

How Do Synchronous Ethernet And Precision Time Protocol Work? Part 2: Precision Time Protocol (PTP)

by Per Lembre, Director of Product Marketing, Xelerated

Precision time protocol (PTP), defined in IEEE-1588, is the dominating packet method for precision time distribution. Currently, there are two versions of PTP technology available. The PTPv1, released in 2002, defines a hierarchical chain of master-slave relationships. While this served the target application well, it faced scaling challenges for WAN applications and carrier infrastructure. In PTPv1, synchronization quality degrades as new layers are introduced and more members are added to the hierarchy. This was why a major revision, PTPv2 was developed and approved in March 2008.

Developed to achieve higher accuracy, PTPv2 introduces a couple of new architecture models and message formats. In addition, the standard allows for profile definitions for its usage in specific applications. One example is IEEE 802.1AS, which focuses on how to apply PTP in audio -video broadcast home networks. Profiles allow optional features to be added, including fault tolerance and security. The profile may also specify architecture implementation by limiting the options available.

PTP Fundamentals

PTP defines two essential functions: the measurement of link and residence (switching) delays, and the distribution of time information. PTP event messages are periodically sent from the grand master(s) in the network to all slave clock devices (see Fig. 1). Link delays are measured by each slave node to all its link partners. This information is then used by the node to select a master clock using the best master algorithm available.

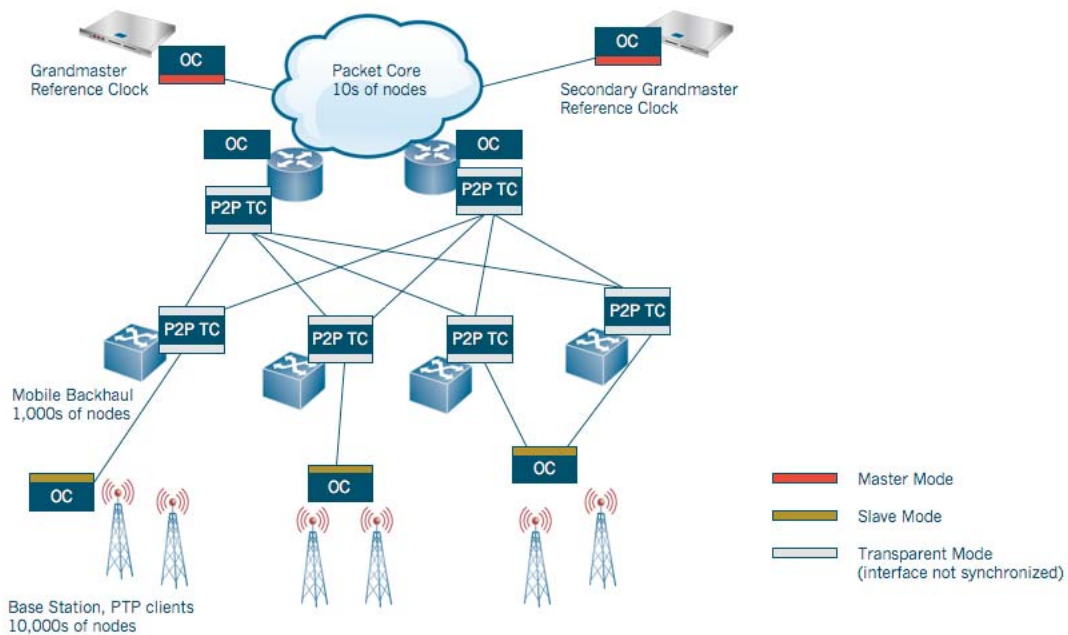


Fig. 1: Example Network Where Switches Operate PTP Transparent Clocks

Because PTP nodes can run more than one clock mode at a time, PTP defines the following clock types:

- **Ordinary clock (OC):** The ordinary clock has a single PTP port in a PTP domain and maintains the time scale used in the domain. It may serve as a source of time (be a master), or may synchronize to another clock, (act as slave). It may provide time to an application or end device.
- **Boundary Clock (BC):** The boundary clock has multiple PTP ports in a PTP domain and maintains the time scale used in the domain. It may serve as a source of time (be a master), or may synchronize to another clock (act as slave). In slave operation, the BC has a single slave port and transfers this time to its master ports. BCs are also used at the interface between PTP domains as parameters may vary between domains. The term Boundary is used because a BC never passes messages from the grand master to connected nodes but instead may act as a new master to them.
- **P2P Transparent Clock (P2P TC, version 2 only):** A transparent clock is not part of the master-slave hierarchy. Instead, it measures the resident time: the time taken for a PTP message to traverse the node. The P2P TC then provides this information to clocks receiving the PTP message. In addition, the P2P TC measures and passes on the link delay of the receiving PTP interface.
- **End-to-End Transparent Clock (E2E TC, version 2 only):** An E2E transparent clock supports the use of the end-to-end delay measurement mechanism between slave clocks and the master clock.

An example of a PTP synchronization architecture using OC and BC clocks is shown in Fig. 2. This configuration is valid for PTPv1 and can also be supported in PTPv2.

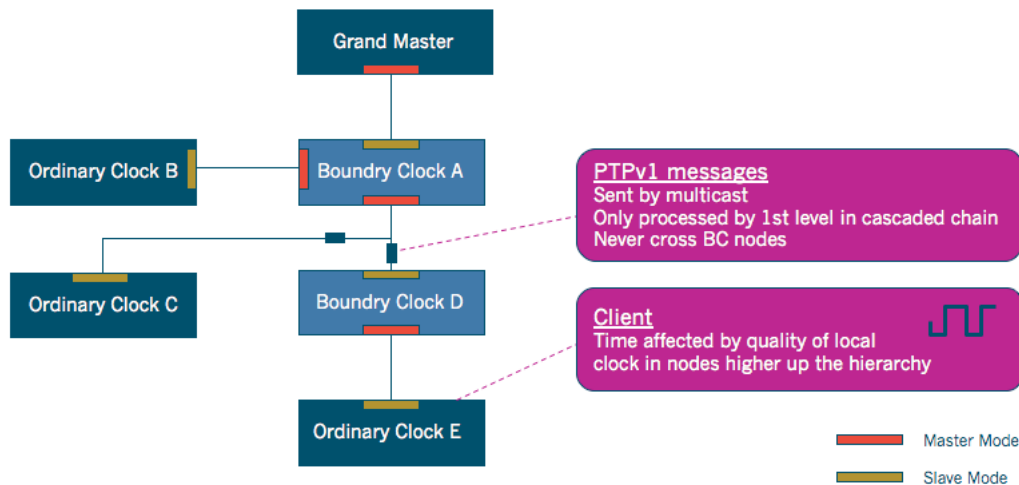


Fig. 2: PTP Synchronization Architecture Using OC and BC Clocks

In the above configuration, sync messages are sent from the grand master to Boundary Clock A. Information from Boundary Clock A's link delay is used to correct its local clock. This clock is then used as master source for Ordinary Clock B, Ordinary Clock C,

and Boundary Clock D. This process is the same for every level in the hierarchy. As every level is dependent of the quality of the clock of the predecessor level, the accuracy degrades as the network scale.

To overcome this problem, the concept of transparent clocks was introduced in PTPv2. Fig. 3 provides a configuration example of transparent clocks in peer-to-peer mode and Fig. 4 illustrates the same configuration using the end-to-end mode. Both are valid, but cannot be used on the same path.

