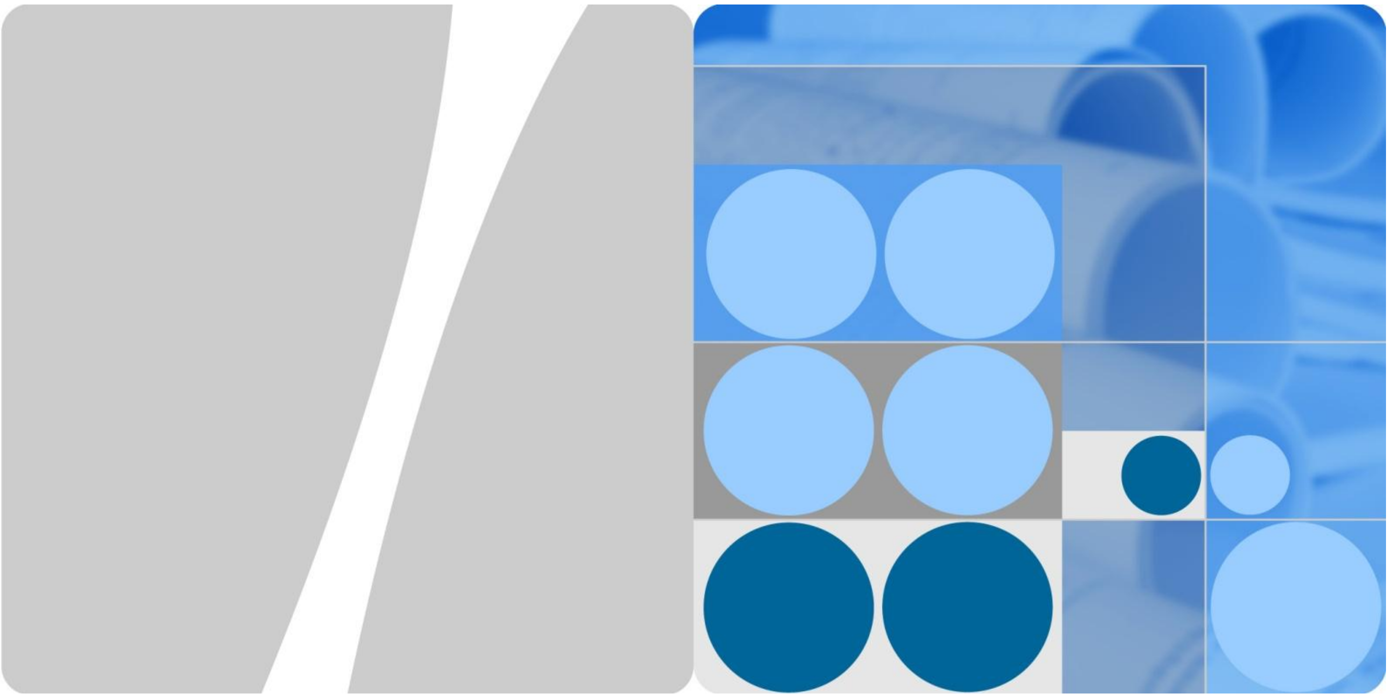


Huawei Switch

IEEE 1588v2 Technology White Paper

Issue 02

Date 2016-09-30

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http:// e.huawei.com

Contents

1 IEEE 1588v2 Overview

1.1 Technical Background

To meet wireless network access requirements, frequency synchronization precision must be ensured for base stations. Otherwise, mobile phones are easily disconnected or even cannot work when they are switched from a base station to another. Some wireless standards require time synchronization in addition to frequency synchronization. Table 1-1 lists frequency and time synchronization requirements of wireless systems using different standards.

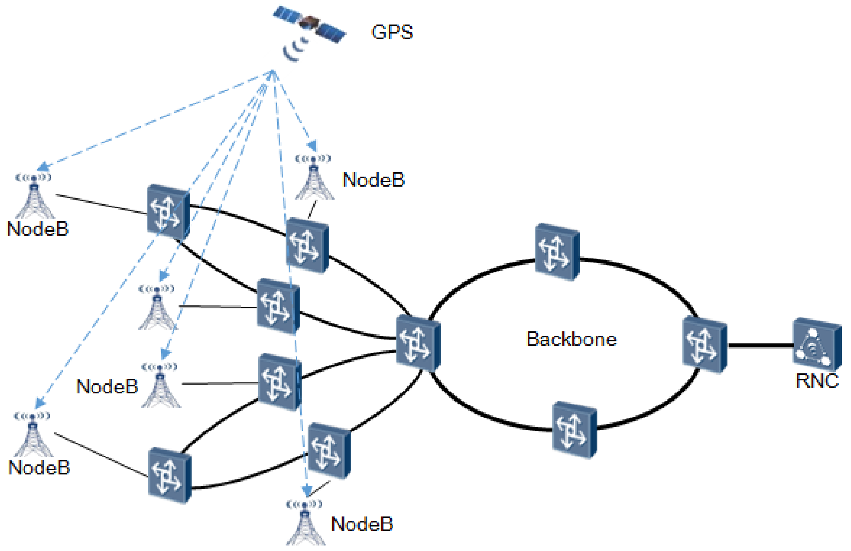
Table 1-1 Frequency and time synchronization requirements of wireless systems using different standards

