

IEEE 1588 FREQUENCY AND TIME TRANSFER MEASUREMENTS AND ANALYSIS: CLOCK, PDV, AND LOAD

Lee Cosart
R&D, Symmetricom, Inc.
2300 Orchard Parkway
San Jose, CA 95131, USA
lcosart@symmetricom.com

Abstract

The IEEE 1588 protocol provides a means of transporting both frequency and time. In order to gain insight into the network conditions under which frequency and time transfer algorithms must perform and to assess the performance of 1588 devices, three types of measurements have proven to be useful: (1) measuring the clock signal at the output, (2) making network timing measurements on the packets, and (3) dynamic characterization of the load. This paper describes these three measurements and explores their relationship.

This paper starts with a description of IEEE 1588 packet measurement equipment, and then turns to a discussion of the one-way and two-way metrics connected with packet frequency and time transfer. Laboratory and field measurement results will be used to illustrate the described measurement and analysis techniques. The paper then will establish links between packet timing measurements and traditional clock measurements made at the output of 1588 client/slave devices. Finally, techniques of measuring dynamic network load with a new prototype hardware device will be introduced, along with a discussion of the links between load and packet delay variation.

INTRODUCTION

As originally conceived, IEEE 1588 deployments were envisioned for a local area network comprised of a limited number of layer 2 switches. As the telecom industry began to explore IEEE 1588 for its own applications, larger networks of a more diverse composition were considered. With this increased network complexity comes less hospitable conditions for the slave device, which must recover precision time and frequency across the network.

One of the challenges of using IEEE 1588 in telecom is overcoming the effects of packet delay variation. Thus, the study of packet delay variation has proven to be essential to the development of optimal slave algorithms. Packet delay variation is itself greatly affected by network load, so it is reasonable to surmise that a knowledge of dynamic network loading conditions can lead to insight into the characteristics of packet delay variation and vice versa.

On-path support in the form of boundary clocks or transparent clocks is one method of addressing the need of precision time and frequency delivery in complex telecom networks, but in those situations where the network can provide limited or no on-path support, the slave must be designed to cope with the packet delay variation (PDV) that results from these networks.

Consequently, the measurement and analysis of packet delay variation is important for several reasons. First, understanding how PDV can behave over a range of network configurations and conditions is critical for producing the best possible performance from a slave through the design of advanced slave algorithms. Second, these measurements can provide guidance for the optimal deployment of masters and slaves in the network. Third, monitoring PDV within an IEEE 1588 element can provide information on slave performance as well as network conditions more generally.

The network packet delay variation is related to the network configuration, including such aspects as number of hops and type of network equipment. Switches, routers, microwave radio links, and DSLAM's all behave differently, as do different models of the same type of device, whether or not from the same manufacturer.

The load through a device and through a network composed of many devices is a critical factor for packet delay variation and packet latency; with increased load, both PDV and latency have a tendency to increase. Thus, measuring dynamic packet network load directly can provide insight into PDV, just like the measurement of PDV can provide insight into packet slave performance.

For the measurement of latency and packet delay variation, the 1588 packets themselves can function as probe packets, with time stamps taken on a particular packet as it passes two points in a network. The basic data set is formed as a sequence of packet transit delay values, with each sample computed as differences between the two time stamps. The packet delay sequence is akin to the phase measurement taken on a timing signal such as an oscillator output, and the analysis techniques for studying packet delay sequences are informed by those applied to the study of oscillator stability, particularly for the case of packet frequency transfer.

For studying packet frequency transfer, the quantity of greatest interest is packet delay variation, as opposed to absolute packet latency. With this focus on packet delay variation, the time stamping equipment at the two network nodes need not share a common time scale. It is only necessary that the clocks move at the same rate, so frequency locking the time stamping equipment to stable sources suffices. Stability metrics, particularly those based on Allan deviation, have proven useful as analysis tools, but packet selection has become an important component of the analysis.

For studying packet time transfer, the time stamping measurement equipment must share a common time scale, such as that provided by GPS. This is because both downstream and upstream paths must be measured and analyzed simultaneously, and, in particular, symmetry between the two paths needs to be assessed. The single-path stability metrics applied to packet frequency transfer are still informative here, but additional metrics focused on the combination of the forward and reverse paths are needed.

MEASURING CLOCKS, PACKET DELAY VARIATION, AND NETWORK LOAD

In comparison to the measurement of clock stability, the measurement of packet transit delay differs in several important ways. The measurement of clock signals involves timestamping signal edges at a single point, while the measurement of packet delay variation involves timestamping packets at two points in a

network. The traditional clock measurement is illustrated in Figure 1. An instrument with a precision phase detector is locked to a primary reference clock, which enables it to timestamp the edges of an input signal precisely. Several samples of a hypothetical measurement made on a 10 MHz signal are shown. Nominally the edges should be spaced by 100-ns intervals, but because of jitter and wander on the signal, they vary slightly from this. The second edge arrives 0.1 ns late, the third edge arrives 0.3 ns early, and the fourth edge arrives 0.5 ns late.

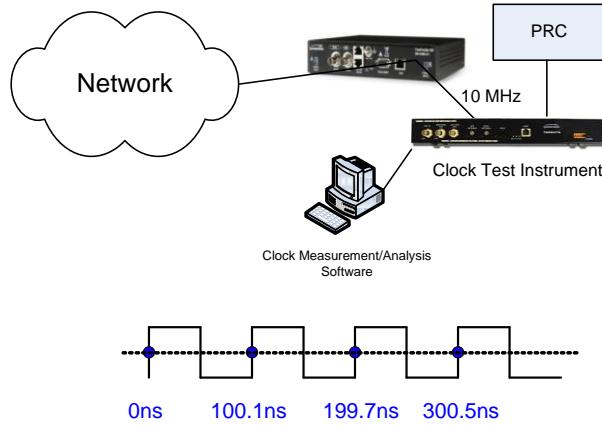
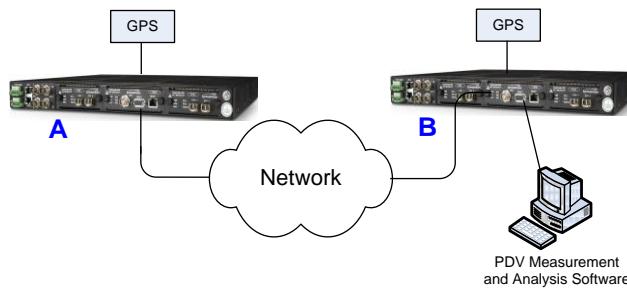


Figure 1. Setup for measuring a clock.

The setup for measuring packet transit delay, as shown in Figure 2 below, requires two sets of timestamping equipment and two primary reference clocks, ideally set to the same time scale. A frequency primary reference suffices for the clock measurement, but in order to determine how long it takes a particular packet to traverse a network, clocks with a common time scale at both ends are necessary. Two GPS-based primary references can fulfill this requirement. Further, it is important that packet delay is studied in both forward and reverse directions. Thus, there are two sets of timestamp pairs produced, one in the forward direction, and one in the reverse direction. This is shown in the “F” (forward) and “R” (reverse) timestamp pairs shown in Figure 2.



	Timestamp A	Timestamp B
F	1233166476.991204496	1233166476.991389744
R	1233166476.980521740	1233166476.980352932
F	1233166477.006829496	1233166477.007014512
R	1233166476.996147084	1233166476.995977932
F	1233166477.022454496	1233166477.022639568
R	1233166477.011771820	1233166477.011602932

Figure 2. Packet transit delay measurement setup.

Just like a clock measurement involves producing a set of samples from timestamps taken at signal edges, a packet delay measurement involves producing a set of samples from timestamp pair differences, with a sequence of packets providing those samples. The rate at which probe packets are produced, say for example 64 Hz, dictates the packet delay sequence sample rate.

