**White Paper**

# Ethernet Time Synchronization

Providing Native Timing Within the Network

October 2008

Connecting everything®

**BROADCOM CORPORATION**
5300 California Avenue
Irvine, CA 92617


Phone: 949-926-5000
Fax: 949-926-5203
E-mail: info@broadcom.com
Web: www.broadcom.com

StrataXGSIV-WP100-R October 2008

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# Abstract

The term Ethernet Time Synchronization (ETS) in this white paper applies to both physical layer and packet-based techniques that provide frequency and time-of-day synchronization across an Ethernet network. This document describes the necessity of ETS and its applications within various market segments. It then provides an overview of the techniques and protocols that provide the ETS function.

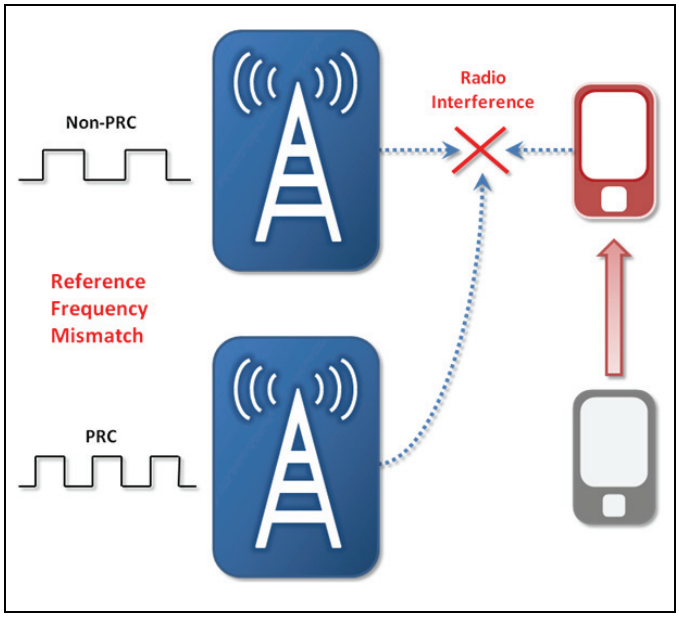
# Background and Applications

Ethernet is fundamentally an asynchronous protocol, optimized for the bursty nature of data traffic. However, Ethernet is quickly displacing existing provider network infrastructure due to enhancements in Quality of Service (QoS), Operations, Administration and Maintenance (OA&M), congestion management, and resiliency.

One of the final pieces missing from a total convergence to Ethernet is the ability to provide timing natively within the network. This would provide Ethernet with the capability to transport time-sensitive applications (such as voice and video) and to distribute precise frequency references. The following sections provide an overview of the necessity of timing synchronization and the motivations behind introducing it into Ethernet.

## Mobile Backhaul

Time synchronization plays a crucial role in mobile backhaul networks. Cellular base stations derive their carrier radio frequencies from a highly accurate reference clock, usually within 50 ppb. This reference clock is typically derived from synchronous Time Division Multiplexing (TDM) interfaces or from expensive GPS receivers located at the base station.




Without timing information traceable to a highly accurate Primary Reference Clock (PRC), local interference between channel frequencies, as well as mutual interference with neighboring base stations will occur, ultimately causing dropped calls and degrading the overall user experience (see Figure 1).

Call handoffs during cellular roaming also suffer perceptible delays when base station clocks are not sufficiently synchronized. An accurate time reference in each base station is required in order to prevent call drops during handoffs.

Mobile billing and maintenance functions also rely on precise timing references. Precise timestamping of data events through each element in the backhaul network facilitates the isolation and traceability of failures and outages.

**Figure 1: Mobile Backhaul Requires Time Synchronization**

Mobile operators now face significant pressure to packetize their backhaul networks due to scalability limitations inherent in their TDM-based infrastructure. Ethernet, with Circuit Emulation Services (CES) provides a clear migration path to cost-effectively provide the flexibility and scalability necessary to serve the ever-increasing demand for bandwidth. As TDM elements gradually approach their end of life, 4G and Long Term Evolution (LTE) Mobile Backhaul networks that replace them will be purely packet-based.

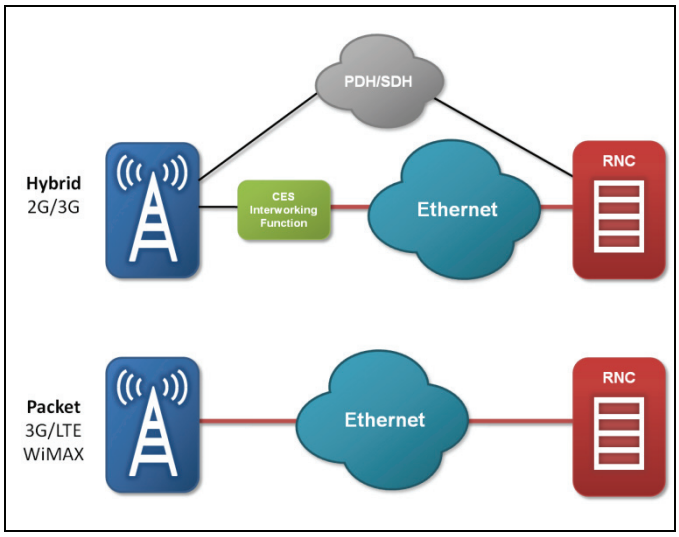



**Figure 2: Mobile Backhaul Evolution**

The migration to a pure packet-based network, however, produces discontinuities in timing distribution as the synchronous frame alignment provided by TDM is lost.

In addition, downward cost pressure on base stations limits the use of expensive GPS receivers to provide timing references.

Either a physical layer or packet-based timing solution must be employed to bridge the discontinuity between synchronous and asynchronous mobile backhaul elements.

Mobile backhaul operators are now evaluating both of these techniques to provide time synchronization within their networks.

## Industrial Automation

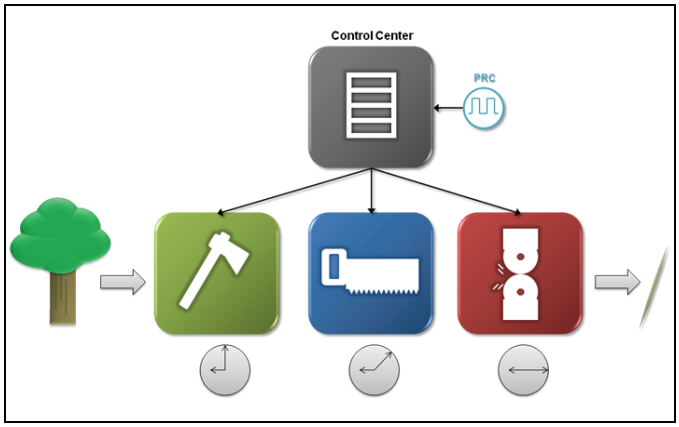



**Figure 3: Time Synchronization in Industrial Automation**

In an industrial automation environment, machinery is typically kept in sync through mechanical means. A single lineshaft drives multiple subsystems through a series of pulleys and gear boxes, ensuring phase lock across the entire system. This approach, however, is inflexible, and mechanical components inevitably fail over time. A malfunction in a mechanical component in any element can cause the entire system to fail.

Rather than link subsystems mechanically, a communications network can be built to interconnect them.

By distributing highly precise timing information to each subsystem, a common point of reference can be used to coordinate their activities.

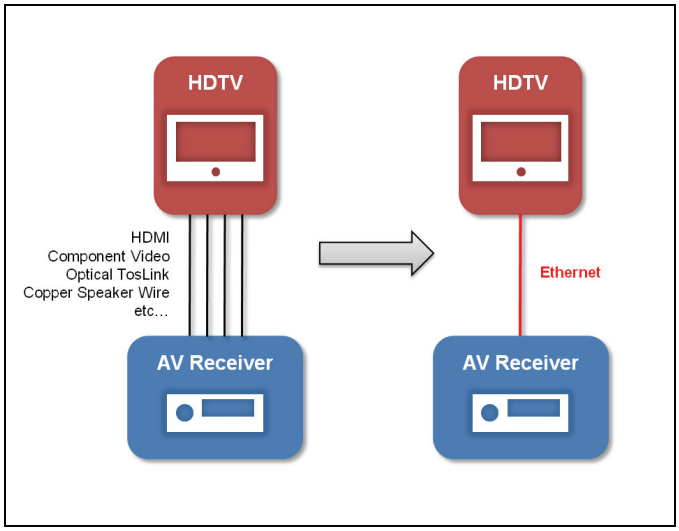
With time synchronization enabled throughout the network, manufacturing activities at each subsystem can be mechanically decoupled from one another. Each manufacturing component has a precise time reference to begin or end their associated actions, thus creating a “virtual lineshaft.”

Industrial automation requires various degrees of synchronization accuracy depending on the exact application (Table 1). Providing this information natively within an Ethernet framework affords an extremely cost-effective means of distributing timing to each subsystem.

**Table 1: Industrial Applications and Time Synchronization**

| Industrial Application | Required Synchronization Accuracy |
|---|---|
| General automation (such as materials handling, chemical processing) | Milliseconds |
| Precise motion control (such as high-speed packaging, printing, robotics) | Several milliseconds |
| High-speed electrical devices (such as synchrophasor measurements) | Microseconds |
| Electronic Ranging (such as fault detection, triangulation) | Submicroseconds |

## Audio/Video




In modern consumer electronics, audio and video data are increasingly gathered, transmitted, and stored in digital form. Yet the most common home networking technologies, those based on the IEEE 802® family, do not provide adequate quality of service for live streaming.

Broadcom has a chairmanship position in the IEEE 802.1® Audio Video Bridging Task Group (AVB TG), an organization that is developing a new standards set that allows very high quality streaming services with only a modest increase in complexity. One component of this effort is the development of a packet-based protocol (802.1AS), which provides timing and synchronization within a bridged Ethernet LAN.

**Figure 4: Ethernet AV in the Home Network**

The ability to distribute timing information at each media playback device provides a precise reference point for streaming multimedia, thus enabling a convergence to Ethernet as a viable consumer electronics interconnect.

# Ethernet Time Synchronization Principles

This section provides an overview of packet and physical layer synchronization.

## Packet Time Synchronization

The IEEE 1588v2 Precision Time Protocol (PTP) defines a packet-based time synchronization method that provides frequency, phase and time-of-day information with sub-microsecond accuracy.

The IEEE 802.1AS Timing and Synchronization protocol introduces the same PTP concepts into native Ethernet.

Both protocols rely on the same fundamental mechanisms, thus for the purposes of this white paper, they will be treated equivalently.

PTP relies on the use of carefully timestamped packets to synchronize one or more slave clocks to a master clock.

Synchronous time information is distributed hierarchically, with a grand master clock at the root of the hierarchy.

The grand master provides the time reference for one or more slave devices. These slave devices can, in turn, act as master devices for further hierarchical layers of slave devices. These relationships are shown in Figure 5.

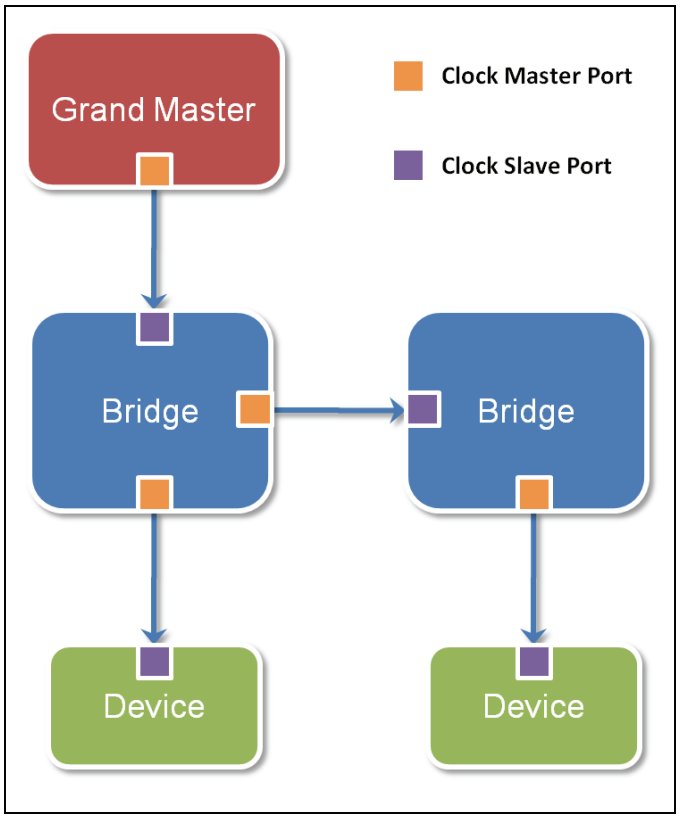



**Figure 5: Time Synchronization Hierarchy**

Synchronizing time across a network requires two essential functions: the measurement of link delays and the distribution of time information. Each node is responsible for independently determining the delays across the network links from it to its link partners. Once this is accomplished, periodic time synchronization messages may be sent from the grand master clock device to the slave clock devices.