Wireless Standards	Requirement on Frequency Synchronization Precision	Requirement on Time Synchronization Precision
GSM	0.05 ppm	NA
WCDMA	0.05 ppm	NA
TD-SCDMA	0.05 ppm	+/–1.5 µs
CDMA2000	0.05 ppm	+/–3 µs
WiMAX FDD	0.05 ppm	NA
WiMAX TDD	0.05 ppm	+/–0.5 µs
LTE FDD	0.05 ppm	NA (except for MB-SFN < +/–1 µs, LBS)
LTE TDD	0.05 ppm	+/–1.5 µs

Generally, WCDMA and LTE FDD standards use the frequency division duplex (FDD) mode that requires only frequency synchronization. The precision is 0.05 ppm. TD-SCDMA and LTE TDD standards use the time division duplex (TDD) mode that requires both frequency and time synchronization.

Traditional wireless system synchronization solution uses the Global Positioning System (GPS) to resolve frequency and time synchronization issues. GPS receivers need to be deployed in each base station. Figure 1-1 shows the GPS-based synchronization solution diagram.

Figure 1-1 GPS-based synchronization solution



As global wireless networks rapidly evolve from 2G to 3G and now towards Long Term Evolution (LTE), the GPS-based synchronization solution encounters the following challenges:

- High costs: Compared with 2G networks, 3G and LTE networks require multiplied base stations to cover the same area. The traditional solution of deploying GPS receivers at each base station leads to high deployment and maintenance costs.
- Difficulty in selecting a deployment site: Indoor base stations require long feeder cables, making it difficult to route feeder cables. When feeder cables are long, amplifiers must be used and feeder supplies need to be taken into consideration.
- High security risks: The GPS-based synchronization solution depends on the GPS. In emergency, an entire network may break down due to loss of synchronization. In addition, the GPS may fail to work at the present time.

To accommodate the high precision requirement of wireless systems and resolve the problems of the GPS solution, carriers are eager for a high-precision synchronization solution.

Currently, IP bearer networks support Synchronous Ethernet (SyncE), Network Time Protocol (NTP), and Precision Time Protocol (PTP) (IEEE 1588v2). Table 1-2 lists corresponding protocol attributes. SyncE supports frequency synchronization but not time synchronization, so SyncE is applicable to only wireless standards that have frequency synchronization requirements. NTP supports time synchronization. However, NTP provides only millisecond-level precision, which cannot meet precision requirements of wireless systems. PTP supports both frequency and time synchronization and can meet precision requirements of various wireless standards.

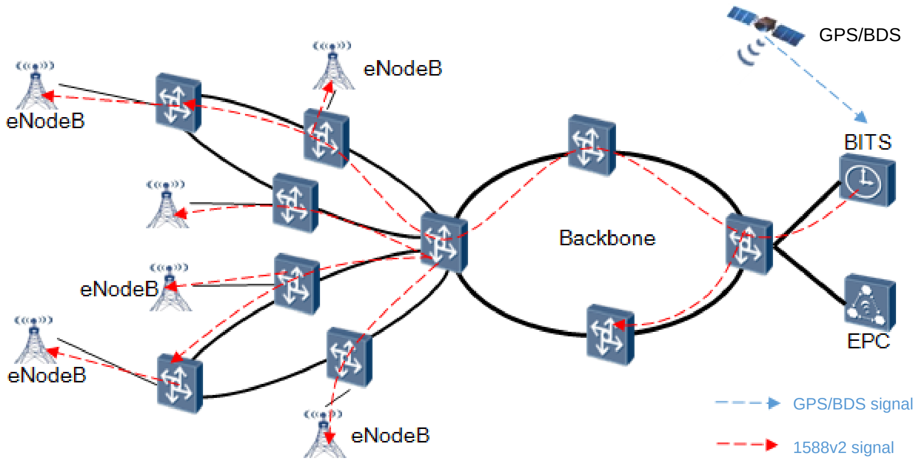
Table 1-2 Clock feature comparison

Standard	Frequency Synchronization	Frequency Synchronization Precision	Time Synchronization	Time Synchronization Precision
SyncE	Supported	Nanosecond-level	Not supported	NA
NTP	Not supported	NA	Supported	Millisecond-level
PTP	Supported	Nanosecond-level	Supported	Nanosecond-level