Packet delay variation has a direct impact on packet slave performance, and packet delay variation is itself greatly affected by network load. It is, therefore, instructive to study these three simultaneously if possible. Figure 3 below shows a measurement setup where a 1588 slave 10 MHz output, the packet delay variation seen at the input to the 1588 slave, and network load are measured at the same time. The first two of these measurements have been described above, and at the end of this paper, the measurement of dynamic network load will be described in detail.

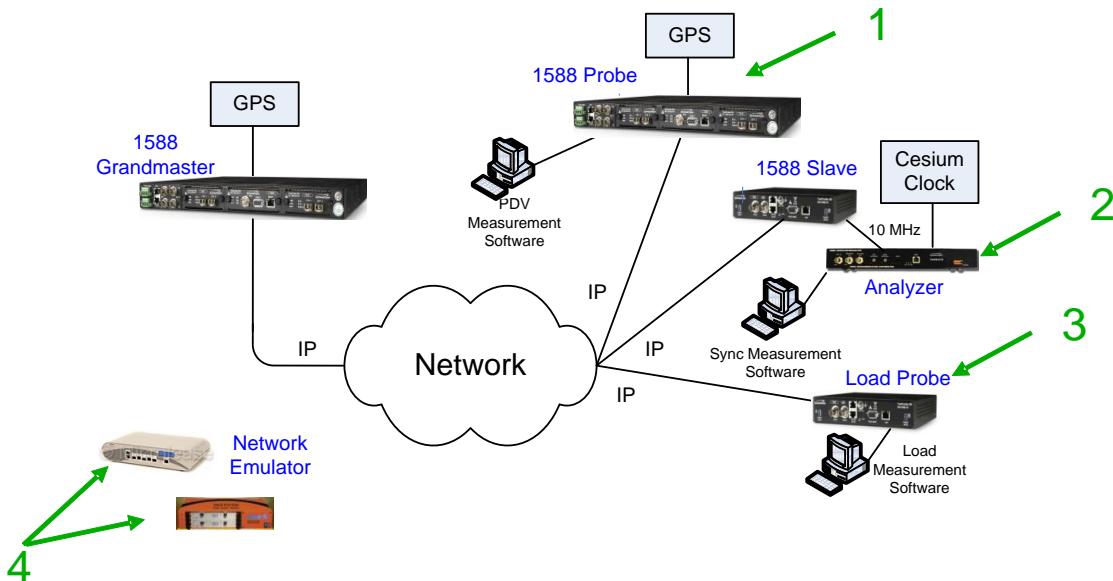


Figure 3. Setup for measuring slave clock output, packet delay variation, and network load.

Figure 3 also shows a fourth kind of equipment, the network emulator. One powerful function available on a network emulator is the ability to reproduce exactly a packet transit delay measurement made in the field or the lab. Thus, it is useful complementary equipment to a probe designed to measure packet delay variation and latency. It facilitates slave clock algorithm development and characterization by enabling the playback of field or lab PDV capture repeatedly and exactly.

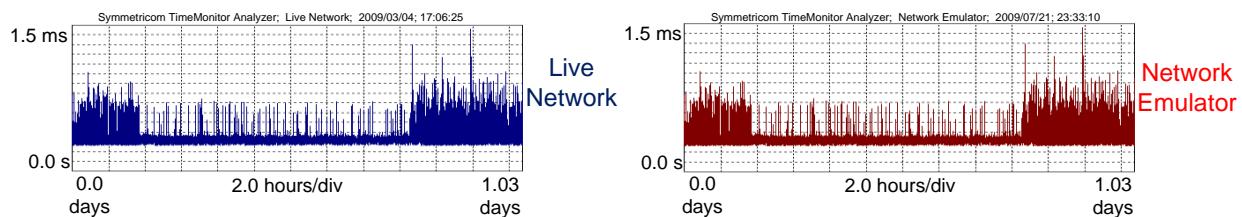


Figure 4. Original IEEE 1588 probe live production network measurement (blue) and network emulator measurement (red) with network emulator playback-based original live network measurement.

Figure 4 shows the measurement of packet transit delay taken in a production network in the field and a measurement of the network emulator playback of that capture in the lab. A comparison of the two plots shows very good agreement between the original measurement and the lab playback.

PACKET TIMING MEASUREMENT ANALYSIS OVERVIEW

Much of the attention with regard to packet timing stability analysis thus far has focused on the transport of frequency. Some of analytical tools carry over from clock analysis to packet timing analysis, in some cases with modification. Other approaches that have less applicability to clock analysis have shown utility in the analysis of packet delay variation. The following list compares and contrasts the approaches to clock analysis and packet timing data analysis:

- Clock data analysis
 - Phase
 - Frequency
 - Phase noise
 - MTIE
 - ADEV/MDEV/TDEV
- Packet data analysis
 - Phase (packet delay sequence)
 - Phase noise
 - Histogram/PDF(probability density function) & statistics
 - CDF (cumulative distribution function)
 - Running statistics
 - TDEV/minTDEV/bandTDEV
 - Two-way metrics: minTDISP etc.

The last item on the packet data analysis list above can be distinguished from the others in that it is focused on transport of time rather than frequency. Following a discussion of the other metrics that are oriented towards the transport of frequency, this new category of metrics will be discussed.

The packet data analysis metrics have been under study with the synchronization experts groups both at the ITU-T (Q13/SG15) and ATIS (COAST-SYNC). There are definitions and descriptions of the packet timing metrics in the new ATIS technical report [1] with discussion of the selection methods in both this ATIS document and in the appendix of the new ITU-T packet network definitions document [2].

The quantity of interest for the study of one-way packet delay is a sequence of samples of packet transit delay. Each of these samples represents a particular packet timestamped at two points in the network. The process of forming this packet transit delay sequence from timestamp pairs is illustrated below in Figure 5.

One of the differences between a clock signal measurement and a packet delay measurement manifests itself in the phase data (packet delay sequence), particularly when longer-term measurements are made. Packet delay phase sequences are typically dominated by short-term variations, and a plot of packet delay phase often appears as a band of data. Often a scatter plot view reveals aspects of the data not seen when the data points are connected [3]. Such is the case for the example shown in Figure 6.

Packet Delay Sequence

```

R,00162; 1223305830.478035356; 1223305830.474701511
F,00167; 1223305830.488078908; 1223305830.490552012
R,00163; 1223305830.492882604; 1223305830.489969511
F,00168; 1223305830.503473436; 1223305830.505803244
R,00164; 1223305830.508647148; 1223305830.505821031
F,00169; 1223305830.519029300; 1223305830.521302172
R,00165; 1223305830.524413852; 1223305830.521446071
F,00170; 1223305830.534542972; 1223305830.536801164
R,00166; 1223305830.540181132; 1223305830.537115991
F,00171; 1223305830.550229692; 1223305830.552551628

```

Forward	Reverse
#Start: 2009/10/06 15:10:30	#Start: 2009/10/06 15:10:30
0.0000, 2.473E-3	0.0000, 3.334E-3
0.0155, 2.330E-3	0.0153, 2.913E-3
0.0312, 2.273E-3	0.0311, 2.826E-3
0.0467, 2.258E-3	0.0467, 2.968E-3
0.0623, 2.322E-3	0.0624, 3.065E-3

Figure 5. Packet time-stamp pairs and the corresponding forward and reverse packet delay sequences.