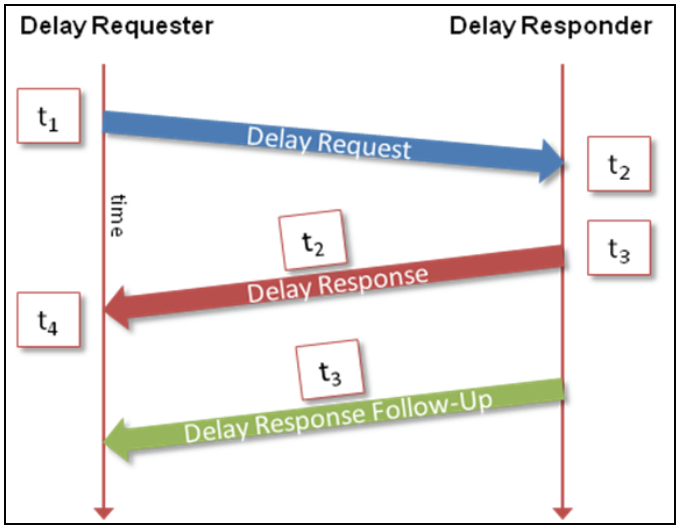
Link-based delays wander over time, so periodic delay measurements are required. Because these delays vary slowly, the period between link delay measurements is typically on the order of seconds.

### *PTP Link Delay Measurement Method*

The PTP delay measurement process is performed as follows:

1. The delay requester transmits a Delay Request to its link partner and captures the timestamp of the transmission time of this packet ($t_1$).
2. The Delay Request message is received by the delay responder, capturing the packet's timestamp ($t_2$).
3. The delay responder issues two packets in response to the preceding request: a Delay Response message and a Delay Response Follow-Up.
   a. The Delay Response conveys the Delay Request receive timestamp ($t_2$). The delay responder captures the transmit timestamp of this Delay Response ($t_3$) as it is transmitted.
   b. The $t_3$ transmit timestamp is then inserted into the Delay Response Follow-Up.
4. The delay requester captures the timestamp upon receipt of the Delay Response message ($t_4$).

The Delay Request/Response flow is shown in Figure 6. The use of follow-up messages eliminates the need to timestamp transmitted packets on the fly, thereby facilitating a hardware implementation.




At the completion of the Delay Request/Response exchange, the delay requester uses four timestamps ($t_1$, $t_2$, $t_3$, $t_4$) to compute the link delay.

The link delay is computed as the average of the two one-way delays using the following formula:

$$T_{delay} = \frac{(t_2 - t_1) + (t_4 - t_3)}{2}$$

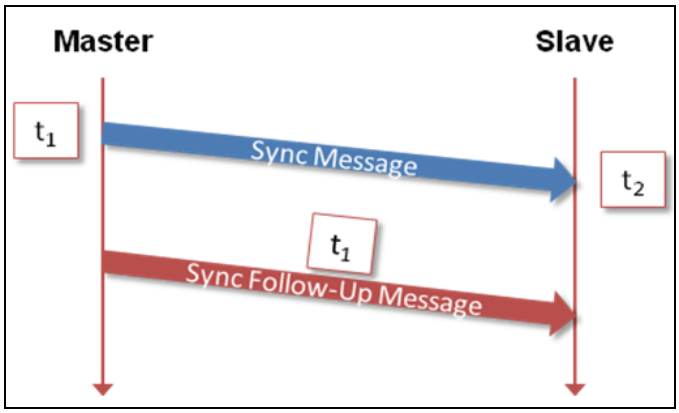
**Note**: The accuracy of this link delay measurement depends on both the symmetry of the one-way link delays and the accuracy of the timestamping process.

**Figure 6: PTP Link Delay Measurement**

The PTP delay measurement process described above is strictly peer-to-peer and is, therefore, limited in scope to the physical ports of the device from one link to its direct neighbor. A more complex end-to-end delay measurement process was proposed during the definition of PTP to calculate propagation delays through multiple network hops that terminate at the slave devices. This end-to-end approach, however, imposed severe scalability limitations on the solution as intermediate PTP devices would need to process thousands of delay conversations flowing through them.

The peer-to-peer method was thus chosen as a more viable solution for PTP.

### *PTP Time Synchronization Method*




A master clock device synchronizes the attached slave clock devices through the use of periodic Sync and Sync Follow-Up Messages (typically every 100 ms). The slave clock devices use the information captured in these messages to perform periodic adjustments to their local clock. In a manner similar to that used for delay calculations, the master clock timestamps the transmission of the Sync Message and the slave timestamps its reception. The transmit timestamp is conveyed to the slave via the Sync Follow-Up Message. This method is shown in Figure 7.

**Figure 7: PTP Time Synchronization - Direct Master/Slave**

The slave device then uses the link delay ($t_d$) and the Sync Message timestamps ($t_1$, $t_2$) to calculate the time offset it needs to compensate its local clock:

$$T_{slave\ Offset} = t_2 - t_1 - t_d$$

A variation in the calculated slave offset over multiple sync measurements indicates a frequency discrepancy between the master and slave clock references. This discrepancy is compensated for at each sync interval to frequency-lock the slave clock to the master.

Figure 7 illustrates the case where the master clock is directly attached to a slave clock. The slave clock derives its timing from the upstream master clock and then acts as a master clock for further downstream devices. PTP allows for a different synchronization model in which the grand master transmits Sync Messages to indirectly attached slave devices.

PTP defines the notion of peer-to-peer transparent clocks which do not participate in time synchronization with the master clock. Rather, they simply accumulate propagation and processing delays of these Sync Messages at each hop, ultimately communicating this to the slave clock for use in its offset calculations. This model is shown in Figure 8.

Residence time is defined as the delay between the reception and transmission of a Sync Message through a transparent clock device. These delays must be fully accounted for in the slave time offset correction.

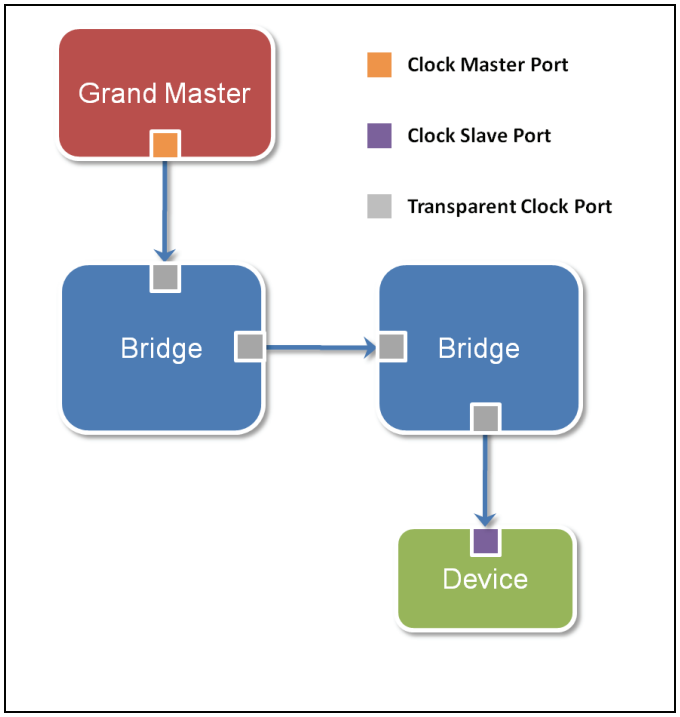



**Figure 8: Time Synchronization with Transparent Clocks**

Figure 9 shows the flow of PTP messages through this example network of transparent clock devices.

The master device transmits Sync Messages on a periodic basis to its slave devices.

The transmit time of the Sync Message is measured by the master device, and this time-stamp value is conveyed in a Sync Follow-Up Message in the usual manner.

Transparent clock device 1 receives the Sync Message and timestamps it.

Upon receipt of the Sync Message's Sync Follow-Up Message, transparent clock device 1 transmits the Sync Message to transparent clock device 2 and timestamps it as it does so.

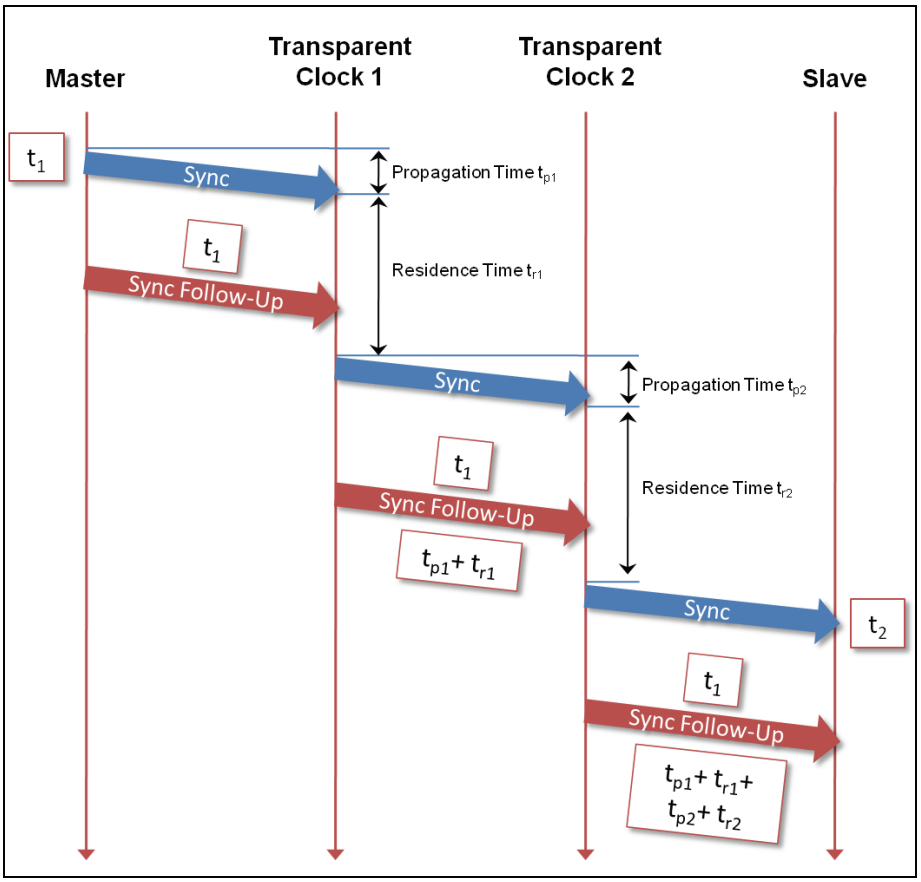



**Figure 9: PTP Time Synchronization - Indirect Master/Slave**

The residence time in transparent clock device 1 is added to the total propagation time. This sum is used as the correction factor included in the Sync Follow-Up Message associated with the Sync Message (the correction factor starts off as zero). As this process continues hop by hop, the Follow-Up Messages maintain a running total of the residence and propagation times; resulting in a grand total delay value from master to slave.

Upon receipt of the final Sync Follow-Up Message, the slave device calculates its offset using the following formula:

$$T_{slave\ Offset} = (t_2 - t_1) - \left( \sum_{i=1}^{N-1} t_{ri} + \sum_{i=1}^{N} t_{pi} \right)$$

**Note**: Although the sum of the propagation and residence delays at each transparent clock ($t_{p1}$, $t_{r1}$, $t_{p2}$, $t_{r2}$) is included in the Sync Follow-Up's offset correction field, the final propagation delay from transparent clock 2 to the slave device must be included in order to fully capture the end-to-end delay.