PTP is a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, which is also called IEEE 1588 (1588 for short). It enables standard Ethernet terminal devices to synchronize time and frequency from a clock source. 1588 includes 1588v1 and 1588v2. 1588v1 provides sub-millisecond precision time synchronization, and 1588v2 provides sub-microsecond precision time synchronization. 1588v2 is a time synchronization protocol. It was initially applied to high-precision time synchronization between devices and now is also used for frequency synchronization. 1588v1 has been replaced by 1588v2. Unless otherwise stated, PTP described in this document refers to 1588v2.

Figure 1-2 shows a typical 1588v2-based synchronization solution. The clock signal is injected to the transport network through the GPS or BeiDou Navigation Satellite System (BDS). The bearer equipment transmits the time information using the 1588v2 protocol, and the base stations obtain the time information from the bearer equipment through their PTP ports, thereby synchronizing them to the clock source. The time precision is within nanoseconds, which completely meets the synchronization requirement of radio base stations.

Figure 1-2 1588v2-based time synchronization solution



1.2 Technical Advantages

1588v2 supports time synchronization within the nanosecond-level precision. In contrast to the GPS solution, the 1588v2 solution achieves the same time synchronization precision but has advantages in terms of cost, maintenance, and security. It has become the most popular time transmission protocol in the industry with the following advantages:

- Low costs: 1588v2 reduces network construction and maintenance costs because no GPS receiver needs to be deployed or maintained for base stations.
- High precision: 1588v2 can provide nanosecond-level precision time synchronization with the assistance of hardware.
- Network transformation trend: 1588v2 is suitable for future integrated networks over IP.
- High security: 1588v2 can protect national security because it allows time synchronization without the GPS.

2 Technology Implementation

2.1 Synchronization Concepts

2.1.1 Frequency Synchronization

Frequency synchronization means that different signals have the same number of pulses within the same time period. It is independent of the pulse sequences and the start and end time of each pulse. Figure 2-1 is used for illustration. If signal 1 has four pulses (1, 2, 3, and 4) while signal 2 has only three pulses (3, 4, and 5) within each one second, the frequencies of the two signals are not synchronized to each other or the two signals have bad frequency synchronization. If signal 1 has four pulses (1, 2, 3, and 4) and signal 2 also has four pulses (3, 4, 5, and 6) within each one second, the frequencies of the two signals are synchronized to each other or the two signals have good frequency synchronization. According to the preceding description, frequency synchronization only concerns the number of pluses in different signals within the same time period. It is independent of the pulse sequences and the start and end time of each pulse and the pulses in the two signals can start or end at different time.

Figure 2-1 Bad frequency synchronization

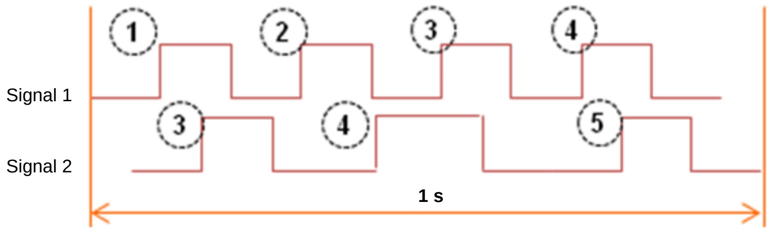
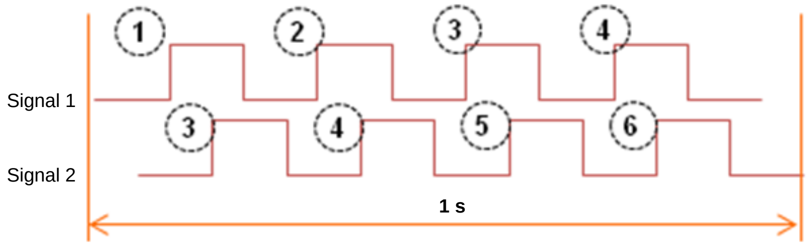


Figure 2-2 Good frequency synchronization



2.1.2 Phase Synchronization

Phase synchronization means that different signals have the same frequency and the same pulse start and end time. It is independent of the pulse sequences. Figure 2-3 and Figure 2-4 are used for illustration. Signals 1 and 2 have the same frequency and the same number of pulses within a second. If the pulse start time and end time of the two signals differ (the rising and falling edges of the two signals are not aligned), for example, pulse 1 of signal 1 and pulse 3 of signal 2 in Figure 2-3, the two signals have bad phase synchronization. If the pulse start time and end time of the two signals are the same, for example, the first pulse of signal 1 and the third pulse of signal 2 in Figure 2-4, the two signals will start and end at the same time. In other words, the rising and fall edges of the two signals are well aligned and the two signals have good phase synchronization.