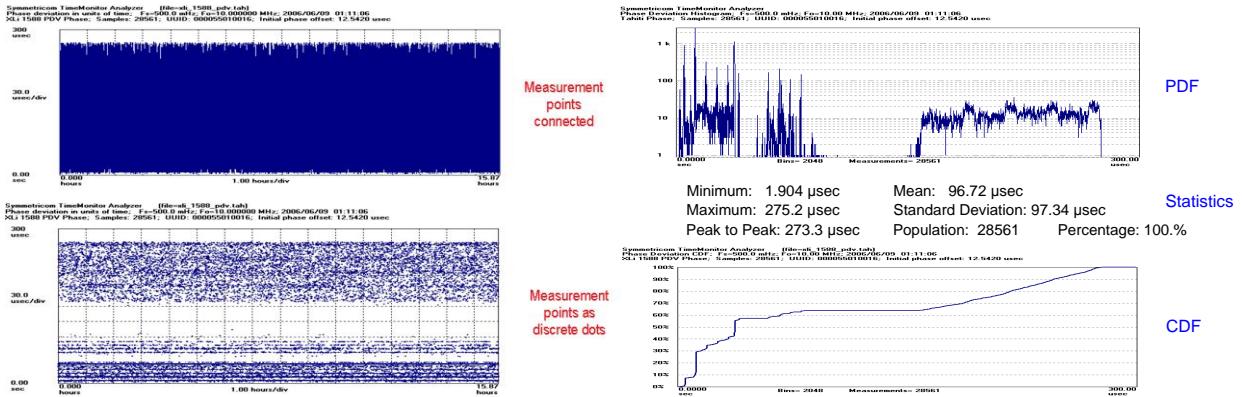


Figure 6. Packet delay sequence with points connected and as scatter plot, histogram (PDF), corresponding statistics, and CDF formed from the packet delay sequence.

If the data are used to form a histogram, which represents the probability density function (PDF), the result is as shown in the plot labeled PDF. Statistics based on this histogram are also shown in the figure. The histogram can then be used as the basis for producing a cumulative distribution function (CDF), also shown in Figure 6. In cases where non-stationarity applies, it may be instructive to track a statistic over time rather than simply histogram all the data and compute a statistic for the whole set of data.

PACKET FREQUENCY TRANSPORT METRICS

Both packet clock algorithms and packet timing analysis include packet selection as a means of optimizing stability given the conditions that exist in packet networks, more specifically the kinds of packet delay variation encountered. Under all but the most benign network conditions some packets – perhaps very few but nevertheless some – take an inordinate amount of time to traverse the network. These outliers are best removed by the packet clock algorithm, and in order to perform analysis that would indicate how well such an algorithm should perform, the calculations need to include a similar operation.

Two different approaches have been adopted for packet selection for packet timing analysis. In the first, referred to as *pre-processed packet selection*, the packet selection is done prior to the stability calculation, which is then performed in the traditional way. For example, if x represents a packet delay sequence, a new packet delay sequence x' could be constructed by searching for minimums over some window of time. These time windows can be either overlapping or non-overlapping. If the x sequence was 1,5,4,3,7,8,2,5,9 and a non-overlapping window of length three is chosen, the x' sequence would be 1,3,2. A standard stability calculation such as TDEV can then be applied to the x' sequence, i.e. $TDEV(x')$.

A second approach, referred to as *integrated packet selection*, changes a standard calculation by incorporating packet selection into the calculation. An example of this is to replace the averaging operation present within the TDEV calculation with a minimum selection process forming a new calculation, i.e. $\text{minTDEV}(x)$.

A variety of selection methods are possible, including the aforementioned *minimum* selection, and others such as *percentile*, which averages together a population of packet delay values at the floor, and *band*, which averages together packet delay values in a region that could be away from the floor. The *percentile* and the *band* methods both involve sorting the data in a window from minimum to maximum and then selecting a region of the data for averaging. Data might be selected between the 10 and 20 percentile and averaged, for example. These three packet selection methods are summarized along with brief descriptions of the *pre-processed* and *integrated* approaches to packet selection in Figure 7.

Packet Selection Processes

- 1) Pre-processed:** packet selection step prior to calculation
 - Example: $\text{TDEV}(\text{PDV}_{\text{min}})$ where PDV_{min} is a new sequence based on minimum searches on the original PDV sequence
- 2) Integrated:** packet selection integrated into calculation
 - Example: $\text{minTDEV}(\text{PDV})$

Packet Selection Methods

- Minimum: $x_{\text{min}}(i) = \min[x_j] \text{ for } (i \leq j \leq i+n-1)$
- Percentile: $x'_{\text{pct_mean}}(i) = \frac{1}{m} \sum_{j=0}^b x'_{j+i}$
- Band: $x'_{\text{band_mean}}(i) = \frac{1}{m} \sum_{j=a}^b x'_{j+i}$

Figure 7. Packet selection approaches and methods.

If these packet selection methods are integrated into calculations, computations such as minTDEV, bandTDEV, and minMAFE are produced. Figure 8 shows examples of these calculations.

TDEV	$\sigma_x(\tau) = TDEV(\tau) = \sqrt{\frac{1}{6n^2} \left\langle \left[\sum_{i=1}^n (x_{i+2n} - 2x_{i+n} + x_i) \right]^2 \right\rangle}$
minTDEV	$\sigma_{x_{\min}}(\tau) = \min TDEV(\tau) = \sqrt{\frac{1}{6} \left\langle [x_{\min}(i+2n) - 2x_{\min}(i+n) + x_{\min}(i)]^2 \right\rangle}$ where $x_{\min}(i) = \min[x_j] \text{ for } (i \leq j \leq i+n-1)$
bandTDEV	$\sigma_{x_{band}}(\tau) = bandTDEV(\tau) = \sqrt{\frac{1}{6} \left\langle [x'_{band_mean}(i+2n) - 2x'_{band_mean}(i+n) + x'_{band_mean}(i)]^2 \right\rangle}$ where $x'_{band_mean}(i) = \frac{1}{m} \sum_{j=a}^{i+n-1} x'_{j+i}$
MATIE	$MATIE(n\tau_0) \equiv \max_{1 \leq k \leq N-2n+1} \frac{1}{n} \left \sum_{i=k}^{n+k-1} (x_{i+n} - x_i) \right , n = 1, 2, \dots, \text{integer part (N/2)}$
MAFE	$MAFE(n\tau_0) \equiv \frac{\max_{1 \leq k \leq N-2n+1} \frac{1}{n} \left \sum_{i=k}^{n+k-1} (x_{i+n} - x_i) \right }{n\tau_0}, n = 1, 2, \dots, \text{integer part (N/2)}$
minMAFE	$\min MAFE(n\tau_0) \equiv \frac{\max_{1 \leq k \leq N-2n+1} \left \sum_{i=k}^{n+k-1} (x_{\min}(i+n) - x_{\min}(i)) \right }{n\tau_0}, n = 1, 2, \dots, \text{integer part (N/2)},$ where $x_{\min}(i) = \min[x_j] \text{ for } (i \leq j \leq i+n-1)$

Figure 8. Standard stability calculations and some of their packet timing stability counterparts.

It is interesting to note the relationship between some of these calculations. The bandTDEV calculation reduces to minTDEV, TDEV, and to percentileTDEV, in certain special cases.

- TDEV is bandTDEV(0.0 to 1.0)
- minTDEV is bandTDEV(0.0 to 0.0)
- percentileTDEV is bandTDEV(0.0 to B) with B between 0.0 and 1.0.

Finally, a view of one of these packet stability calculations applied to real data is informative. Figure 9 shows both TDEV and minTDEV calculations applied to network data at various loads.

Referring to Figure 9, while noise levels show a consistent increase for the TDEV calculation, for the minTDEV calculation, all the plots converge for tau values greater than about 100 seconds. This indicates that a packet slave device with an oscillator of sufficient stability to allow time constants of several hundred seconds could, with a minimum transit delay selecting algorithm, perform nearly as well at 50% load as it does with no load.