### *PTP Slave Time Correction*

Each PTP network element maintains a PTP-free running time-of-day counter used as a basis for generating recovered clock signals and computing latencies, offsets and drift rates. The free-running counter runs on the local system clock, asynchronously to the clocks maintained by the other members of the PTP network. Correction factors are applied to the free-running clock in order to arrive at a local time value that is synchronized to the grand master clock. The PTP free-running time value consists of a 32-bit nanoseconds field and a 48-bit seconds field. At each rollover of the nanoseconds [31:0] field (109 nanoseconds), the 48-bit seconds [47:0] counter is incremented.

| seconds[47:0] | nanoseconds[31:0] |
|---|---|

The PTP network element uses the information calculated from the master clock's sync messages to perform local clock adjustments. These adjustments are shown in Figure 10.

**Drift Adjustment**

If the trend of slave offset values calculated from the Sync Messages continues to increase or decrease over time, the local reference clock that increments the free-running counter is operating at a rate slightly slower or faster than the master reference. A drift adjustment can be made to the free-running counter by slightly increasing or decreasing the rate at which the counter increments. Doing so locks the frequency of the counter to the master reference (syntonization). Syntonization is the adjustment of a clock signal to match the frequency, but not necessarily the phase, of another clock signal.

**Offset Adjustment**

Once the drift rate has been measured and compensated for correctly, the slave clock offset should remain fairly constant at each Sync interval. Ideally, once an offset is computed and put in place, it is only rarely changed. The offset is applied to the local time value to synchronize the local time with the master's.

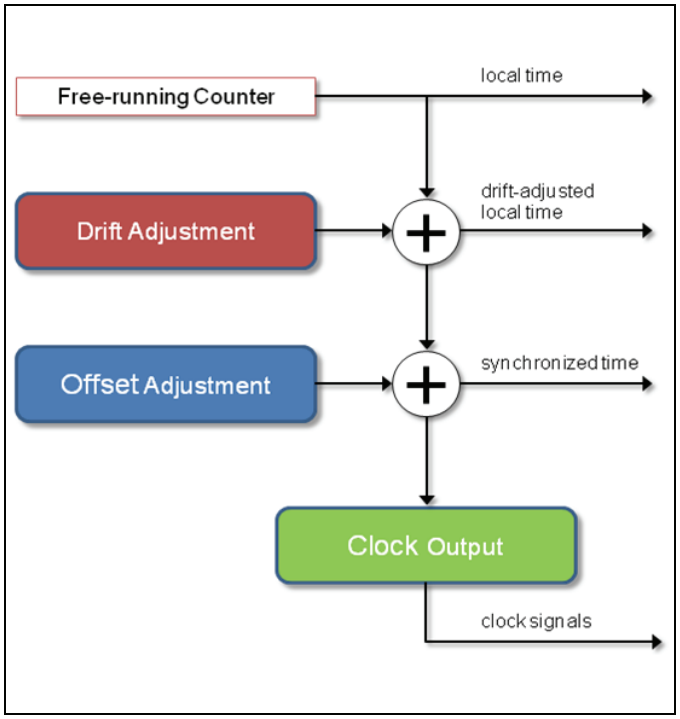



**Figure 10: Slave Clock Adjustments**

**Clock Output**

Ultimately, the slave clock uses the drift and offset adjusted counter to generate a usable clock signal externally. Through PTP, each slave free-running counter is both frequency and time-of-day locked to the master clock. The master clock device communicates phase information to each of its slave clock devices (using methods outside the scope of PTP) to align the generated clock signal edges. The slave clock uses the free-running clock and phase information to generate a frequency and phase aligned clock signal traceable to the master clock.

## Physical Layer Time Synchronization

In modern networks, Ethernet is deployed as a full-duplex switched transport. Full-duplex Ethernet links transmit continuously, with the physical layer transmit clock typically derived from an inexpensive 100 ppm local oscillator.

The receive PHY locks onto this transmit frequency using clock recovery methods to sample the received data. Ethernet packetization eliminates the need for frequency stability as packets are buffered at each hop.

Although Ethernet operates perfectly well without high-accuracy frequency references, one could opt to derive the Ethernet transmit clocks from a Stratum 1 PRC. By recovering this clock using low-jitter circuitry and then using this signal as the local transmit clock, a timing path can be built at the physical layer traceable back to the PRC.

The benefit of this approach is that the Ethernet datapath functions normally, completely unaffected by the timing path built at the physical layer. This timing path remains isolated from Ethernet packetization issues such as drops and packet delay variations. This technique is dubbed Synchronous Ethernet and is specified in ITU-T G.8261.

**Note**: This technique only provides frequency syntonization, and does not provide phase or time-of-day information.

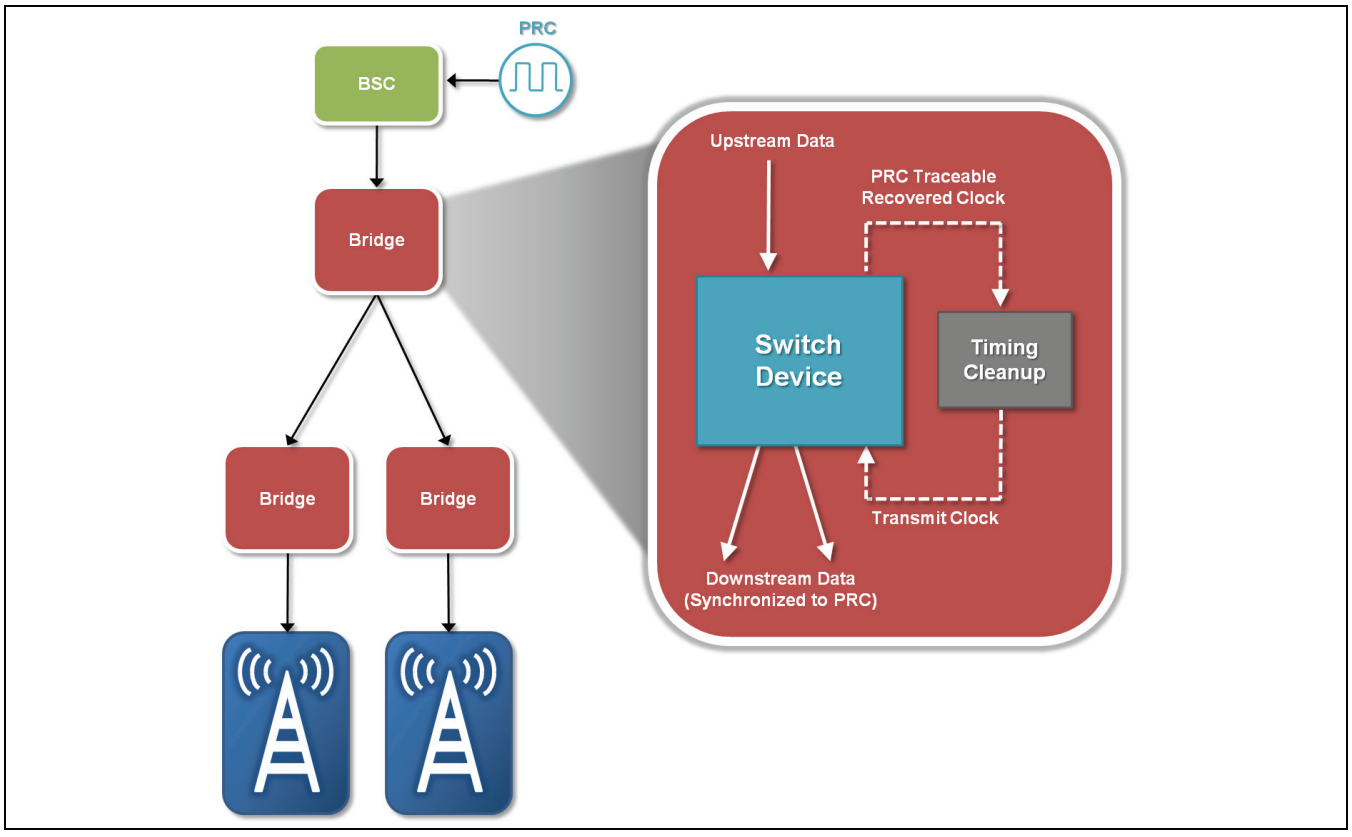



**Figure 11: Synchronous Ethernet Example: Mobile Backhaul**

In the example network shown in Figure 11 on page 9, the Base Station Controller (BSC) has access to a high accuracy PRC which it uses to time its transmit Ethernet interfaces. Each bridge in the network then recovers the PRC-traceable clock at its receive interfaces and subsequently times its transmit interfaces with the recovered clock. The base stations at the ends of the timing path ultimately use the recovered clock timing to derive their carrier radio frequencies.

**Note**: Jitter can accumulate at each hop in the network, degrading the quality of the recovered clock. However, this is no different from the Stratum reference hierarchy and with carefully engineered clock recovery methods, signal degradation can be clearly characterized and controlled.

# The Broadcom® ETS Solution

Broadcom® has introduced Ethernet Time Synchronization (ETS) functions into its line of PHY and network switching products. Integrated ETS provides a switch- and PHY-only time synchronization solution, thereby eliminating the need for an external PTP ASIC and potentially freeing up an additional Ethernet interface.

## Packet Time Synchronization Solution

### *PTP Chip Processing Flows*

Broadcom switch chips implement hardware timestamping at the Media Independent Interface (MII) of the integrated Media Access Control (MAC) modules. Timestamping as close to the physical layer as possible increases the accuracy and quality of the timing information used in the PTP clock adjustments. The Broadcom transmit timestamping process is shown in Figure 12.

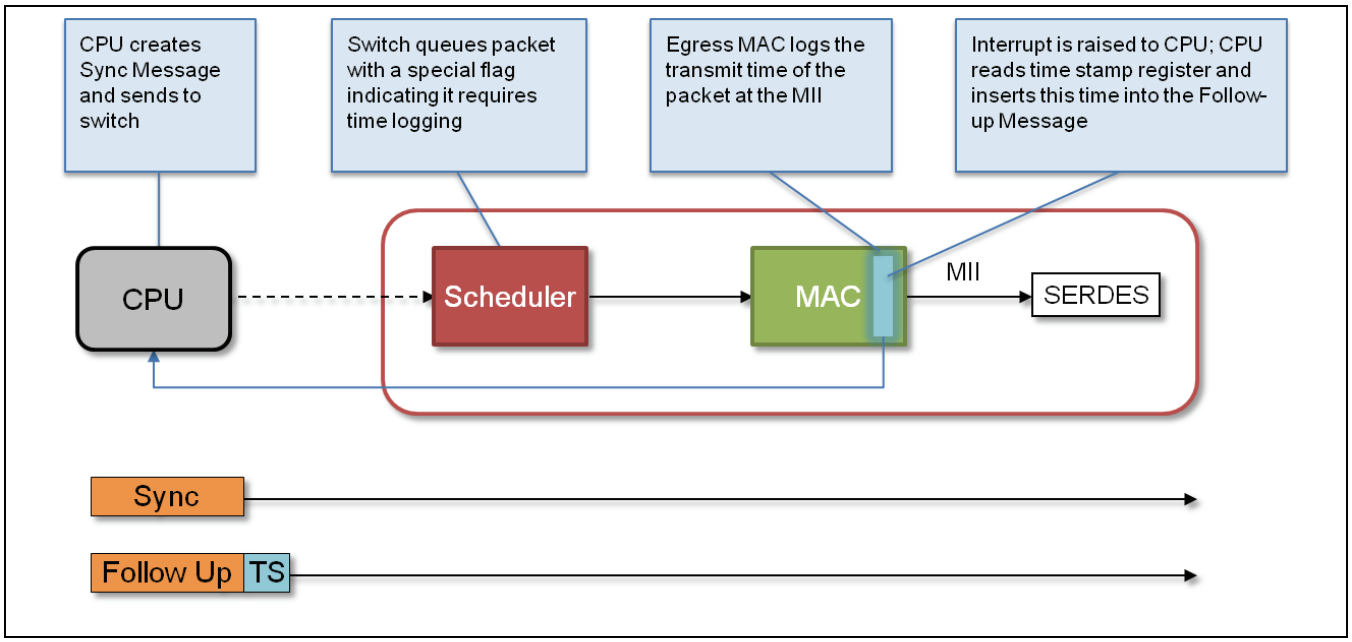



**Figure 12: Broadcom PTP Message Transmit Processing**

The PTP application running on the host CPU initiates the PTP message (Sync or Delay Request). The switch forwards the packet to the appropriate port and the MAC subsequently performs the transmit timestamping. The Time Stamp (TS) is communicated back to the CPU with an interrupt-based model so that the PTP application can send the Follow Up Message.