Figure 2-3 Bad phase synchronization

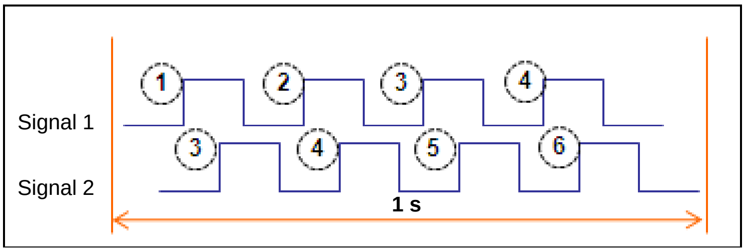
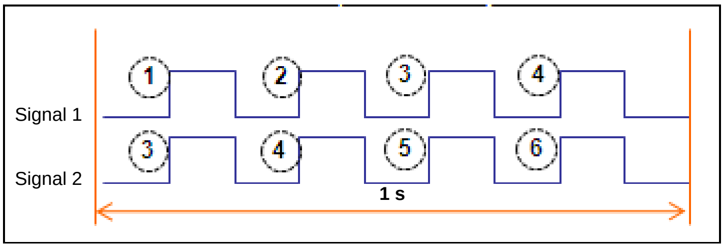


Figure 2-4 Good phase synchronization



2.1.3 Time Synchronization

Time synchronization indicates that two signals have the same frequency and phase, and the same sequence of pulse signals. Figure 2-5 and Figure 2-6 are used for illustration. Signals 1 and 2 have the same frequency and the pulses in the two signals appear in the same sequence.

In other words, signals 1 and 2 are transmitted in the sequence of pulses 1, 2, 3, and 4. If their pulse phases are not fully synchronized, their time is not synchronized, as shown in Figure 2-5. Figure 2-6 shows good time synchronization.

Figure 2-5 Bad time synchronization

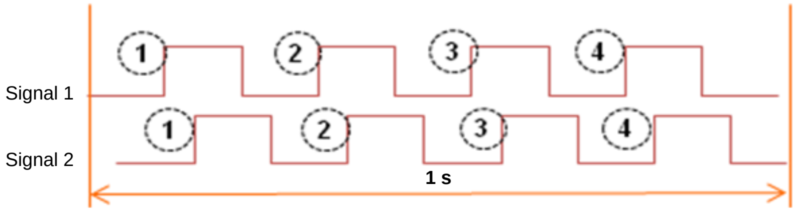
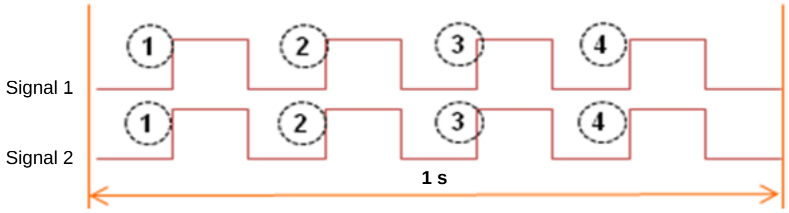


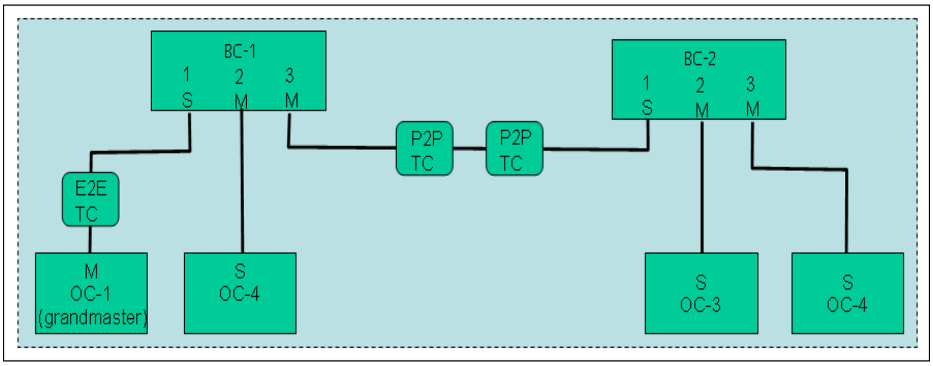
Figure 2-6 Good time synchronization



2.2 1588v2 Device Models

The 1588v2 standard defines five network node models: ordinary clock (OC), boundary clock (BC), end-to-end transparent clock (E2E TC), peer-to-peer transparent clock (P2P TC), and management node. Figure 2-7 shows four node models used for synchronization. The management node only manages synchronization nodes but does not provide the synchronization function.