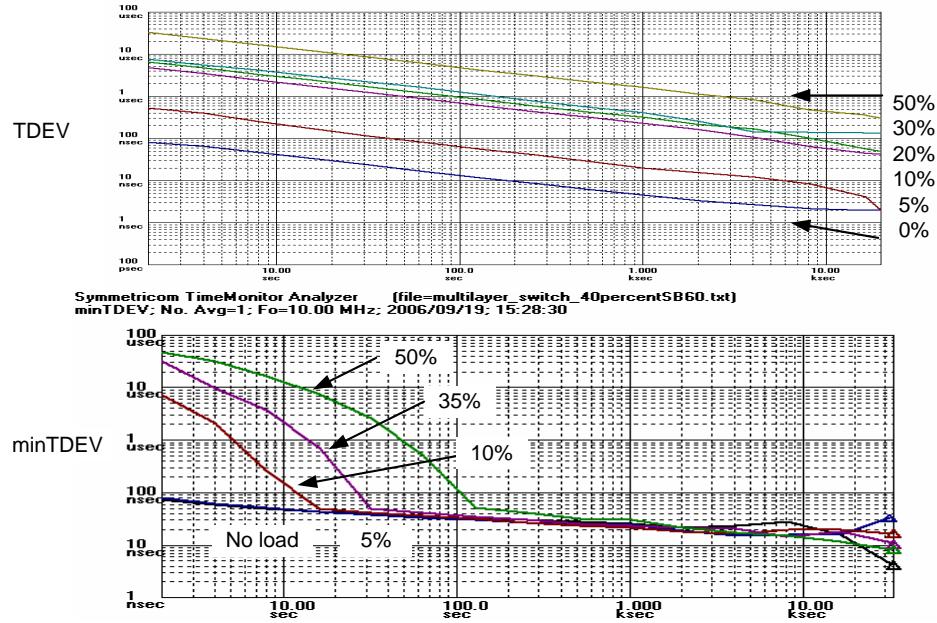


Figure 9. Standard stability calculations and some of their packet timing stability counterparts.

PACKET TIME TRANSPORT METRICS

Achieving the transport of time using a packet protocol requires and fully exploits a two-way packet timing protocol. Likewise, simply analyzing a single one-way packet delay sequence is insufficient for time transport analysis. The first step in packet time transport analysis is to combine forward and reverse one-way data into a single structured set of data. This is illustrated in Figure 10, where forward and reverse timestamp pairs are first used to construct forward and reverse packet delay sequences, which are then combined into forward/reverse one-way packet delay pairs, and which are each associated with a particular timestamp.

Figure 10 also shows the formation of a modified two-way sequence by subjecting both forward and reverse packet delay values to a minimum window search. In this example, the window is comprised of three values of which the minimum is determined. Minimum searches of the first three raw forward values of 1.47, 1.54, and 1.23 and first three raw reverse values of 1.11, 1.09, and 1.12 produces 1.23 and 1.09 respectively.

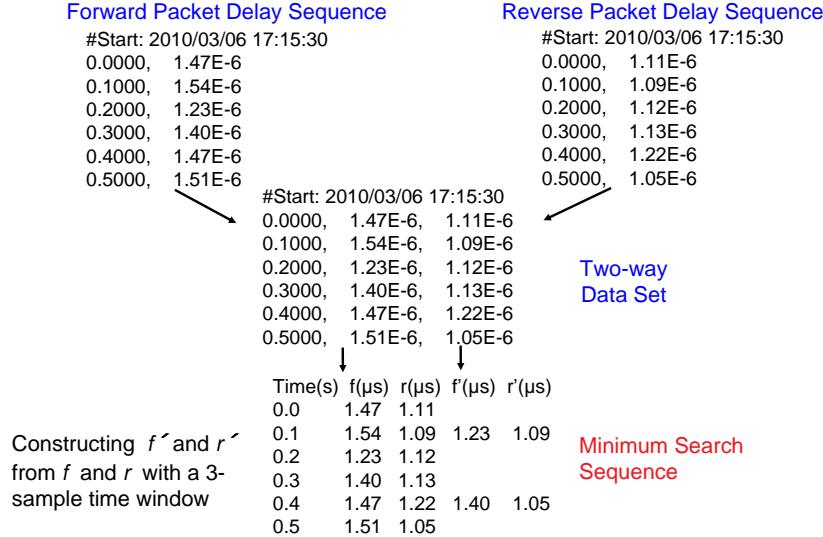


Figure 10. Forming a two-way data set which is then the basis for a modified two-way data set from windowed minimum searches.

These two-way sequences are used to derive among other things roundtrip and offset calculations. These and other two-way calculations are shown in Figure 11. The roundtrip calculation is based on the sum of forward and reverse one-way delay, and the offset calculation represents the difference between forward one-way delay and reverse one-way delay. It is useful to normalize the round trip and offset calculations so that they represent these quantities from the standpoint of a slave/client device which must derive time from one-way delay between itself and the master/server. This normalization is accomplished by performing a division by two.

$$\text{Normalized roundtrip: } r(n) = \left(\frac{1}{2}\right) \cdot [F(n) + R(n)]$$

$$\text{Normalized offset: } \eta_2(n) = \left(\frac{1}{2}\right) \cdot [F(n) - R(n)]$$

$$\text{minRoundtrip: } r'(n') = \left(\frac{1}{2}\right) \cdot [F'(n') + R'(n')]$$

$$\text{minOffset: } \eta_2'(n') = \left(\frac{1}{2}\right) \cdot [F'(n') - R'(n')]$$

minTDISP (minimum time dispersion): minOffset (y) plotted against minRoundtrip (x) as a scatter plot

minOffset statistics: minOffset statistic such as mean, standard deviation, or 95 percentile plotted as a function of time window tau

Figure 11. Time transport metrics, which are based on two-way forward/reverse packet delay sequences.

The *minRoundtrip* and *minOffset* calculations are the same as normalized roundtrip and normalized offset except they are based on the minimum search sequences rather than the raw forward and reverse one-way

delay sequences. The *minTDISP* (minimum time dispersion) calculation plots *minOffset* against *minRoundtrip* as a scatter plot. An example of this is shown in Figure 12. Examples of some other two-way metrics, the *minOffset statistics* calculations are also shown in Figure 12. They are calculated by forming separate *minOffset* sequences using different tau window values, computing a particular statistic such as mean or standard deviation on the particular *minOffset*(τ) sequence, and then plotting these against window tau τ .

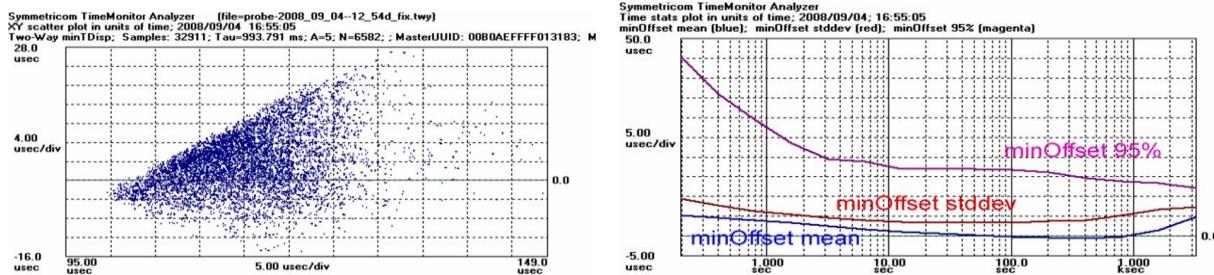


Figure 12. Two-way minimum time dispersion (minTDISP) and minimum offset statistics vs. τ .

In Figure 13, a minTDISP plot is shown based on a measurement of Ethernet wireless backhaul where asymmetry existed between the downstream and upstream paths. The apex of the minTDISP plot, which is where it converges at the value of minimum roundtrip, indicates an offset of -2 μ s. A measurement performed on a packet slave at the same time and, thus, subject to these conditions shows a bias of 2 μ s, as would be expected.