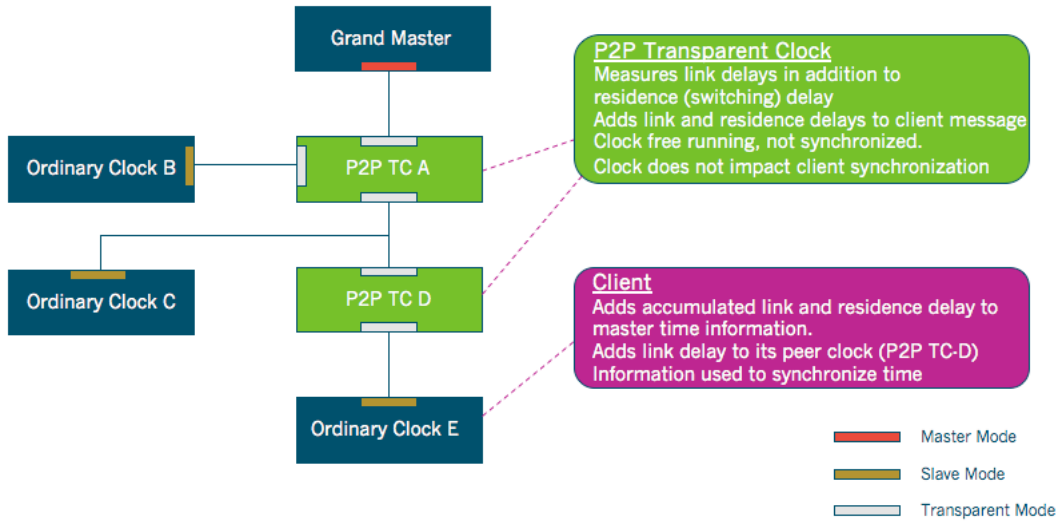


Fig. 6: Peer-to-Peer Transparent Clocks

In transparent mode, PTP accuracy is independent of the local clock running in TC mode. The PTP messages are time stamped on the incoming interface and the TC must measure the time taken before the message is sent on the outgoing link – thus preventing the succession of accuracy declines found with ordinary and boundary clock architectures. This time data can then be added to the same message (one-step operation), or a follow-up message (two-step operation).

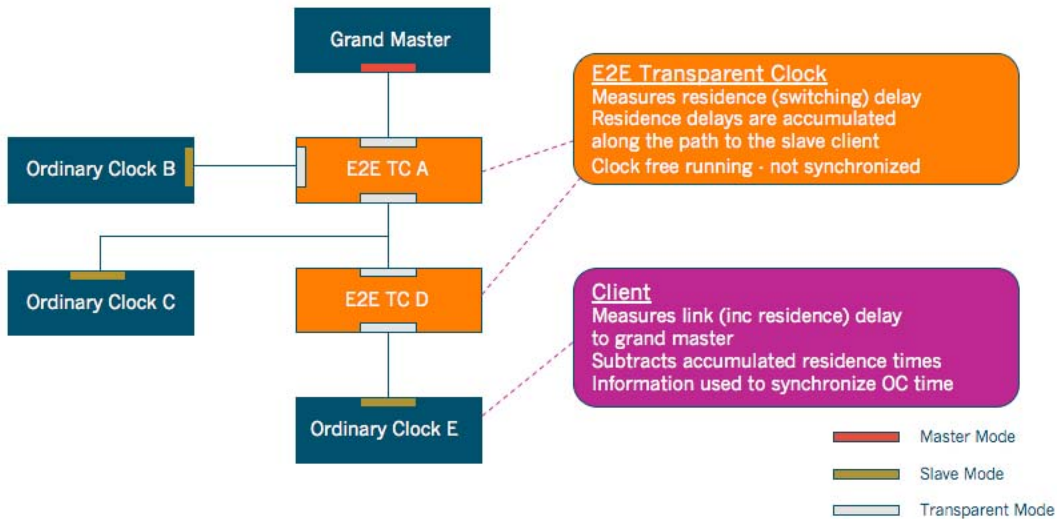


Fig. 4: End-to-End Transparent Clocks

The P2P TC version is gaining most interest for usage in switches and routers in carrier deployments; however, multiple architectures are being evaluated and it is feasible to use different techniques in various parts of the network. For example, locally cascaded cell sites may define one PTP domain while the aggregation and core networks may be managed in a separate PTP domain.

Measuring Link Delay

PTP was designed to accurately measure delay on intermediate links and nodes. Each node is responsible for independently measuring link delays to its link partners. As link delays vary slightly over time, the PTP defines a mechanism to periodically measure link delays. Updates are typically in the order of seconds.