Figure 2-7 1588v2 device models



1. OC: An OC provides only one physical port to communicate with the network and works as either a clock source or clock sink. An OC is a pure clock source or clock sink. It advertises time to downstream clocks or synchronizes time with its upstream clock.

When an OC works as the system clock source, it is also called the grandmaster clock (GMC). The GMC functions as the reference clock, which is the highest-stratum clock of the entire system. Time of the GMC will be synchronized to the system through 1588v2 message interactions between clock nodes. The GMC can be statically configured or dynamically elected through the best master clock (BMC) algorithm.

2. BC: A BC provides multiple physical ports to communicate with the network. Functions of these ports are similar to those of the network communication port on an OC. A BC is similar to a time regenerator, and is a mixture of the two types of the OC. The BC can recover the clock, and can act as the source clock for the downstream devices. A BC is a clock node in the middle position. One port of the BC receives clock signals from an upstream device, and the other ports send the clock signals to downstream devices.
3. TC: A TC does not recover time or frequency. It does not terminate 1588v2 messages except signal and management message; instead, it only corrects forwarding delays in these 1588v2 messages. A TC on a network only provides support for processing 1588v2 messages, and does not recover time or frequency. Therefore, the TC does not need to support the BMC algorithm. The TC is available in end-to-end TC (E2E TC) and peer-to-peer TC (P2P TC).

   E2E TC: An E2E TC provides several ports, forwards all 1588v2 messages, and measures and corrects the residence time of PTP event messages traversing the E2E TC.

   P2P TC: A P2P TC has multiple ports. In addition to functions of an E2E TC, a P2P TC is also able to measure and correct the link delay between each port and a similarly equipped port on another node sharing the link, that is, the link peer.

   The residence time is obtained by calculating the interval between receiving a message to sending the message. A P2P TC obtains link delay information through the Pdelay mechanism. For details, see section 2.4.2 "1588v2-based Time Synchronization."
4. Management device: A management device provides multiple PTP message management ports.

Additionally, Huawei provides the following extended device types:

1. TC+OC: A TC+OC provides multiple PTP ports. One of the ports is configured as the OC to recover the clock frequency (TC+OC does not recover time). The other ports are configured as pure TCs (E2E or P2P) to transparently transmit messages. The OC needs to be configured with a clock source, which is used as the reference clock source of the TC+OC. The TC+OC is synchronized to the master clock. The TC works in free-running mode. TC+OC nodes are classified into the E2E TC +OC and P2P TC +OC.

   E2E TC+OC: An E2E TC+OC has multiple ports. In addition to functions of an E2E TC, an E2E TC+OC can have one port configured as the OC for frequency synchronization.

   P2P TC+OC: A P2P TC+OC has multiple ports. In addition to functions of a P2P TC, a P2P TC+OC can have one port configured as the OC for frequency synchronization.
2. TCandBC: A TCandBC has multiple PTP ports. One of the ports is configured as the BC to recover the frequency and time. The other ports are configured as pure TCs (E2E or P2P) to transparently transmit messages. The reference clock source of a BC is selected through the BMC algorithm.

2.3 1588v2 Message

2.3.1 1588v2 Message Types

The 1588v2 standard defines event messages and general messages. Accurate timestamps are added to event messages at the device ingress and egress; no timestamp will be added to general messages.

Table 2-1 1588v2 message types

Message Category	Message Type	Description
Event message	• Sync • Delay_Req • Pdelay_Req • Pdelay_Resp	Event message is tagged with a timestamp when reaching or leaving a port. PTP devices calculate the link delay based on the timestamps carried in event messages.
General message	• Announce • Follow_Up • Delay_Resp • Pdelay_Resp_Follow_Up • Management • Signaling	General message is used to establish a master-slave hierarchy, and to request and send time information. General messages are not tagged with timestamps.

The Sync, Delay_Req, Follow_Up, and Delay_Resp messages are used to generate and communicate the timing information needed to synchronize OCs and BCs using the delay request-response mechanism.

The Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages are used to measure the link delay between two clock ports implementing the Pdelay mechanism. The link delay is used to correct timing information in Sync and Follow_Up messages in systems composed of P2P TCs. OCs and BCs that implement the Pdelay mechanism can perform synchronization using the measured link delays and the timestamp information in the Sync and Follow_Up messages.

The Announce messages are used to establish the synchronization hierarchy. That is, information related to the BMC algorithm is carried in the Announce messages.