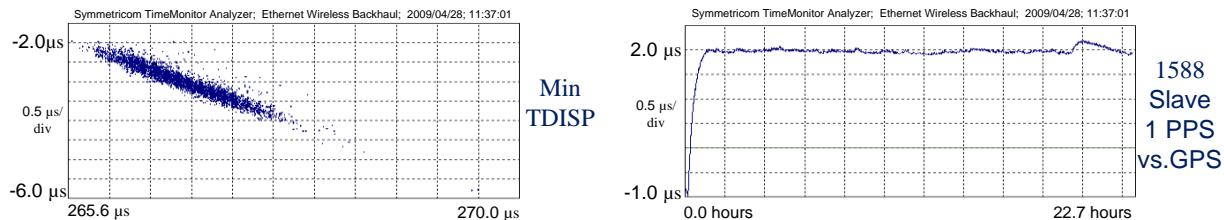


Figure 13. Ethernet wireless backhaul asymmetry and IEEE 1588 slave 1PPS under these asymmetrical network conditions.

CORRELATING PACKET DELAY ANALYSIS WITH PACKET CLOCK PERFORMANCE

The packet timing metrics discussed above provide insight into the packet timing characteristics of a network. Moreover, as the following two examples show, they can be effective predictors of packet slave performance. Both of these examples show strong correlation between the PDV metrics and packet slave timing performance.

(1) PDV AND FREQUENCY OFFSET

In this first measurement example, packet delay variation and the output of four slaves were measured simultaneously. The four slaves and PDV probe were all connected to the same network node during the measurement. The PDV data were processed with a selection of packets focused on minimum transit delay into an MATIE calculation. This is shown in the upper graph in Figure 14, which also includes a straight line derived with a 1PPB offset target in mind using knowledge of the characteristics of a slave algorithm.

The MATIE calculation for this particular set of network PDV data comes very close to this straight line. In the lower graph, the measurements of the outputs of the four slaves are shown along with a 1PPB MTIE network limit (the plot with connected dots); the segment with a constant positive slope in the right half of the plot represents a frequency offset of exactly 1PPB. Note that the four MTIE plots from the four slave measurements cluster around this 1PPB network limit segment.

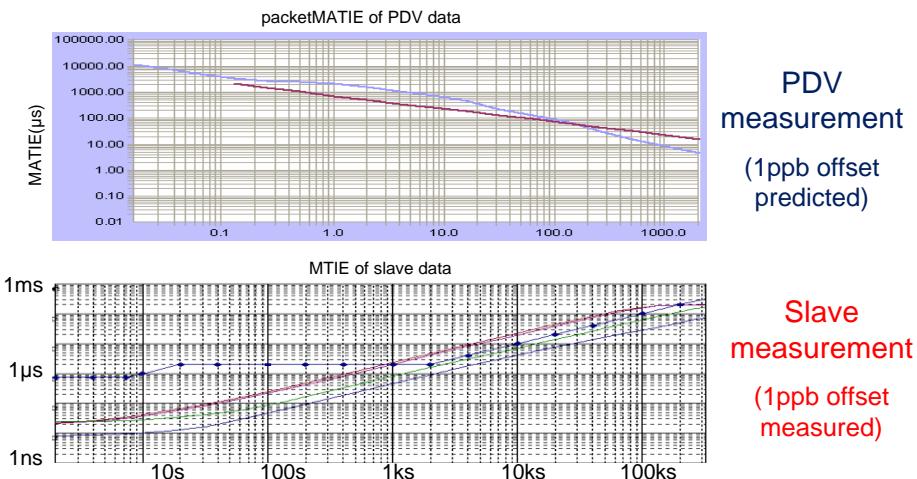


Figure 14. PDV measurement predicts 1ppb offset.

(2) PDV OVER VARYING LOAD AND SLAVE STABILITY PERFORMANCE

In a second experiment, network load was varied while testing PDV and the performance of two slaves. In this case, the slaves were from two different manufacturers and were subject to identical network conditions (connected to the same network node and measured simultaneously). The results are shown in Figure 15. For both slaves, peak MDEV was calculated and plotted against the minTDEV calculation performed on the PDV data.

The individual measurement points, representing load ranging from 20% on a five-node network to 80% on a ten-node network, are shown as blue diamonds for one of the slaves and as red squares for the other slave. A cursory glance at the data reveals the linear relationship that exists in both cases. Performing linear regression on both sets of data shows a correlation of 97% (square root of the R^2 term shown in the figure) or better for both. PDV metrics are shown in this case to be effective predictors of PTP client

performance. It is also worth noting that the Vendor X client is better than the Vendor A client; the lower slope indicates greater immunity to increased load.

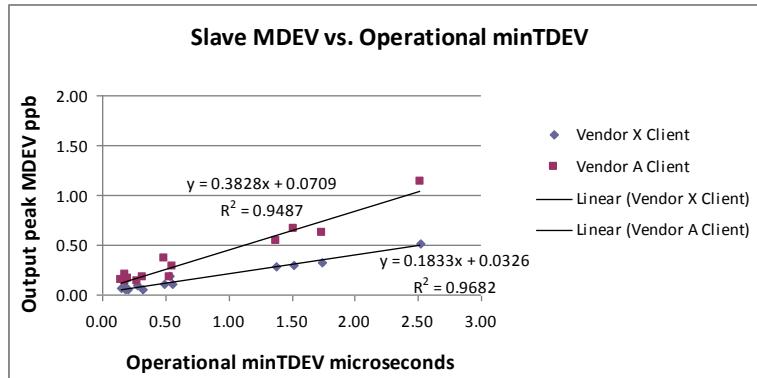


Figure 15. Client output frequency stability (peak MDEV) vs. operational minTDEV (98% and 97% correlation respectively for two vendors).

PACKET NETWORK LOAD PROBE

In contrast to an instrument designed to measure packet delay variation – the PDV probe – where a sequence of probe packets are timed precisely at two points in a network, a load probe investigates an entire stream of packets at a particular node in a network. The individual packets are not individually time-stamped; rather, certain aspects of the packet flow are tracked over time.

The load probe measurement setup is simpler in several ways as compared to the PDV measurement setup. First, the measurement data are all taken at a single node for the load probe. The PDV measurement requires data taken at the master be combined with data taken at the probe. Second, the reference clock requirement is relaxed for the load probe. The PDV probe measurement setup requires two highly stable, coordinated clocks, while the load probe can operate well with an internal oscillator with no requirement for primary reference traceability.

As is usually the case with measurement instruments, the packet load probe measurement approach is based on sampling. The fundamental quantities tracked are idle time, busy time, and number of packets. Idle time is an interval of time during which no packets are detected. Busy time is the opposite, a time during which one or more packets are detected. The final quantity tracked during a sample interval is total number of packets detected. The relationship between busy time and number of packets varies because packet size varies; if all the packets are small, the packet count during a fixed busy interval could easily be ten times the packet count during the equivalent time interval in which all the packets are large.

To understand the nature of the measurement, consider a network node alternating between busy and idle as shown in Figure 16. The initial idle time of 5 milliseconds is followed by 4 milliseconds of packet flow, then 2 milliseconds of idle time, then 17 milliseconds of busy time, and so on, for a total of 200 milliseconds. The specific idle and busy times in the alternating sequence are shown in the first table below the plot. This table also indicates number of packets for each of the busy intervals.

To keep the statistics tracked over time manageable, these parameters are processed into samples. For a given sample interval, cumulative, minimum, and maximum idle and busy time are tracked, as is total

packet count. Certain of these statistics are not entirely independent. Cumulative idle time and cumulative busy time should add to 100%.

In the example shown in Figure 16, a sample rate of 10 Hz applies; the sample interval is, thus, 100 milliseconds. The samples are constructed by searching for minimum/maximum idle and busy times through the first and then the second 100 milliseconds. The cumulative idle and busy times are derived by adding up the totals within the sample interval and then normalizing by the sample interval. Likewise, the number of packets in a sample interval is simply the sum of all the busy time packet counts.