In a transparent clock mode, the CPU accumulates the propagation and residence delays and includes the total in the error correction field of the Follow Up Message.

**Note**: Each MAC module maintains a 32-bit counter corresponding to just the nanoseconds field of the PTP time format. This counter rolls over at 1s intervals; therefore, the CPU must maintain the 48-bit seconds portion of the PTP time. Since PTP Sync Messages are transmitted and received ten times per

second, the CPU will be able to detect all hardware timestamp rollover events and adjust its 48-bit seconds counter accordingly.

The Broadcom receive timestamping process is shown in Figure 13.

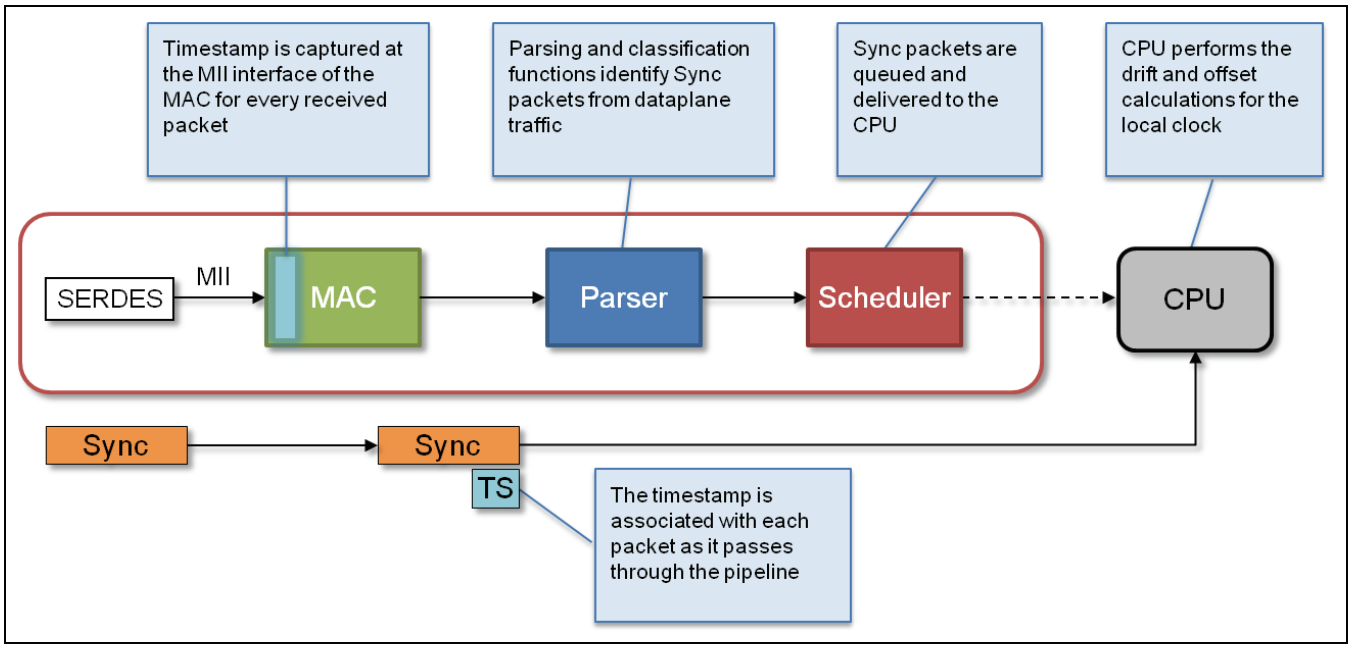



**Figure 13: Broadcom PTP Message Receive Processing**

The Broadcom switch performs the classification and timestamping of PTP packets and passes this information to the host CPU. The PTP application running on the host CPU uses the information in the PTP messages to manage the drift, offset and link delay calculations.

Broadcom provides the necessary logic to perform both hardware and software-based drift and offset adjustments to the local clock. If the system does not require a synchronized clock signal output, the drift and offset adjustments can be performed purely in software. The PTP application must maintain its own free-running clock and perform drift and offset adjustments based on the slave offset values it calculates from each Sync Message to generate the appropriate correction factors in the Sync Follow-Up Messages.

If the system requires a synchronized clock signal output, the hardware offset and drift adjustment logic provided by Broadcom's BroadSync™ interface must be utilized.

### *BroadSync™ Global Time Module*

The BroadSync synchronous time I/O interface provides a means to deliver its clock information to external functions or to be supplied with clock information by external functions. The interface relies on a Free-Running Counter (FRC) and a hardware drift and offset adjustment block called the Global Time Module (GTM). The GTM is used to provide a hardware synchronized time-of-day value and to synthesize a clock signal for use externally, as shown in Figure 14.

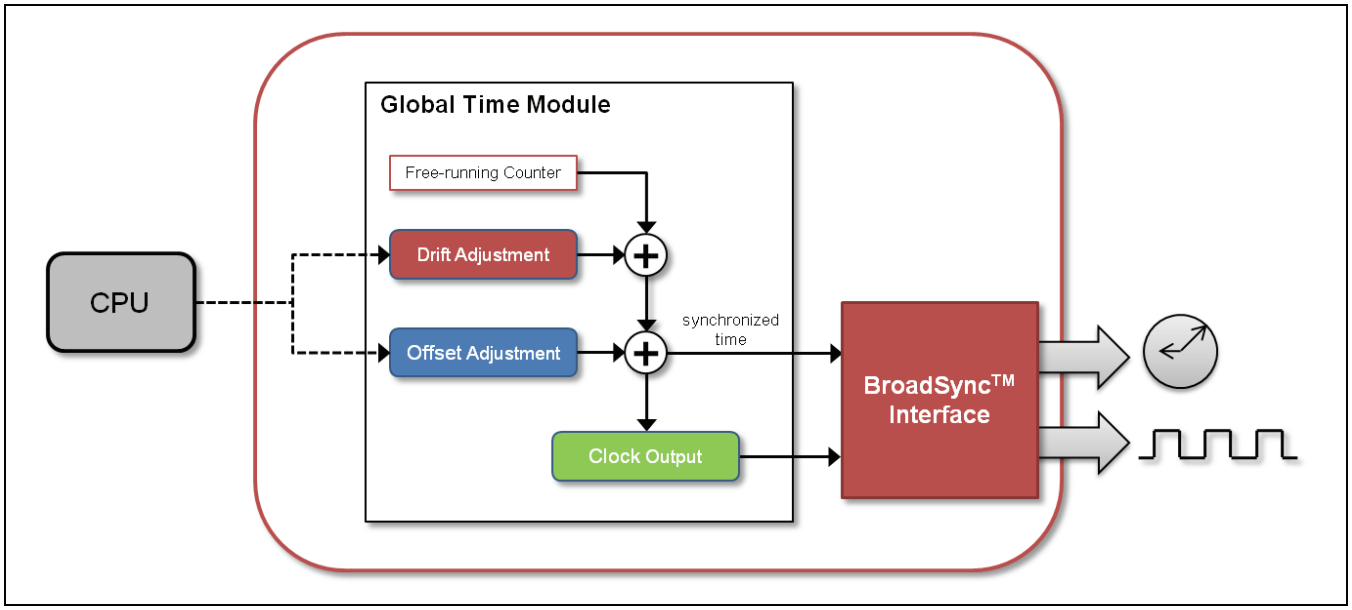



**Figure 14: BroadSync Global Time Module**

The GTM FRC operates on a 125 MHz reference clock, therefore the increment quanta is 8 ns each cycle. The host CPU configures the drift and offset adjustment registers in the GTM based on the trend of slave offset and propagation delay values it calculates from received PTP messages.

#### GTM Drift Adjustment

The drift adjustment portion of the GTM is implemented with a signed 30-bit fractional nanosecond counter, providing highly granular adjustments to the FRC increment rate. At each 125 MHz cycle, a programmable drift rate is added to or subtracted from the fractional nanosecond counter. If the fractional nanosecond counter rolls over or under, the FRC is incremented or decremented respectively. Since the FRC always increments by 8 ns at every 125 MHz cycle, the drift adjustment results in the FRC effectively being incremented by 9 ns or 7 ns at a periodic interval governed by the drift rate.

This process is shown in Figure 15.

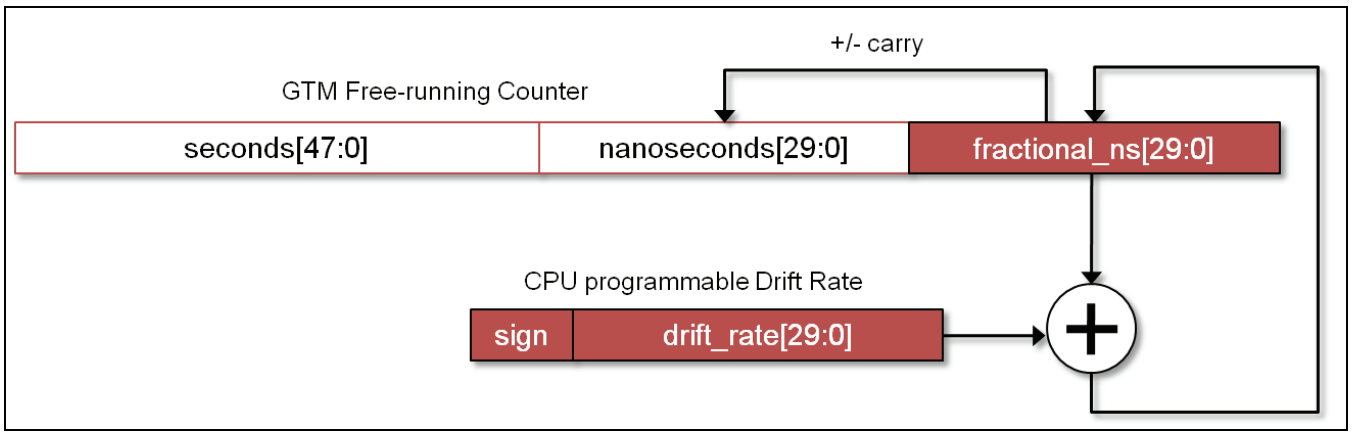



**Figure 15: BroadSync GTM Drift Adjustment**

This drift adjustment allows for very fine-grained control on the rate at which the free-running counter increments, thereby enabling highly precise syntonization to the master clock.

### GTM Offset Adjustment

If the drift adjustment is properly tuned, the offset programmed into the GTM offset adjustment block should be a constant value. The GTM offset adjustment logic simply adds or subtracts a constant value from the drift-adjusted version of the FRC, as shown in Figure 16.