The Management messages are used to query and update the PTP data sets for clock maintenance. These messages are also used for PTP system customization, initialization, and fault management. Management messages are used between the management node and clock equipment.

The Signaling messages are used for communication between clocks for all other purposes. For example, signaling messages can be used for negotiation of the rate of unicast messages between a master clock and its slave clocks.

2.3.2 1588v2 Message Encapsulation

MAC encapsulation and user datagram protocol (UDP) encapsulation can be used for 1588v2 message encapsulation. UDP encapsulation includes IPv4 encapsulation and IPv6 encapsulation. You can configure unicast or multicast encapsulation for PTP messages based on the topology of the master and slave devices:

- Unicast encapsulation: applies to point-to-point clock signal synchronization. This mode is used when a single device needs to synchronize with the master clock. If the destination MAC address or IP address of PTP messages to be sent is set to a unicast address, PTP messages are encapsulated in unicast mode.
- Multicast encapsulation: applies to point-to-multipoint clock signal synchronization. By default, devices that reside in the same PTP domain and use the same delay measurement mechanism join the same multicast group. Therefore, when multicast encapsulation is used, the destination MAC address or IP address does not need to be configured for PTP messages.

MAC encapsulation for 1588v2 messages: The Ethernet type of a message is 0x88F7. In the multicast encapsulation situation, the default multicast destination MAC address is 01-1B-19-00-00-00 when the delay request-response mechanism is used; the default multicast destination MAC address is 01-80-C2-00-00-0E when the Pdelay mechanism is used.

IPv4 UDP encapsulation for 1588v2 messages: The UDP destination port for event messages is port 319 and the UDP destination port for general messages is port 320. In the multicast encapsulation situation, the default destination IP address is multicast address 224.0.1.129 when the delay request-response mechanism is used; the default destination IP address is multicast address 224.0.0.107 when the Pdelay mechanism is used.

2.4 1588v2-based Synchronization Implementation

After the master-slave hierarchy is established, the frequency and time synchronization process starts. PTP was initially used to implement high-precision time synchronization but now can also be used for frequency synchronization between devices.

PTP calculates the link delay and time difference between the master and slave devices based on the timestamps generated when the devices exchange event messages. PTP then synchronizes the time and frequency between the master and slave devices. Timestamps can be carried in PTP messages in either one-step or two-step mode.

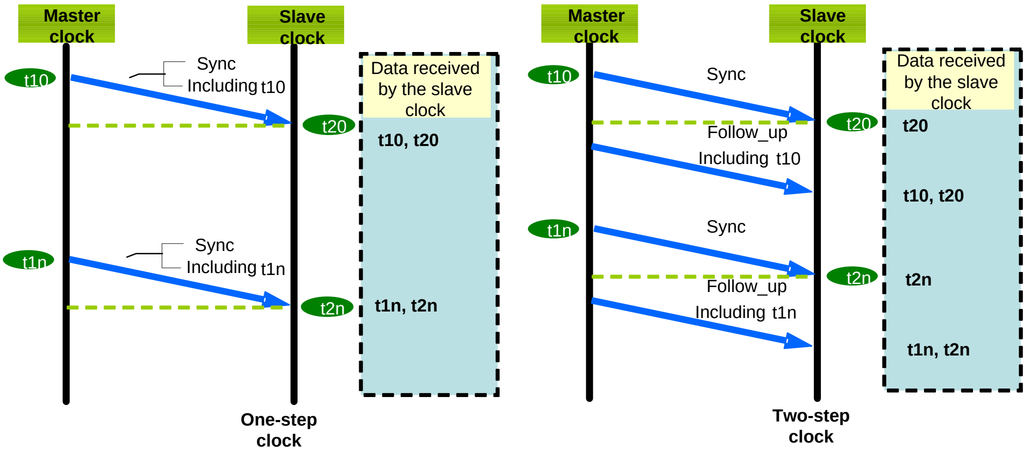
- One-step clock mode: An event message (Sync or Pdelay_Resp message) carries the timestamp when the message is sent. Time information is generated and advertised when the message is sent and received.
- Two-step clock mode: An event message (Sync or Pdelay_Resp message) does not carry the timestamp when the message is sent. Instead, subsequent general messages (Follow_Up and Pdelay_Resp_Follow_Up messages) carry the timestamps when Sync and Pdelay_Resp messages are sent. In two-step clock mode, time information is generated and then advertised. This mode applies to the devices that cannot add timestamps to event messages.

2.4.1 1588v2-based Frequency Synchronization

1588v2 implements frequency synchronization by exchanging Sync messages between master and slave clocks. In Figure 2-8, the master clock sends a Sync message at t10. If the master clock works in one-step clock mode, t10 is sent to the slave clock through the Sync message. If the master clock works in two-step clock mode, t10 is sent to the slave clock through the subsequent Follow_Up message. The two modes use the same frequency synchronization principle.