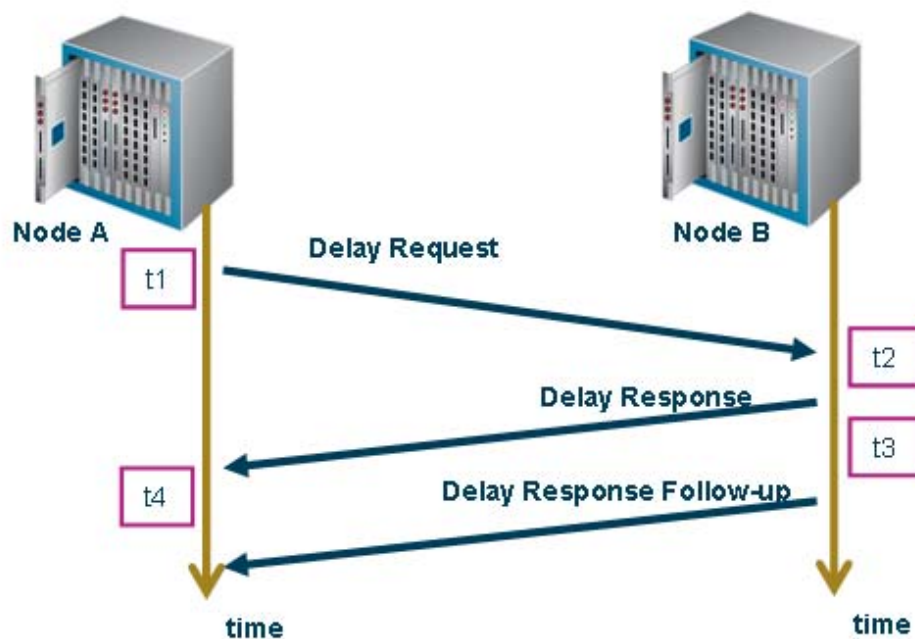


Fig. 5: Link Delays in Peer-to-Peer Transparent Clock Mode

The basic principle to determine link delays, (see Fig. 5) has the following sequence:

1. Node A sends a peer delay request to its direct link partner of the PTP link. It captures the departure time stamp (t1) of the outgoing message.
2. The peer delay request is captured by node B (of any clock type). B captures the arrival time stamp (t2).
3. Node B sends a peer delay response including t2. It captures the departure time stamp (t3) of the response message.
4. Node A receives the peer delay response message and captures the arrival time stamp (t4).
5. Node B sends a follow-up message with t3. This is collected by Node A. Node A can now calculate the link delay from:

$$Peer-link_{delay} = ((t2-t1) + (t4 - t3)) / 2$$

The accuracy of this method is dependent on

1. Precise time stamping by the hardware of the node.
2. Link delay is assumed to be equal for sending (Tx) and receiving (Rx) ports.

Link delays aren't necessarily equal for Tx and Rx; however, this architectural limitation doesn't contribute to errors jeopardizing overall performance goals of PTP. Instead, Tx and Rx link equality needs to be accounted for in traffic engineering PTP domains.

More important is how precise the packet processing hardware performs time stamping on incoming and outgoing interfaces.

In this method of measuring link delays, peer requests can be initiated by any peer of the link. The delay-response messages may be operated in a multicast or unicast mode for PTPv2. Algorithms are used to extrapolate the difference in packet arrival times based on the delay variations to provide an accurate clock service to the client.

The link delay measurement method for E2E operations is based on the same principle, but is a bit more complex. In this case, the slave communicates directly to the master, and the intermediate TC nodes then add their respective resident timing to the delay request and delay response messages. The E2E delay-response model requires more processing from the participating nodes, and is therefore less scalable than the transparent clock model.

Distribution of Time Information

The Grand Master Clock periodically sends out sync messages to all connected clients. This update frequency can be configured and depends on the quality requirements of the application in use. In other words, the more frequent the updates, the more time information is available to the client for adjusting the local clock.

Unlike the Grand Master, transparent clocks only need to add time information as the sync message travels on its path to the client slave node. This is because all transparent clocks maintain link propagation delays for their peers and residence times are already known. Because every PTP TC adds its individual link propagation and residence values to the same field, the slave node receives the accumulated information: making the calculating of the correct time for the slave straightforward:

$$t_{Slave} = t_{Master} + \sum t_{propagation\ time\ of\ all\ links} + \sum t_{residence\ times\ of\ all\ nodes\ along\ the\ path}$$

The PTP standard defines two methods for distributing the messages containing propagation and residence times.