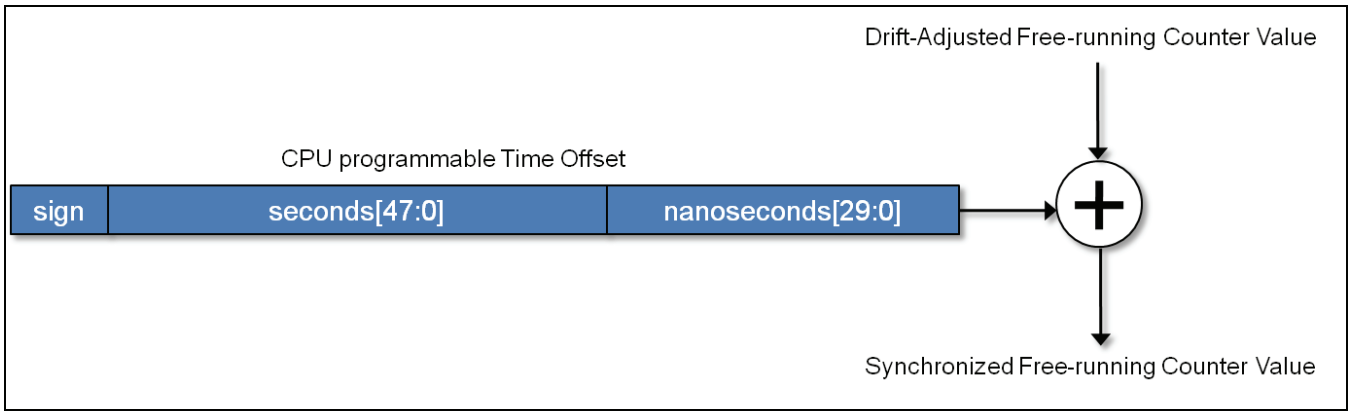



**Figure 16: BroadSync GTM Offset Adjustment**

### GTM Clock Output

The BroadSync GTM clock output block uses the synchronized FRC value to output a digitally synthesized clock signal at a programmable frequency. As typical Synchronous Digital Hierarchy/Plesiochronous Digital Hierarchy (PDH/SDH) reference frequencies are not integral factors of 125 MHz, the clock signal is synthesized using a CPU-programmable half-period register. The half-period register indicates when to toggle the clock signal from a starting point determined by a second register (toggle time). The toggle time register effectively controls the phase of the signal. Each register utilizes fractional nanoseconds to achieve frequency accuracy for non-integral factors of the 125 MHz core clock, as shown in Figure 17.

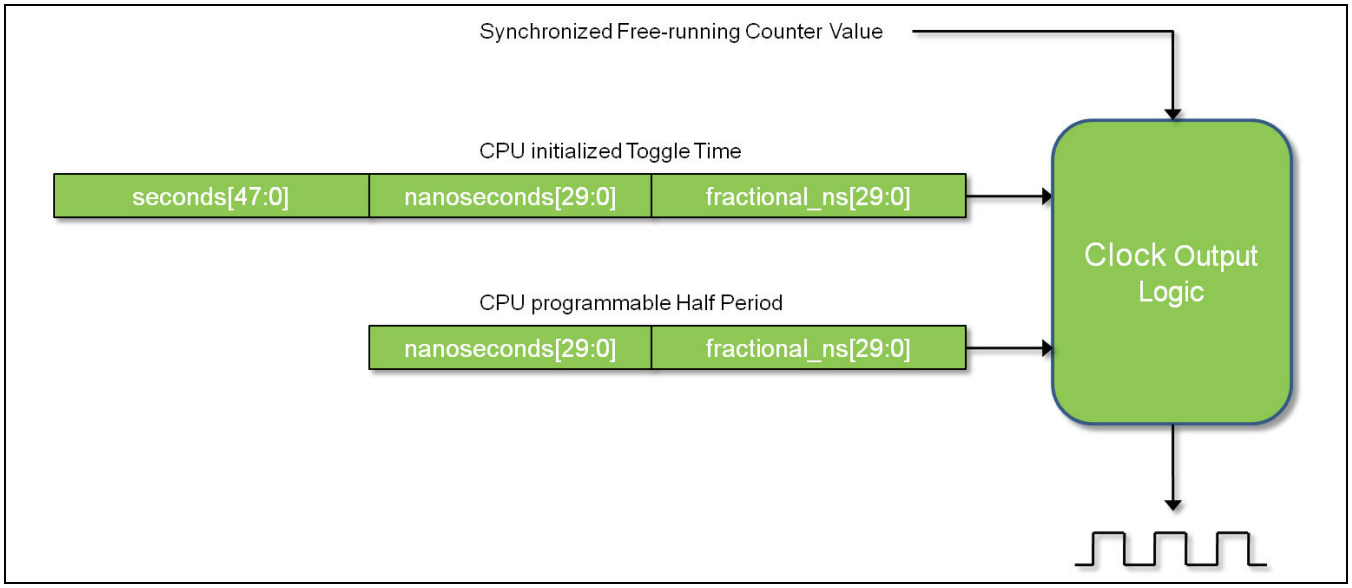



**Figure 17: BroadSync GTM Clock Output**

**Example**: A 1.544MHz reference clock signal is generated by programming the half period register with a value of 323.8342 ns. The CPU reads the current synchronized time value and subsequently sets the toggle time register to a point in time in the future where the first clock transition should occur.

The clock output logic block toggles the output clock signal at the exact time specified by the toggle time register. The half period is then added to the toggle time register in hardware to specify the next point in time when the clock should toggle. This process repeats continuously and does not require any additional CPU intervention after initializing the half-period and toggle time registers.

**Note**: Due to the 8 ns quantization errors introduced by the 125 MHz core clock, the digitally synthesized clock signal will have an inherent +/- 4 ns jitter. If the inherent jitter does not satisfy the application requirements, an external precision Phase-Locked Loop (PLL) must be employed to filter it out.

### *BroadSync Interface*

The BroadSync interface provides a means to externalize the timing information and clock signals generated by the GTM. It can also be used to receive timing from an external source or synchronize timing information within a multichip system when used as an input. The BroadSync interface consists of just three bidirectional signals:

1. heartbeat
2. bitClock
3. timeCode

#### heartbeat

The heartbeat signal runs continuously at a programmable frequency from 1 Hz to 8 KHz and is synchronous with the bitClock. It provides a common reference point in a multichip system to sample local timing information at each device.

To synchronize heartbeat signals, a master chip timestamps the rising edge of its heartbeat input signal and conveys a set of offset and modulo values (such as the heartbeat's period and the time of an upcoming rising edge) to the slave devices. The slave devices use these values to establish the correct phase and frequency of their heartbeat output signals. The frequency and phase information (such as the times at which this signal is expected to toggle) are conveyed from the master to the slaves through the control-plane (which is outside the scope of PTP).

The heartbeat signal can also be used to synchronize free-running counters in each device. This is described in "Broadcom PTP in Modular Systems" on page 23.

#### bitClock

The higher-speed bitClock signal is used to output the digitally synthesized clock signal from the GTM Clock Output block. It is also used to shift the 97-bit timeCode values out of the device.

When acting as an input, the bitClock receives a clock signal and uses it to shift a 97-bit timeCode value into the device from an external timing source.

The bitClock signal runs synchronously with the heartbeat and must have a frequency that is at least 100 times that of the heartbeat signal to shift the 97-bit timeCode value into or out of the device.

### timeCode

The timeCode value is shifted into or out of the device serially once per heartbeat period. The timeCode value consists of a lock indication, the time-of-day value for the current heartbeat period and an indication of the quality or reliability of the time value. This bit stream is preceded by a start bit (the timeCode signal is asserted for one bitClock period) and is protected from corruption by an 8-bit CRC value. The brief delay (two or more bitClock periods) between the rising edge of heartbeat and the transmission of the start bit provides time to prepare the timestamp value for shifting out of the interface. The timeCode is formatted to deliver the following information:

1. start (always a one)
2. lock
3. time[79:0]
4. clkAccuracy[7:0]
5. crc[7:0]

The lock bit is asserted if the clock's accuracy is better than an adjustable threshold indicating to the timing slave that the timeCode is usable for synchronization purposes.

The 80-bit time value represents the timing source device's synchronized (drift and offset adjusted) clock value at the instant of the most recent rising edge of the heartbeat clock. This time value is sampled from the BroadSync GTM after it has applied its drift and offset adjustments to the FRC.

The optional clkAccuracy[7:0] value is an indication of the precision of the recovered clock. Table 2 defines its values.

**Table 2: clkAccuracy Definitions**

| clkAccuracy[7:0] | Definition |
|---|---|
| 0 through 31 | Reserved |
| 32 | The time is accurate to within 25 ns |
| 33 | The time is accurate to within 100 ns |
| 34 | The time is accurate to within 250 ns |
| 35 | The time is accurate to within 1 µs |
| 36 | The time is accurate to within 2.5 µs |
| 37 | The time is accurate to within 10 µs |
| 38 | The time is accurate to within 25 µs |
| 39 | The time is accurate to within 100 µs |
| 40 | The time is accurate to within 250 µs |
| 41 | The time is accurate to within 1 ms |
| 42 | The time is accurate to within 2.5 ms |
| 43 | The time is accurate to within 10 ms |
| 44 | The time is accurate to within 25 ms |
| 45 | The time is accurate to within 100 ms |
| 46 | The time is accurate to within 250 ms |

**Table 2: clkAccuracy Definitions (Continued)**

| clkAccuracy[7:0] | Definition |
|---|---|
| 47 | The time is accurate to within 1s |
| 48 | The time is accurate to within 10s |
| 49 | The time is accurate to > than 10 ns |
| 50 through 253 | Reserved |
| 254 | Unknown |
| 255 | Reserved |

The 8-bit CRC value that follows clkAccuracy[7:0] is a standard CRC8 of the preceding 89 bits of information using the polynomial: $x^8 + x^5 + x^4 + 1$.

The BroadSync Interface timing diagram is shown in Figure 18.

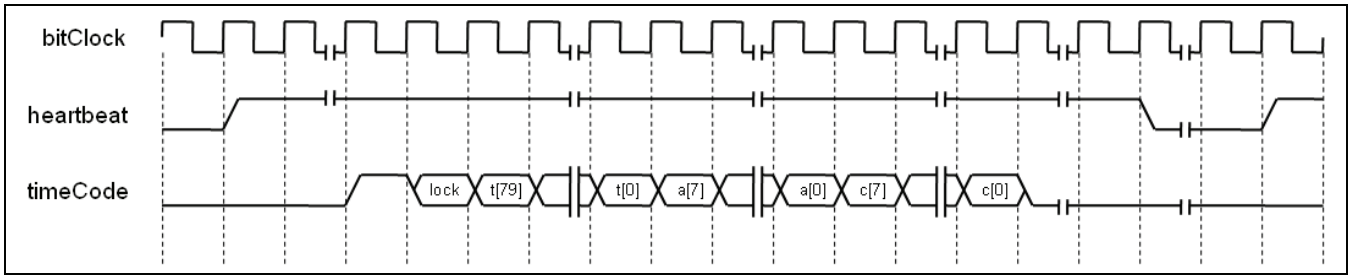



**Figure 18: BroadSync Interface Timing Diagram**

## *Using the BroadSync Interface*

### Master Mode: Timing Input

External hardware provides the bitClock and heartbeat signals as shown in Figure 18. During each heartbeat period, the external hardware also shifts in the timeCode values; consisting of the 80-bit time value and 8-bit accuracy value. The time value shifted in corresponds to the time of the most recent rising edge of the heartbeat signal.

The internal time value is calibrated to the external signals through the following process:

1. The rising edge of heartbeat is used to sample the device's internal free-running clock value.
2. The sampled free-running clock value is compared to the time value that it is subsequently shifted in via the timeCode signal.
3. These pairs of values (shifted-in time and sampled free-running time) are provided to the CPU at each heartbeat rising edge.
4. The differences and rates of change of the differences of the two time bases are used to derive drift and offset compensation values.
5. The computed drift and offset compensation values are used to correct the free-running clock-based timestamp values for use in the follow-up messages.

For very high-precision applications, the bitClock input from the external time source may be multiplied up to a workable frequency by a precision PLL and used to drive the chip reference clock, which ultimately drives the free-running clock. Doing so eliminates the need for drift compensation and requires only offset compensation to derive very precise time information.

### Slave Mode: Timing Output

External hardware accepts the bitClock and heartbeat signals and uses them to synchronize its behavior with that of the grand master. The bitClock signal may be used directly, it may be filtered and stabilized by an external PLL, or it may be multiplied to some more suitable frequency.

The values shifted out via the timeCode signal may be used by the external hardware if time-of-day information is required. To accomplish this, the external hardware must sample its own local time of day value at the rising edge of heartbeat. The difference between that sampled local time and the time value provided via BroadSync is then used as an offset to correct the local time to match the time at the grand master.

## Physical Layer Time Synchronization Solution

Next-generation Broadcom switch and PHY devices provide the capability to export a recovered clock from any of the serializer/deserializer (SerDes) blocks. Both primary and secondary recovered clocks are selectable from any input port. Careful steps were taken in the physical design of these clock paths to minimize jitter and crosstalk, thereby preserving the quality of the reference signal (see Figure 19).