The master clock periodically sends Sync messages to the slave clock. If the slave clock frequency is synchronized to the master clock frequency, then the accumulative time errors within the same time periods are the same, as long as the path delay changes are neglected. In other words, t21 - t20 = t11 - t10, t22 - t21 = t12 - t11, t23 - t22 = t13 - t12 … t2n - t20 = t1n - t10. If t2n-t20 is greater than t1n-t10, the slave clock frequency is higher than the master clock frequency and needs to be reduced. If t2n-t20 is smaller than t1n-t10, the slave clock frequency is lower than the master clock frequency and needs to be increased.

Figure 2-8 1588v2-based frequency synchronization



2.4.2 1588v2-based Time Synchronization

1588v2 calculates the mean path delay and time difference between the master and slave devices based on the timestamps generated when the devices exchange event messages. 1588v2 then synchronizes the time between the master and slave devices. Measurement of mean path delay includes two mechanisms: Delay-Req and Pdelay.

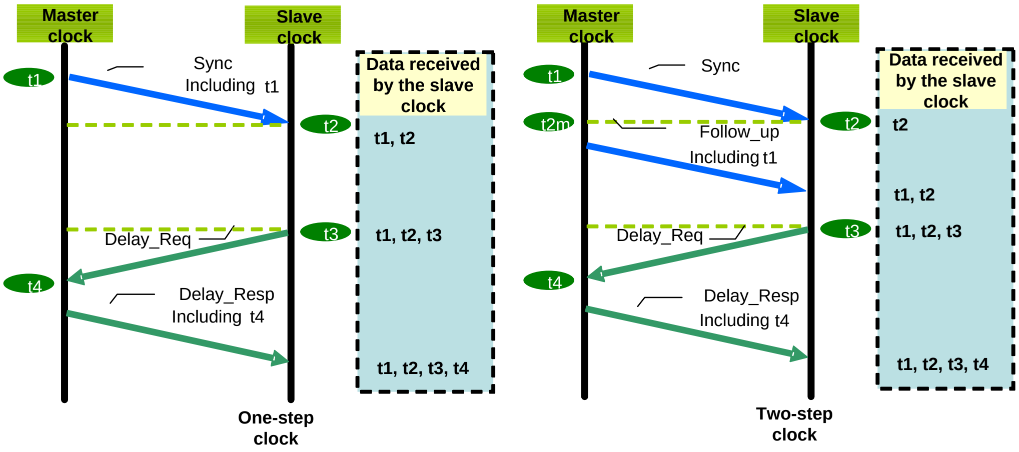
Delay-Req mechanism: represents the delay request-response E2E mechanism. The mechanism calculates the time difference based on the total delay of links between the master and slave clocks.

Pdelay mechanism: represents the peer delay P2P mechanism. The mechanism calculates the time difference based on the delay of each link between the master and slave clocks.

- **Measurement of mean path delay based on the Delay-Req mechanism**

Figure 2-9 shows how the Delay-Req mechanism calculates the mean path delay between the master and slave clocks.

Figure 2-9 Measurement of mean path delay based on the Delay-Req mechanism



1. The master clock sends a Sync message at t1. If the master clock works in one-step clock mode, t1 is sent to the slave clock through the Sync message. If the master clock works in two-step clock mode, t1 is sent to the slave clock through the subsequent Follow_Up message.
2. The slave clock receives the Sync message at t2 and obtains t1 from the Sync message in one-step mode or Follow_Up message in two-step mode.
3. The slave clock sends a Delay_Req message to the master clock at t3.
4. The master clock receives the Delay_Req message at t4.
5. The master clock sends a Delay_Resp message recording t4 to the slave clock.

In the preceding process, devices generate the timestamps when the messages are received or sent based on their system clocks. 1588v2 defines the timestamp length as 80 bits in a message.