The second table at the bottom of Figure 16 shows the cumulative number of packets, cumulative idle time, cumulative busy time, minimum idle time, maximum idle time, minimum busy time, and maximum busy time for each of the two 100 millisecond samples. The cumulative idle time and cumulative busy time could be expressed in units of time, but have been normalized by the sample interval and, hence, are expressed as percentages.

The load probe operates as a passive probe and needs access to the traffic, or a copy of the traffic. This can be accomplished using a network tap or port mirroring. Unlike an IEEE 1588 slave or IEEE 1588 PDV probe, which only require the IEEE 1588 packets, the load probe needs to access all of the traffic.

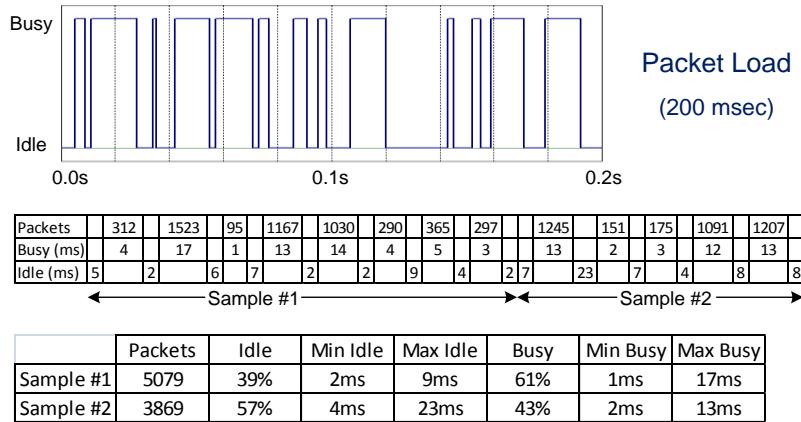


Figure 16. Instantaneous packet flow and the derivation of dynamic packet load parameters.

LOAD PROBE MEASUREMENTS

(1) CONTROLLED PRECISION LOAD GENERATION

To verify a new piece of test equipment, two approaches come to mind. One is to measure equivalent conditions with an existing, independent, comparable piece of test equipment. Another is to carefully control conditions with a complementary piece of equipment. An example of the latter would be testing a newly developed temperature probe by placing it in a precisely controlled temperature chamber. A similar approach was adopted for this experiment.

A traffic generator was used to generate a series of load values in a network averaging to 60% load. The sequence was generated using a flicker noise random number generator with values ranging between 30% and 90%, but for the purposes of this test, the numbers could have been regular steps between two values; the important thing was to vary the load using prescribed values and measure the load to see how well they agree.

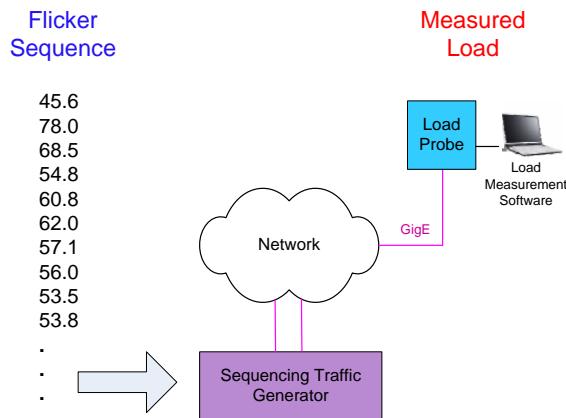


Figure 17. Testing a load probe by setting a sequence of precision levels of load with a traffic generator in a network.

Figure 17 shows the setup for this test. The traffic generator was set to generate load based on the numbers in the flicker sequence, with each load value set for 140 seconds. The load probe was connected to the network using port mirroring to replicate the traffic so that the load probe could have access to all generated packets. The flicker sequence consists of 256 values, so the duration of the test was approximately 10 hours. The traffic was generated as two streams, one with small and medium length packets, and another with large packets, so that the combination would add to the load sequence setting. All interfaces were gigabit Ethernet.

Results from this test are shown below in Figure 18. In this figure, the numbers in the flicker sequence used to generate load levels with the traffic generator are plotted along with measured load. The measured load is represented by cumulative busy time normalized as a percentage as described in Figure 16 and the surrounding text above. If the sample interval is 500 milliseconds, for instance, and the cumulative busy time for that sample is 250 milliseconds, then the cumulative busy value for that sample would be 50%.

The measured load is seen to match the input to the traffic generator well; the upper right graph in Figure 18 illustrates this most transparently. There are some small differences between the blue and red plots there, particularly when the jumps in load are the greatest. This is likely accounted for by the small amount of dead time during load transitions in the traffic generator. There is also a curious anomaly at 4 hours 45 minutes where, based on the traffic generator settings, the measured traffic should change, but does not change; perhaps the traffic generator

sequence was constructed with a repeated value there in place of the correct value or perhaps the traffic generator failed to sequence properly.

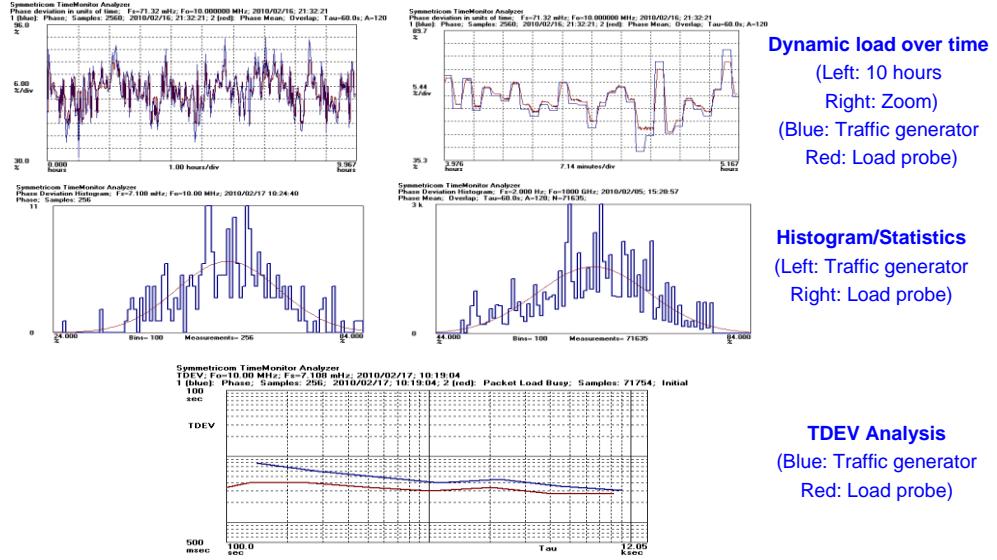


Figure 18. Measured load (cumulative busy) plotted along with input load sequence, histograms with Gaussian fits, and TDEV of input load sequence and measured load.

The same two sets of data, the flicker sequence and the measured load, were used to construct histograms. Both histograms show a similar Gaussian shape; each is shown as a raw histogram along with a Gaussian fit represented by a curve drawn through the data.

A TDEV calculation was also performed on both sets of data, also shown in Figure 18. The TDEV of the input flicker sequence matches reasonably well with the TDEV of the measured load. Both show the zero slope TDEV characteristic of flicker noise.

(2) STEPPED PACKET LOAD AND PDV

In the following experiment, load was increased in a network and the effect on PDV was observed. The setup included both the load probe and 1588 PDV probe (with 1588 grandmaster on the other side of the network) like that shown in Figure 3. A traffic generator was used to generate traffic. The load was changed every 30 minutes, with load levels ranging from 5% to 100%.