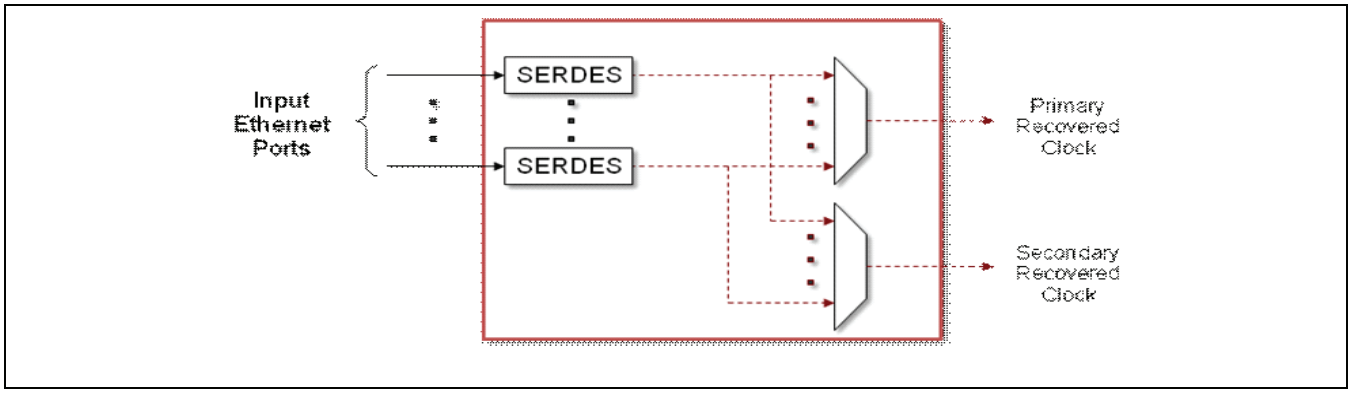



**Figure 19: Exporting Recovered Clocks at ETS Slave**

Clock recovery is performed at the PHY device for copper applications. The switch-recovered clocks are used in 100/1000FX applications where PHY devices are not present. For 10 GbE applications, the clock recovery is performed in the external 10G SerDes devices. In all cases, routing and insertion delay of the multiplexing logic is carefully managed, thus minimizing jitter generation characteristics. In a physical layer ETS network, the timing master maintains a highly accurate frequency reference derived from a PRC. PRC timing is embedded into the physical layer path by driving the local transmit clocks with the high-accuracy frequency reference as shown in Figure 20.

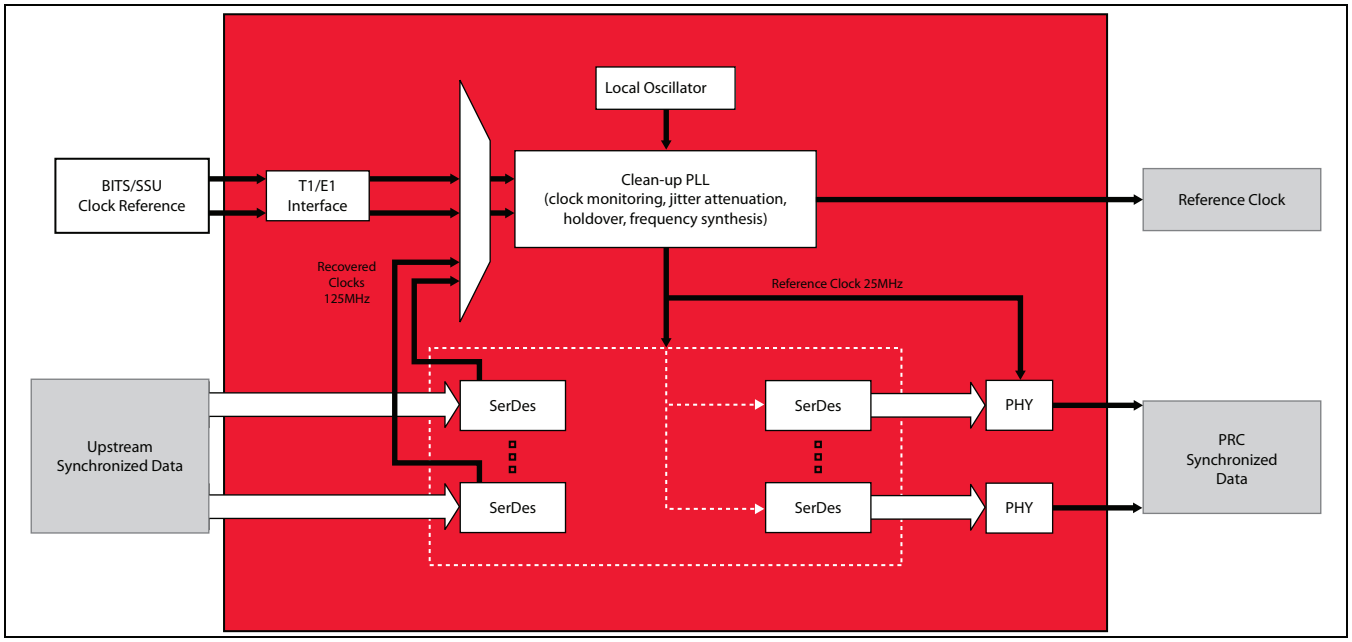



**Figure 20: Physical Layer ETS - Timing Master (SFP Uplinks)**

**Example**: In this example system, the system timing reference is selectable from either the building integrated timing supply/Synchronized Supply Unit (BITS/SSU) clock reference or the synchronous Ethernet uplink ports. The BITS timing reference provides a working and protection clock signal to a clock selection multiplexer. The two recovered line clocks from the Broadcom switch are also fed into the selection multiplexer.

The multiplexer drives a set of clock signals into a PLL that is used to lock the local oscillator to the timing reference. The PLL generates a locked 25 MHz clock signal to drive the RefClk pins of the Broadcom switch and PHY devices. This synchronizes the data on all of the transmit ports of the device to the timing reference and embeds the locked clock signal into the datastream. If an external clock signal is required to provide a timing reference to the rest of the system (such as chassis line cards), the PLL must synthesize the desired frequency.

**Note**: The Broadcom switch does not provide clock monitoring, holdover, or clock division functions on its recovered clocks. An external component must be present to perform these functions, if required.

Timing slaves receive synchronized data from the master system and derive PRC timing by performing clock data recovery on upstream ports. Clock recovery can be performed at the Broadcom switch or PHY device. In the slave system, the recovered clock signals from the Broadcom switch or PHY feed into a clean-up PLL to attenuate accumulated jitter/wander. The PLL also provides clock monitoring and holdover functions to maintain an output reference clock signal in the event that the recovered clock signal is lost or degraded.

Using the reference clock signal to drive its transmit interfaces, the timing slave can then act as a timing master for further downstream devices. If the timing slave is a leaf node in the ETS network, the recovered clock is used as an accurate frequency source for use externally (PDH/SDH clock, radio frequency generation, etc). The timing slave system is shown in Figure 21.

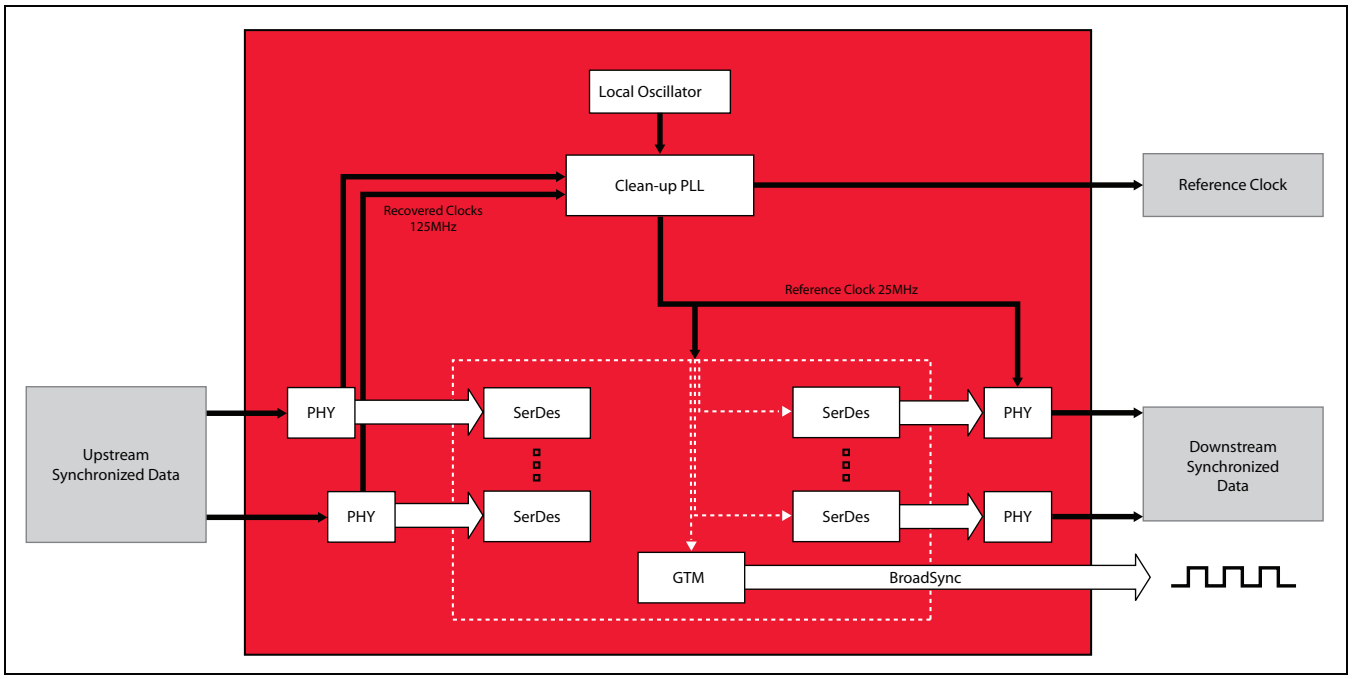



**Figure 21: Physical Layer ETS - Timing Slave (Copper Uplinks)**

The reference clock signal derived from the recovered clock is also used to drive the GTM tied to the BroadSync interface. The BroadSync interface can therefore be used to generate a digitally synthesized clock signal for use externally.

As stated previously, due to the 8 ns quantization errors introduced by the 125 MHz core clock, the digitally synthesized clock signal will have an inherent +/- 4ns jitter. If the inherent jitter does not satisfy the application requirements, an external precision PLL must be employed to filter it out.

The signal quality requirements of the recovered clock depends on the Synchronous Ethernet application. This is defined in ITU-T G.8262, *Timing Characteristics of Synchronous Ethernet Equipment Slave Clock*. One of the key requirements defined by ITU-T G.8262 is the jitter transfer system gain. Jitter transfer is defined as the ratio of input jitter to output jitter in dB.

The physical layer timing path traverses three PLLs; the Clock-Data Recovery (CDR), Clean-up, and Clock Multiplier Unit (CMU) PLLs as shown in Figure 22. The jitter transfer gain of multiple PLLs in series is the arithmetic sum of the individual PLL jitter transfer gains. Since the Clean-up PLL attenuates jitter at a wide range or frequencies, the overall system jitter transfer gain is dominated by the jitter transfer gain of the Clean-up PLL. The external Clean-up PLL should therefore be selected to meet the exact application requirements.

