS}}_{\mathrm{1way}}(k) = T_4(k) - T_3(k) = d^{\mathrm{MS}}(k) + \varepsilon\ .$$

Since ε is nearly constant, the total amount of noise power in the slave's end-to-end estimate $\tilde{d}^{\mathrm{MS}}_{\mathrm{1way}}(k)$ is only σ^{MS^2}.

Now suppose that $d^{\mathrm{MS}}(k)$ changes by d_0 seconds. The respective SNRs for detecting this CDC event will be:

$$SNR_{1way} = {d_0}\Big/{\left(\sigma^{\mathrm{MS}^2}\right)}$$

$$SNR_{2way} = {(d_0 / 2)}\Big/{\left((\sigma^{\mathrm{SM}\,2} + \sigma^{\mathrm{MS}\,2}) / 2\right)}$$

$$= {d_0}\Big/{\left(\sigma^{\mathrm{SM}\,2} + \sigma^{\mathrm{MS}\,2}\right)}$$

Thus, for CDC detection, the unidirectional protocol has roughly twice the SNR of the bidirectional protocol given the same noise level for $d^{\mathrm{SM}}(k), d^{\mathrm{MS}}(k)$!

IV. PERFORMANCE TESTING OF A UNIDIRECTIONAL PROTOCOL

In this section we present recently conducted laboratory tests of our TDMoIP-based unidirectional sync transmission using adaptive clock recovery at a slave node equipped with an OCXO reference clock. RTP was not used. These tests were performed according to test Cases 1, 2 and 3 of Appendix VI of the recently consented ITU-T G.8261 (formerly G.pactiming). For all tests network Traffic Model 1 (see VI.2.2.1.1/G.8261) was used, and QoS enhancing mechanisms were not employed.

Our network was composed of five Gigabit Ethernet switches in series. For Traffic Model 1 the packet size profile is:

- 80% of the load is minimum size packets (64 octets)
- 15% of the load is maximum size packets (1518 octets)
- 5% of the load is medium size packets (576 octets)

Maximum size packets occur in bursts lasting between 0.1s and 3s.

G.8261 further differentiates between three test cases. In all three cases a stabilization period is allowed before performing the measurements, and the packets used to load the network are generated according to a specific network traffic model specified in the Recommendation.

Test Case 1 (VI.2.2.2/G.8261)

Here a static load is modeled. Network disturbance load is maintained at 80% for one hour assuming that the clock recovery is in a stable condition.

Test Case 2 (VI.2.2.3/G.8261)

This case models sudden large and persistent changes in network loading, i.e. CDCs. It tests stability under sudden changes in network conditions, and wander performance in the presence of low frequency PDV. The network disturbance load alternates between 80% for an hour and 20% for an hour.

Test Case 3 (VI.2.2.4/G.8261)

This case models the slow change in network load over an extremely long timescale. It tests stability under very slow changes in network conditions, and wander performance in the presence of extremely low frequency PDV. The network disturbance load varies smoothly from 20% to 80% and back over a 24-hour period.

Our MTIE test results are presented in Figure 3. We see that that timing delivered using adaptive clock recovery falls well within the limits for the common Deployment Cases anticipated by G.8261.

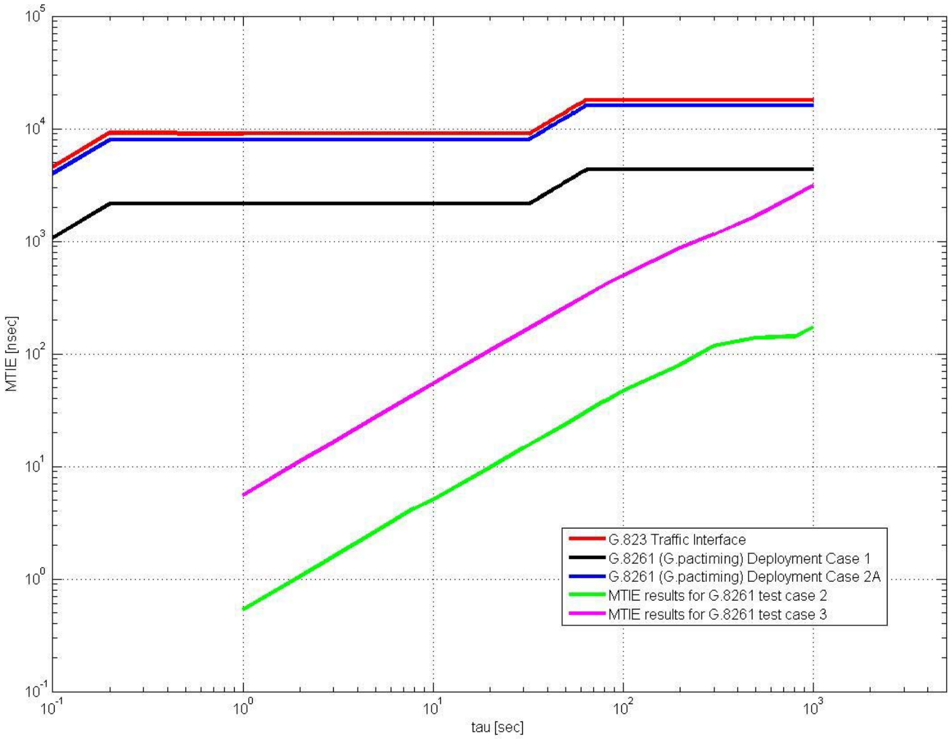

Figure 3 – MTIE tests results for test cases 2 and 3 / G.8261

V. CONCLUSION

While bidirectional protocols like IEEE 1588 allow slave nodes to evaluate the absolute delay the protocol packets (i.e., timestamps) undergo when traversing the path from the remote master to the slave's receiver and therefore deliver absolute time, they are also inherently noisier than unidirectional protocols, thus less well-suited for detecting disruptive events which cause CDCs. The IEEE 1588 solution also requires that almost every network element along the timing distribution path support IEEE 1588, which means a very unappealing wholesale upgrade for carriers. Unidirectional protocols for circuit emulation services (CES) like TDMoIP, SAToP, and CESoPSN are more than sufficient for delivering sync over a PSN.