Two 3-hour cycles are shown below in left of Figure 19, both as a load probe measurement (upper plot) and as a PDV probe measurement (lower plot). Note that the two cycles are not identical, something that can be seen in both plots. The first two load levels in the first cycle are lower than the first two load levels in the second cycle, though the final four load levels are identical.

The composition of the traffic, that is relative proportions of small (64 byte), medium (576 byte), and large (1518 byte) packets, was kept the same throughout the test, so the load can be represented using packet counts. The packet flow ranged from 15,000 packets per second to 200,000 packets per second (upper left plot in Figure 19).

The 30-minute dwell time for each load level is just as apparent in the PDV measurement as it is in the load measurement. As the load increases, the average latency increases; thus, seeing this in the measured PDV would allow one to surmise that the load is increasing, even without knowledge of the load profile.

Several additional comments can be made with regard to the effect of load on packet delay variation in this network. First, there is a general tendency for the data to spread as load is increased. Second, while it is maintained even at high levels of load, the population of packets at the minimum transit time decreases with increased load.

In the right of Figure 19, the mean and standard deviation of the measured PDV are tracked for the first 3-hour cycle. The PDV measurement in this case includes the actual packet latency, so the “measured PDV” and the “mean PDV” represent packet transit time.

Note that the mean PDV does increase with each increase in load. The standard deviation of the PDV increases for the first several steps in load and then flattens out after reaching a maximum of 25 microseconds. Under certain circumstances, the standard deviation of the PDV data, which represents spread of the data, can decrease as the network moves towards congestion and the number of minimum transit time packets decreases.

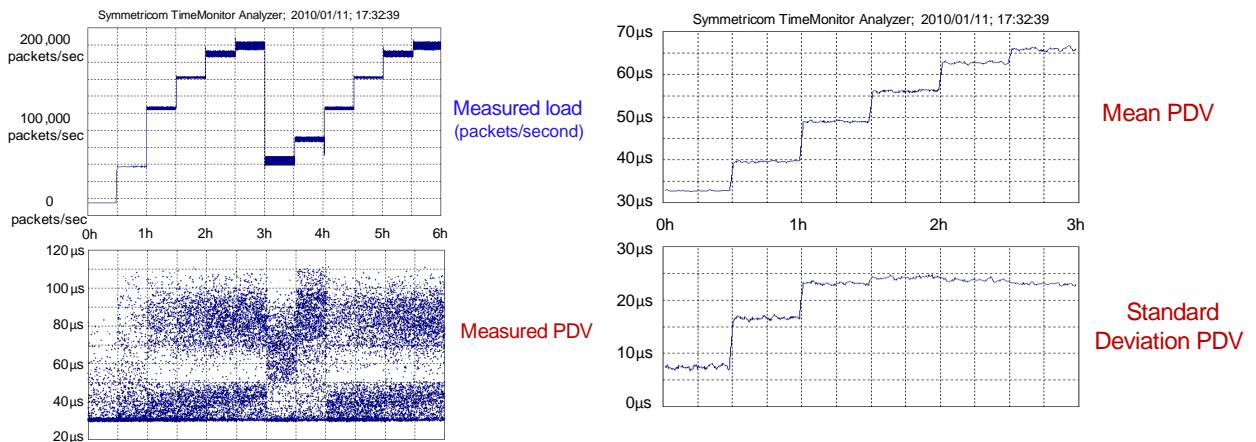


Figure 19. Measured load in packets/second compared to packet delay variation measured at the same node for stepped load, mean PDV (mean packet latency), and the PDV standard deviation tracked over one 3-hour cycle.

(3) DYNAMIC PACKET LOAD AND MIN/MAX IDLE/BUSY TIME

To investigate how minimum busy time, maximum busy time, minimum idle time, and maximum idle time are affected as load changes, a load profile much like that used for the controlled precision load generation experiment described above was applied over a 3-day period. See Figure 16 above and the surrounding text for a description of load probe minimum/maximum busy/idle time and how they are arrived at sample by sample; they are among the statistics tracked by the load probe.

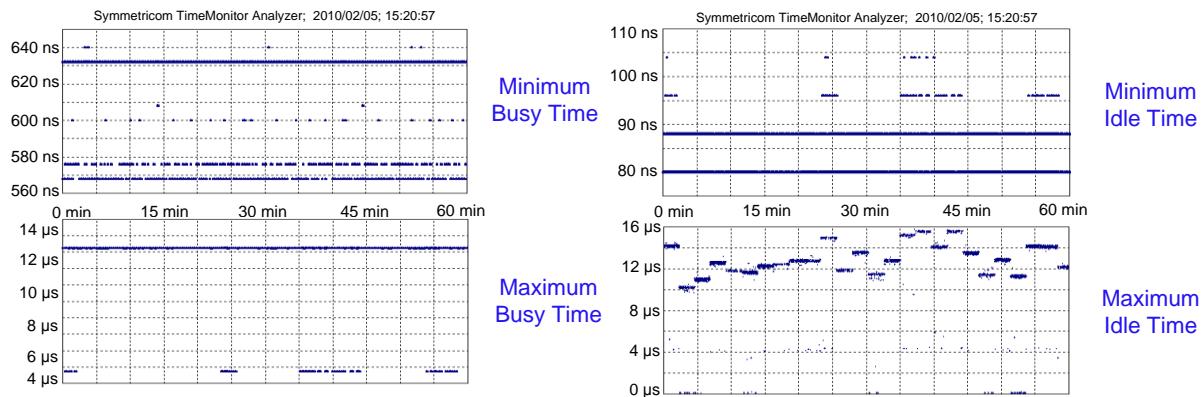


Figure 20. Measured load sample-by-sample minimum busy, maximum busy, minimum idle, and maximum idle time tracked over the first hour of a 65-hour random flicker noise dynamic load profile.

As was the case for the earlier test, a fixed nominal load was varied using a sequence created by a flicker random number generator. Load was measured dynamically using a load probe, like that shown in Figure 3 above. Though the measurement was made over 3 days (the full 3-day set of maximum idle time data was used for the TDEV calculation shown in Figure 21), it proved to be instructive to focus on an hour of the measurement data that was in fact representative of the whole set of data.

The four statistics are shown in Fig. 20. Three out of four of them are modal with minimum busy time taking on six values, maximum busy time taking on two values, and minimum idle time taking on four values. The most interesting of the four is maximum idle time, which shows the actual structure of the load profile with no further processing of the data.

In Figure 21, a TDEV calculation is performed on both the maximum idle time from the load measurement and on the corresponding data from the PDV measurement. There is remarkable correspondence between the two. Based on this result, a load probe could be used to show aspects of PDV behavior. Conversely, this comparison also illustrates that a PDV measurement could be used to show load characteristics directly.

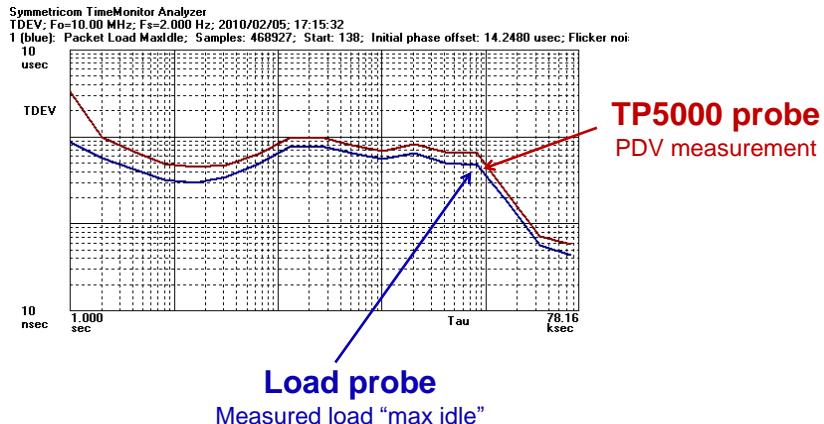


Figure 21. TDEV of maximum idle time data from the load measurement and TDEV of the PDV measurement data.