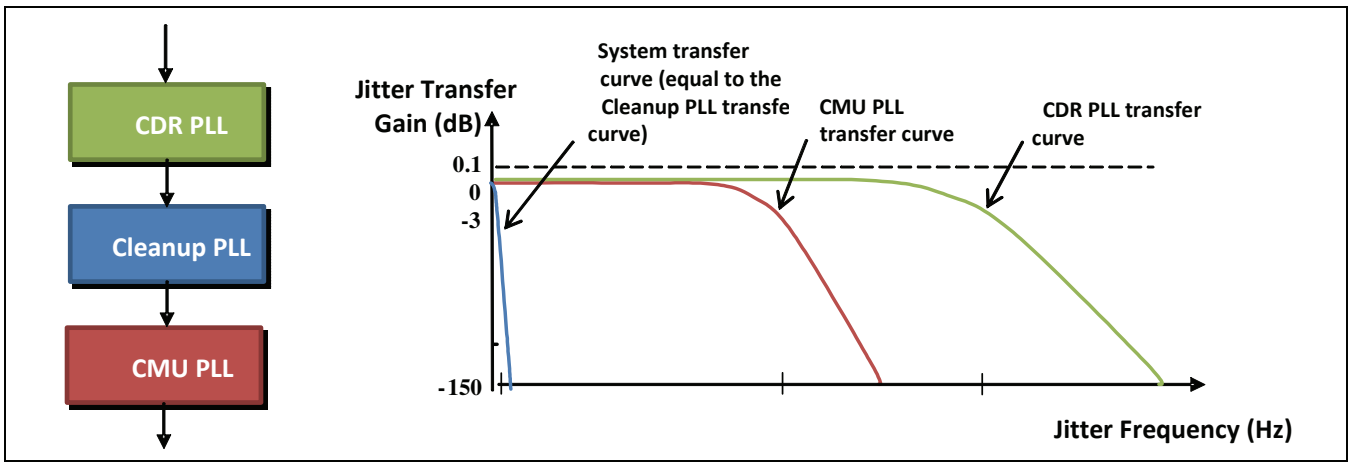



**Figure 22: SyncE Key Parameter: Jitter Transfer**

## Broadcom PTP in Modular Systems

Next-generation Broadcom silicon provides multiple options to distribute PTP timing within a modular system comprised of Broadcom switch silicon.

### BroadSync Method

The BroadSync interface can be used to synchronize local clocks at each line card. The line card that receives PTP Sync messages from the grand master synchronizes its local GTM time to the master clock. This master line card utilizes the BroadSync GTM to generate a low-frequency heartbeat signal to provide a point of reference for the entire system. This heartbeat signal must be distributed across the backplane to each line card that requires synchronization, as shown in Figure 23.

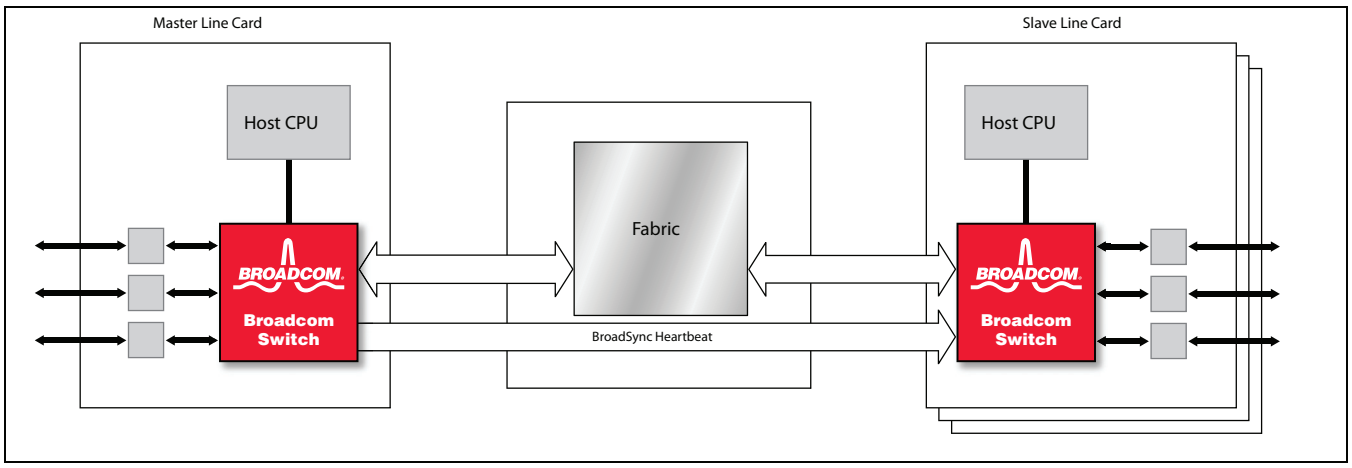



**Figure 23: Broadcom PTP in a Modular System Using BroadSync**

The slave line cards configure their BroadSync interfaces as inputs and sample their local GTM time at each rising edge of the heartbeat signal. The master line card device also samples its synchronized GTM time at each rising edge. A central management entity reads the sampled GTM time values at each line card device and compares them against the master line card's GTM time to calculate the appropriate offset adjustment it needs to make at each line card.

If each slave line card runs off of a local low-accuracy oscillator, appropriate drift adjustments are also made to the GTM time to frequency lock it with the master GTM time, as shown in Figure 24.

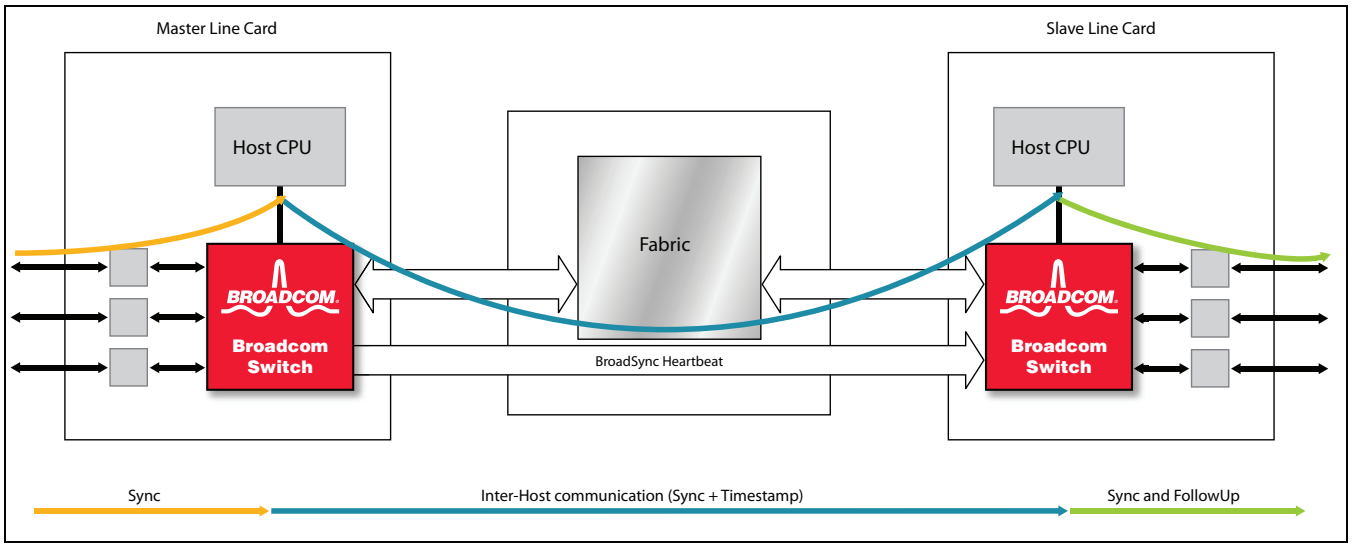



**Figure 24: Broadcom PTP Sync Message in Modular System**

As the timestamps at each line card are all internally consistent due to the use of the BroadSync heartbeat signal, the residence time of the system can be computed accurately by comparing timestamp information from the ingress and egress line cards.

Upon reception of a PTP Sync message, the master line card timestamps the packet and forwards the information to the local host CPU. The host CPU then forwards the packet to the central management entity or directly to the slave line card host CPUs through the control-plane and includes the receive timestamp information. The slave line cards then propagate the Sync Message downstream to their local ports, and subsequently timestamp the outgoing packet. This timestamp is communicated back to the slave host CPU where it can calculate the correction factor based on the receive timestamp provided by the master line card.

### Constant Delay Fabric Method

If PTP control messages are carefully prioritized, packet delay variations through the fabric and the backplane can be minimized. The fabric can then be treated as a transparent element contributing to the link delay between the master and slave line cards, as shown in Figure 25.

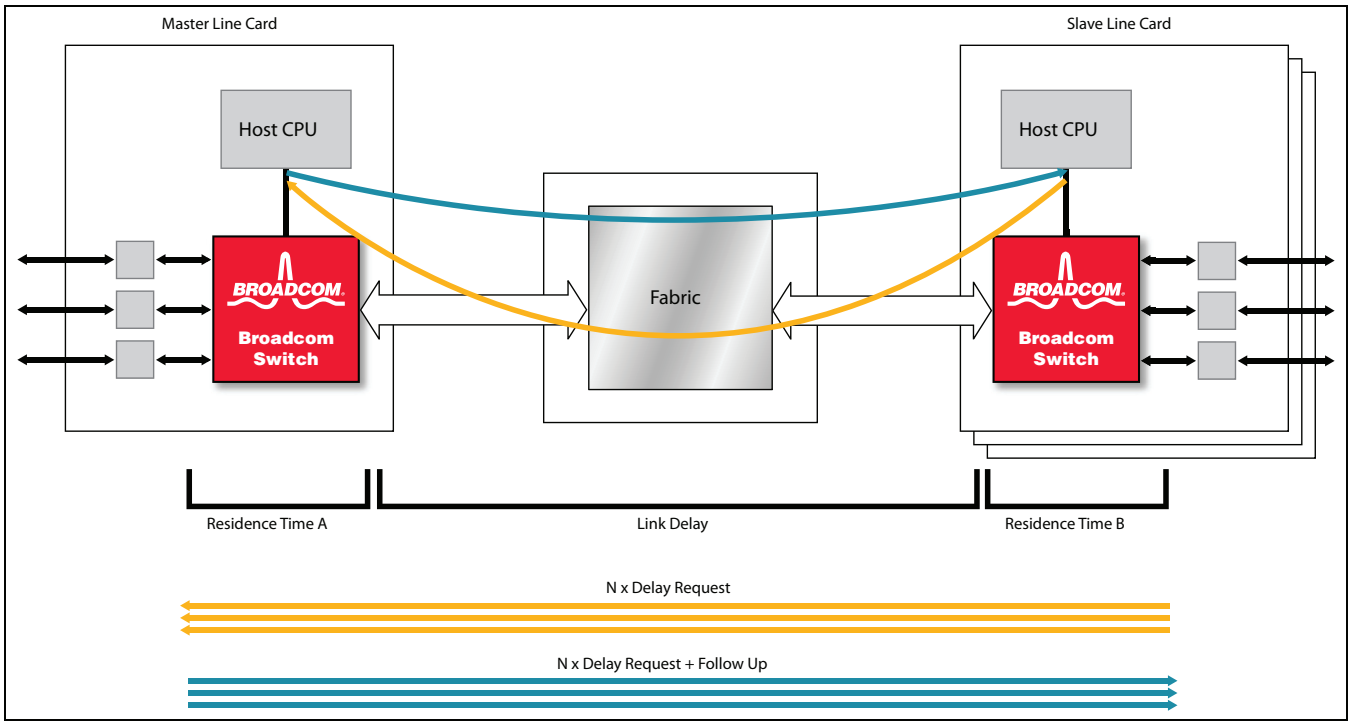



**Figure 25: Broadcom PTP Using Constant Delay Fabric**

In this method, the slave line cards initiate periodic link delay calculations through the fabric element. The fabric element propagates these Delay Request Messages to the master line card. Upon receipt of the PTP messages, the master line card responds with a Delay Response and Follow-Up pair, and includes its local timestamps into the appropriate messages.

**Note**: All intra-system PTP messages should be unicasted to the individual line cards so that the messages are not multicasted by the fabric to other slave line cards.

Using the information from each Delay Response, each slave line card calculates the link delay between the master and slave. This information is stored in the slave host CPU for use in the Sync Message correction field when propagating the messages downstream.

### HiGig™ Method

Next-generation Broadcom silicon supports timestamping functions on HiGig™ interfaces. In a chassis system based on a Broadcom StrataXGS® packet fabric and HiGig backplane, PTP runs within the system on the HiGig links. Each switch element essentially acts as a transparent clock device as shown in Figure 26.