- In one-step operation, clocks add their residence time to the sync message, as illustrated overleaf in Fig. 6.
- In two-step operation, clocks add their residence time to a sync follow-up message. Operation serial numbers ensure the follow-up messages are associated with the correct sync message. This mode is also illustrated overleaf in Fig. 7.

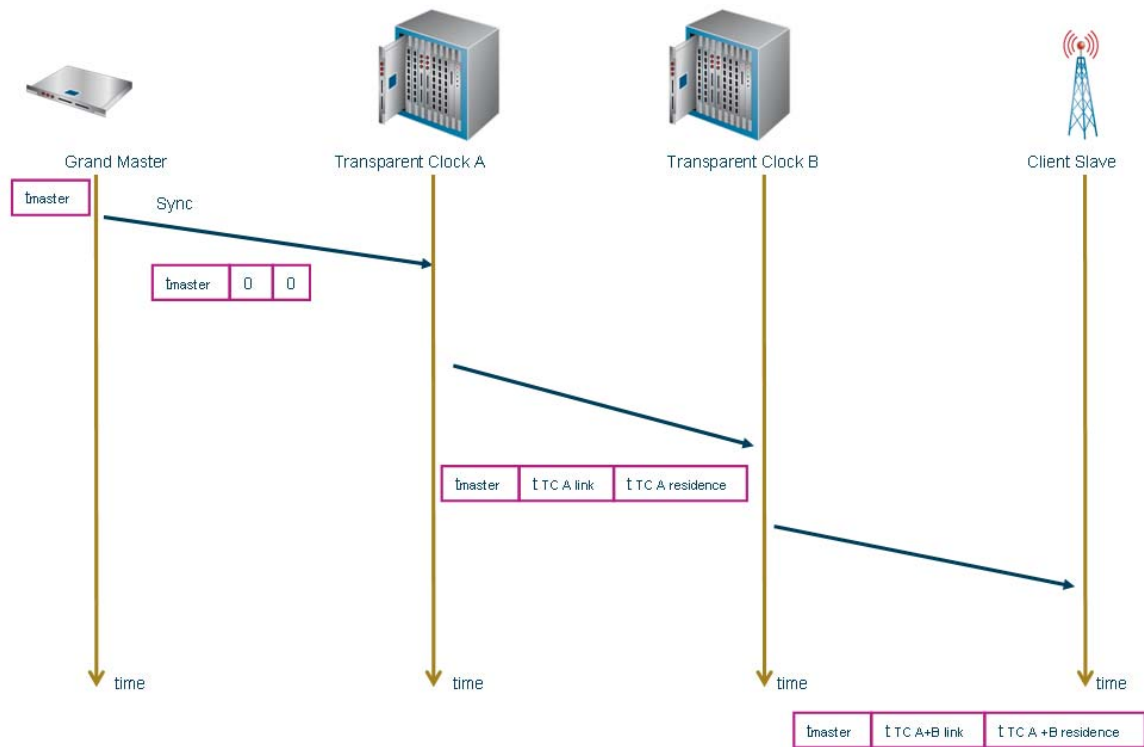


Fig. 6: One-Step Operation for Distributing Time Messages

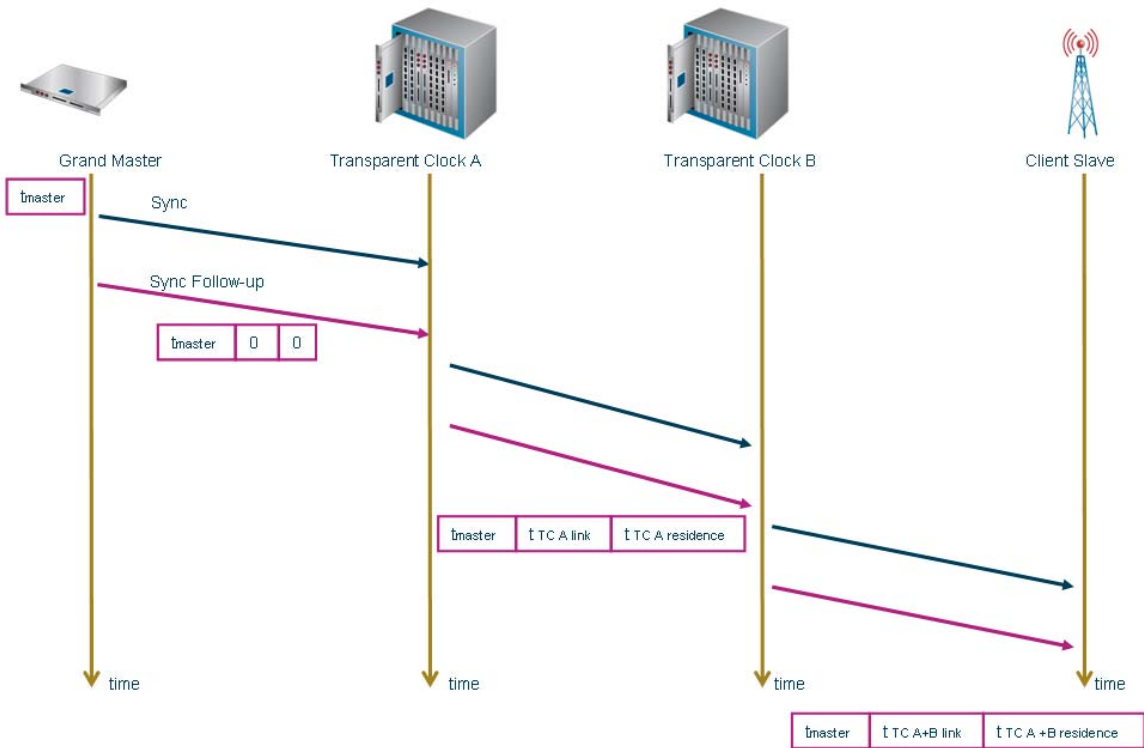


Fig. 7: Two-Step Operation for Distributing Time Messages

Conclusion

When deploying 3G networks and preparing for LTE, mobile backhaul providers will make network design decisions that the industry will have to live with for decades to come, including implications for synchronization functionality. Vendors and service providers should aim for “functionality headroom” in the network design to allow customization for future standard updates.

In reviewing synchronization methods, both Synchronous Ethernet and PTP have their respective merits. In fact, functionality is now available to support all options available and allow for flexible configurations. Vendors should closely review the key feature requirements and benefits provided by synchronization, as outlined in the Table below, for their packet-based mobile backhaul platform design selections.

Requirements	Benefit
PTPv1 and v2	Flexible PTP network architecture
PTP and Synchronous Ethernet supported on all interfaces	Flexible configurations of PTP and Synchronous Ethernet in combination
Synchronous Ethernet on any traffic interface	Both external or backplane interfaces should be configurable to recover clock and synthesize frequency. Ability to lock any SerDes output lane to a local or remote reference clock source.