(4) TRAFFIC GENERATOR CHARACTERISTICS

In this experiment, a traffic generator was studied using the load probe. A 24-hour load ramp was produced with a combination of two packet streams, one with small- and medium-sized packets and another with large packets. These two streams were combined to produce traffic load ranging from 20% to 80% over the first 12 hours and back to 20% over the next 12 hours.

Figure 22 shows busy and idle times tracked over the 24-hour period. As expected, they are inversions of each other; at any given sample busy and idle should add to 100%. The plot on the left shows this for each half-second sample. As the load has been set up in the traffic generator with as a succession of bursts, the load can vary a large amount from sample to sample. Note that initially when the load is nominally at 20%, the “busy” parameter nevertheless ranges from 5% to 50%, as seen at the beginning of the blue “busy” upper plot.

The plot on the right in Figure 22 is derived from the plot on the left by subjecting the samples to a moving average of 120 samples, or expressing this in time, the averaging interval is 60 seconds. Now the linear load ramp between 20% and 80% is much more apparent.

Note, however, that it fails to attain 80% at the peak. The maximum average load is more on the order of 73% (see the blue “Busy” line in the lower plot). This is likely accounted for by two factors. First, it was observed through traffic generation statistics that during periods of instantaneous congestion, packets were being dropped. Second, contention between the dual streams of traffic can also lead to dropped packets.

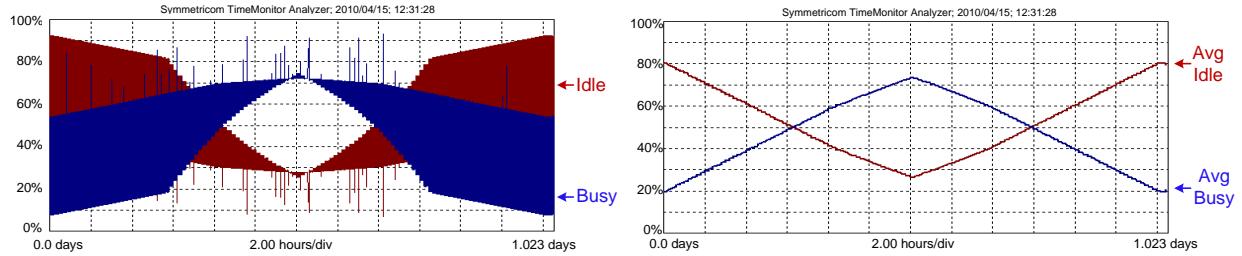


Figure 22. Traffic generator load for 24-hour ramp 20% to 80% from combined small/medium and large packet streams with 0.5-second samples in the upper plot and 60-second averages in the lower plot.

The traffic generator was set up in such a way that the two traffic streams are produced differently. While the large packet stream was produced with bursts, the small/medium packet stream was produced with uniform load.

To show the differences between these two streams, each was measured independently. The measurements of the small/medium packet stream and large packet stream are shown in the left plot and middle plot of Figure 23 respectively. While the change in load every 12 minutes is apparent in both plots, the load is consistent in the former during each step, while it varies considerably for the large packet stream with its bursts. When the large packet stream data is averaged, however, the step increases of load become apparent, as is shown in the plot on the right.

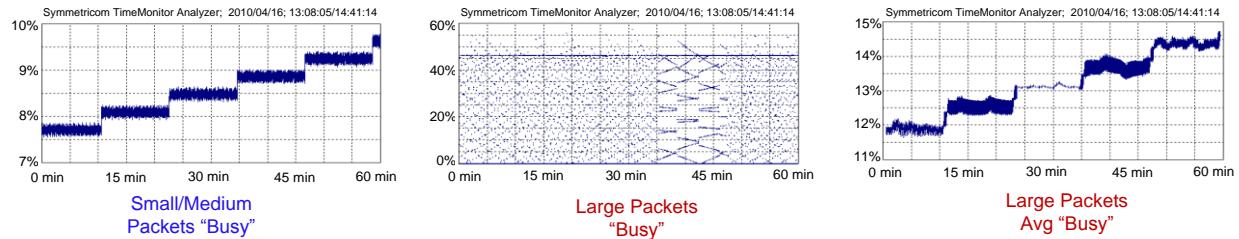


Figure 23. Separate 1-hour measurements on small/medium packet stream and large packet stream.

(5) PACKET STREAM FILTERING

All the measurements made with load probe above derive their statistics from sensing every packet at a particular network node. It can be imagined that there are circumstances under which some subset of the entire packet stream might be of interest for the study of load. Indeed, the load probe discussed here has been designed with the ability to filter traffic according to packet type. For example, the stream of IEEE 1588 PTP packets can be studied separately, or individual VLAN's can be characterized for load. In fact, it is possible to study multiple types of traffic at the same time with a single probe.

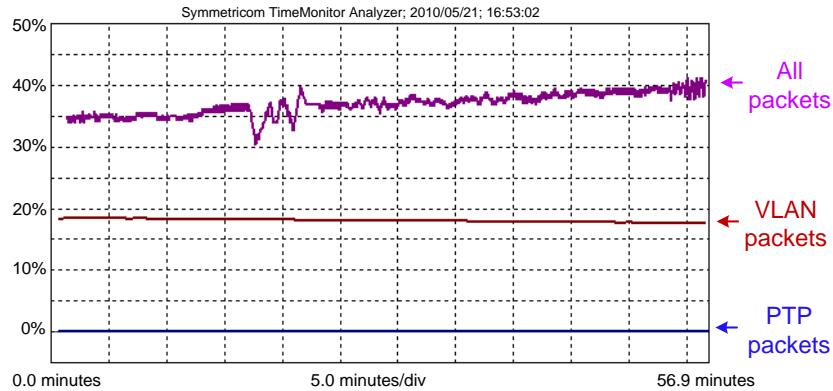


Figure 24. A load probe percentage “busy” measurement characterizing PTP packets, the packets in a single VLAN, and all packets.

The measurement shown in Figure 24 shows a probe measurement simultaneously characterizing PTP packets, the packets in a particular VLAN, and the total packet stream. The PTP bandwidth is very small, showing a fairly constant 0.005% which shows up on the plot as indistinguishable from 0%. The VLAN decreases from 19% to about 17% load. The total packet stream – which includes PTP packets, VLAN packets, as well as other types of packets – increases in load from 35% to 40%, with some interesting modulation 18 minutes into the measurement.

REFERENCES

- [1] ATIS-PP-0900003.2010, “Metrics Characterizing Packet-Based Network Synchronization,” November, 2010.
- [2] L. Cosart, 2009, “Packet Network Timing Measurement and Analysis Using an IEEE 1588 Probe and New Metrics,” in Proceedings of the IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication (SPCS), 12-16 October 2009, Brescia, Italy (IEEE), pp. 116-121.
- [3] L. Cosart, 2009, “Studying Network Timing with Precision Packet Delay Measurements,” in Proceedings of the 40th Annual Precise Time and Time Interval (PTTI) Systems and Applications Meeting, 1-4 December 2008, Reston, Virginia, USA (U.S. Naval Observatory, Washington, D.C.), pp. 165-186.
- [4] L. Cosart, 2007, “Precision Packet Delay Measurements Using IEEE 1588v2” in Proceedings of the 2007 IEEE International Symposium on Precision Clock Synchronization (ISPCS) for Measurement, Control and Communication, 1-3 October 2007, Vienna, Austria (IEEE), pp. 85-91.
- [5] ITU-T Recommendation G.8260, “Definitions and terminology for synchronization in packet network,” August 2010.
- [6] ITU-T Recommendation G.8261/Y.1361, “Timing and synchronization aspects in packet networks,” April 2008.