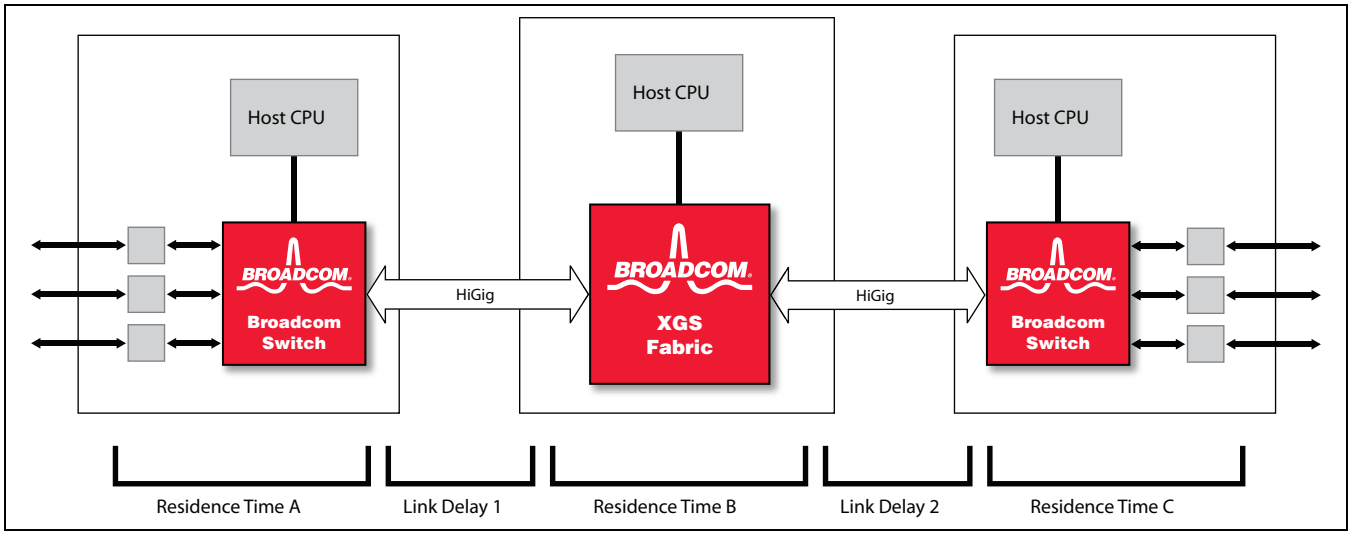



**Figure 26: Broadcom PTP in Modular XGS System**

Each line card and fabric card participates in the PTP protocol independently, propagating residence times and link delays through the HiGig interfaces via PTP messages originated and terminated at each host CPU. Since each element participates in the protocol autonomously, there is no requirement for a central entity to manage the PTP synchronization process.

# Broadcom PTP Scalability and Performance

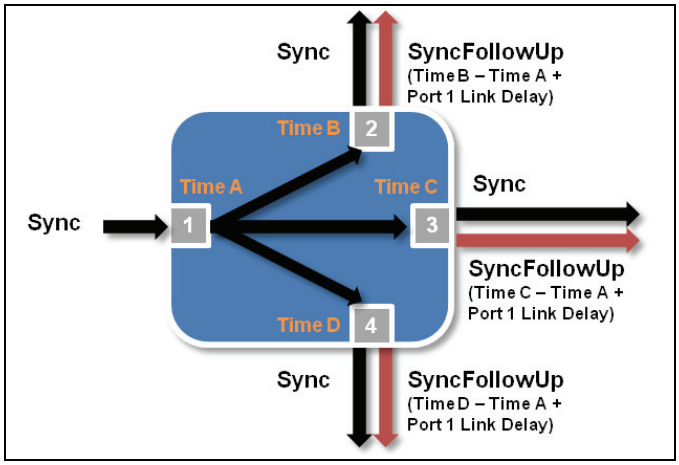
## Broadcom PTP Scalability

One of the goals of PTP (and specifically 802.1AS) is to minimize processing and bandwidth overhead. Although a grand master clock may provide timing to thousands of indirectly attached slaves, PTP intelligently restricts the scope of its messaging to achieve this goal.

The PTP Sync Message is multicast, therefore limiting the number of unique packets that must be time-stamped and processed by the PTP application. For every Sync Message received at each slave or peer-to-peer transparent clock device, only one receive timestamp is required. Since the peer-to-peer transparent clock operation requires a Sync and Sync Follow-Up Message per physical-port rather than per slave clock endpoint, the number of transmit timestamps required is limited to the physical ports of the switch.

A transparent clock device represents the PTP element with the highest PTP processing overhead since it must perform both receive and transmit timestamping functions on multiple bridge ports (see Figure 27). The rate of PTP messages that must be processed at the TC, however, is low because the PTP Sync Message interval is relatively long.

Since the PTP drift and offset calculations are fairly straightforward, the processing overhead will be dominated by the Direct Memory Access (DMA) of the PTP packets between the Broadcom switch device and the host CPU (where the PTP application runs).




For a 24 port switch with a link delay measurement interval of 2 seconds, the PTP overhead amounts to roughly 24 packets per second (pps) received and processed by the PTP application, and 24 pps transmitted by the PTP application (Delay Request and Delay Request Follow-Up Message received and transmitted per port, per interval).

With a PTP sync interval of 100 ms, the sync overhead is 20 pps received (Sync and Sync Follow-Up Message received at master port, per interval) and 460 pps transmitted (Sync and Sync Follow-Up Message transmitted to each downstream port, per interval).

**Figure 27: PTP Transparent Clock Bridge**

This level of packet reception and generation represents a small fraction of the processing capacity of even low-cost microprocessors used in networking applications.

Broadcom switch silicon manages all packet timestamping operations and provides the BroadSync interface and the GTM to perform hardware drift and offset adjustments, thus offering a PTP solution with a protocol overhead that is low from both a bandwidth and CPU loading perspective. Low-cost host CPUs should be able to run a full PTP application without difficulty.

### Broadcom PTP Performance

The IEEE has performed simulations using the proposed 802.1AS standard to demonstrate that it is possible to meet the high-accuracy requirements set forth by the group. Broadcom has taken this characterization effort a step further by building a PTP test network with Broadcom silicon. The PTP test network setup is illustrated in Figure 28 and consists of 802.1AS-capable bridges all based on a reference design using the BCM5395 5-port switch and a cost-effective BCM4704 MIPS® processor.

Each bridge acts as a transparent clock device for distributing timing from the grand master to a single slave clock. The grand master and slave clock each share a common timing source so that accurate correlations can be made between the time communicated by the PTP, and the time at the grand master.

PTP application software was installed on each host CPU to perform the appropriate offset adjustments to the local clock using the timing information gathered from the PTP Sync Messages.

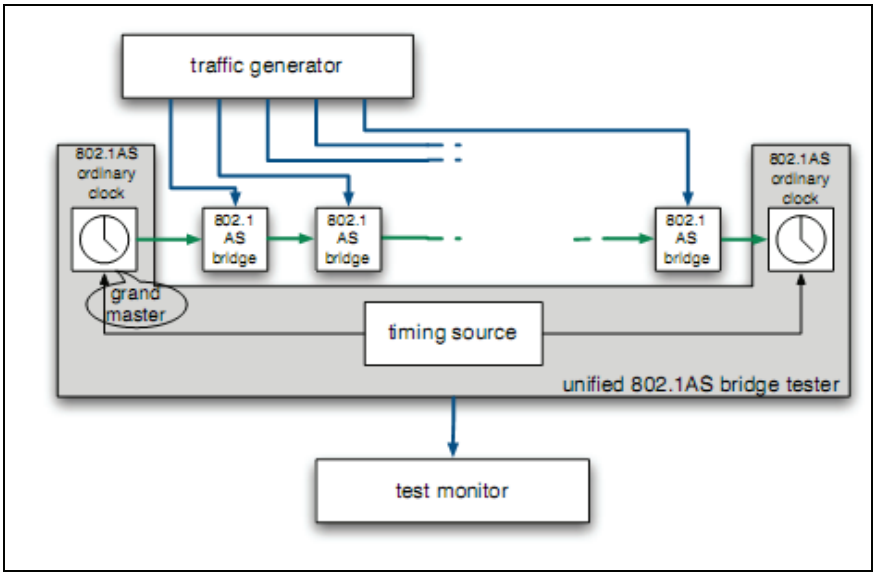



**Figure 28: Broadcom PTP Network Test Setup**

These cost-effective switch chips perform timestamping functions using a 25 MHz clock, which provides a relatively coarse 40 ns timing granularity. However, even with this level of granularity, sub-microsecond accuracy was achieved with various numbers of PTP hops. The results of the test experiment are shown in Figure 29.

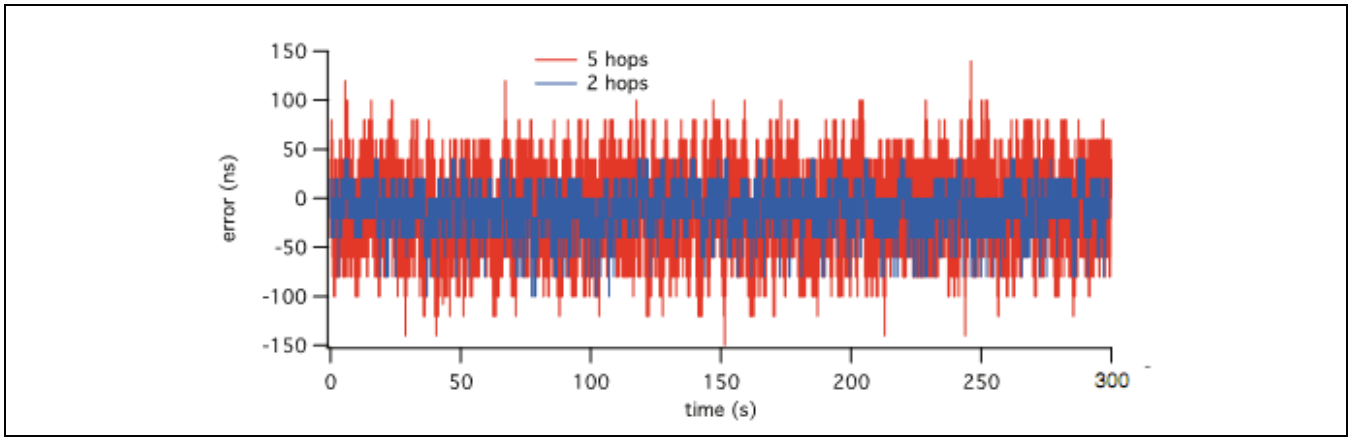



**Figure 29: PTP Hop Test Experiment**

In conjunction, Figure 30 and Table 3 show the Broadcom BCM5395 PTP test and error results.

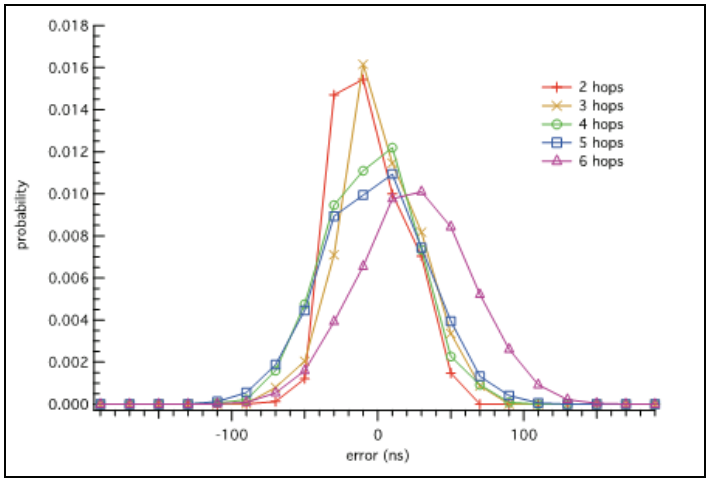



**Figure 30: Broadcom BCM5395 PTP Test - Error Results**

**Table 3: PTP Error Results**

| Number of Hops | Max Negative Error (ns) | Max Positive Error (ns) | Mean Error (ns) |
|---|---|---|---|
| 2 | –100 | 40 | –15.6 |
| 3 | –144 | 100 | –8.9 |
| 4 | –220 | 80 | –14.9 |
| 5 | –188 | 140 | –12.4 |
| 6 | –224 | 160 | –15.0 |

# Conclusion

As the transition from synchronous networks to asynchronous packet networks gains further momentum, a means of providing precise timing distribution becomes critical. Broadcom is leading the effort in both defining these protocols and implementing the products that support them.

Connecting everything®

**BROADCOM CORPORATION**
5300 California Avenue
Irvine, CA 92617


Phone: 949-926-5000
Fax: 949-926-5203
E-mail: info@broadcom.com
Web: www.broadcom.com

StrataXGSIV-WP100-R October 2008