Strong Monitoring	Monitoring of eight recovered clocks for frequency and phase offset relative to a selectable reference clock.
Flexible output clock	Ability to select “master” of any of the recovered clock and divided by a programmable value for flexible output frequency
Time Stamping Unit located on all interfaces	Time stamping for PTP implementation executed close to physical interface for high accuracy and low jitter. 3.3 ns quantization jitter achievable for GbE.
Deterministic processing	Guaranteed processing of Sync, Delay requests and Follow-up messages.
PTP multiple clock modes	Ordinary Clocks and Transparent Clocks can operate in parallel at no extra cost.
Programmable data plane	Future-proof solution. Enables upgrades to support additional functions such as new encapsulation formats or emerging interoperations between Synchronous Ethernet, NTP and PTP.
Configurable Beat and Sync Clock Rates	Allows for custom system designs.
Hardware support for first and second order PLL	Enables flexible control of the integrated real time clock.

About The Author

Per Lembre is the Director of Product Marketing at Xelerated. He has held a range of product management and marketing positions in the networking industry over the last 10 years. Prior to joining Xelerated, he was head of broadband marketing at Juniper Networks for Europe, Middle East and Africa. He was awarded a Master of Science in Industrial Engineering from the Royal Institute of Technology, Stockholm.

e-mail: per.lembre@xelerated.com

Cell Phone: +46-70-261 21 12



As Published on EN-Genius.net