Assume that the transmission delay from the master clock to the slave clock is Delayms, the transmission delay from the slave clock to the master clock is Delaysm, and the time difference between the two nodes is offsetFromMaster. All of them are variables:

- t2 - t1 = Delayms + offsetFromMaster
- t4 - t3 = Delaysm - offsetFromMaster

After conversion, the following equations are given:

- Delayms + Delaysm = (t4 - t1) – (t3 - t2)
- offsetFromMaster = [(t2 - t1) + (t3 - t4) - (Delayms - Delaysm)]/2

If delays of the links receiving and sending messages between the master and slave clocks are symmetrical, or Delayms equals Dlaysm, the following equation is tenable:

Mean path delay:

meanPathDelay = (Delayms + Delaysm)/2 = [(t4 – t1) - (t3 – t2)]/2

Time difference:

offsetFromMaster = [(t2 - t1) + (t3 - t4)]/2

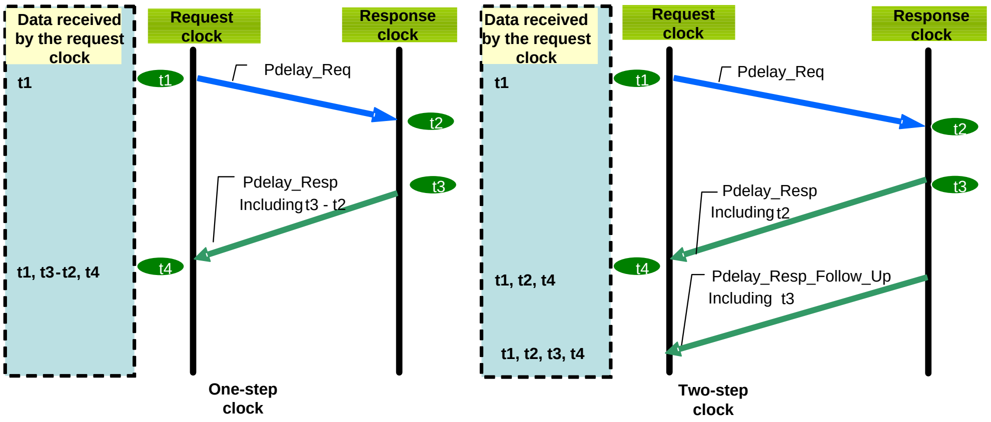
- **Measurement of mean path delay based on the Pdelay mechanism**

The Pdelay mechanism calculates the transmission time (link delay) between two communication ports that support the P2P mechanism using Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages. The Pdelay mechanism is similar to the Delay-Req mechanism except that it measures the delay of each link and transmits the link delay and residence time of a Sync message on a TC to downstream devices in a Pdelay_Resp or Pdelay_Resp_Follow_Up message. The Pdelay mechanism calculates the mean path delay of the master and slave nodes based on the delay of each link and residence time on the TC.

In the Pdelay mechanism, the link delay can be measured between two connected ports that support the Pdelay mechanism regardless of their port states.

Figure 2-10 shows how the Pdelay mechanism calculates the mean path delay between clocks.

Figure 2-10 Measurement of mean path delay based on the Pdelay mechanism



1. The request clock sends a Pdelay_Req message at t1.
2. The response clock receives the Pdelay_Req message at t2.
3. The response clock sends a Pdelay_Resp message at t3. If the response clock works in one-step mode, t3-t2 is sent to the request clock through the Pdelay_Resp message. If the response clock works in two-step mode, t2 is sent to the request clock through the Pdelay_Resp message and t3 is sent to the request clock in the subsequent Pdelay_Resp_Follow_Up message.
4. The request clock receives the Pdelay_Resp message at t4.

In the preceding process, devices generate the timestamps when the messages are received or sent based on their system clocks. 1588v2 defines the timestamp length as 80 bits in a message.

The request clock obtains four timestamps (t1, t2, t3, and t4) or three timestamps (t1, t3-t2, and t4), and calculates the mean path delay between the request and response clocks.

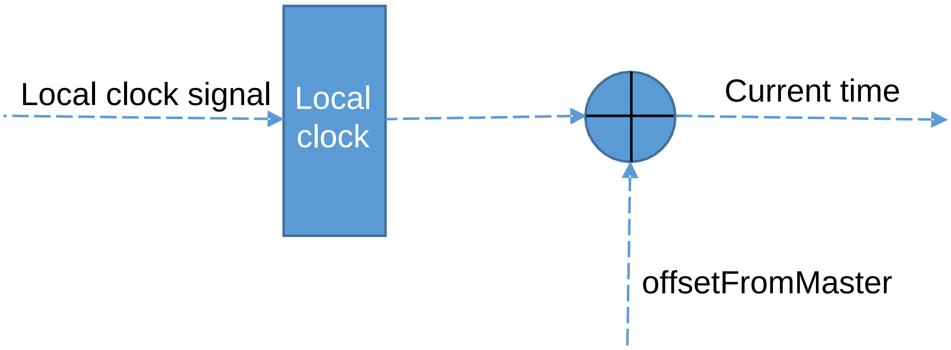
Mean path delay:

meanPathDelay = [(t4 – t1) – (t3 – t2)]/2

- **Time difference calculation**

The slave device adjusts its local time according to the time difference to maintain synchronization with the master device.

Figure 2-11 1588v2 time calibration diagram



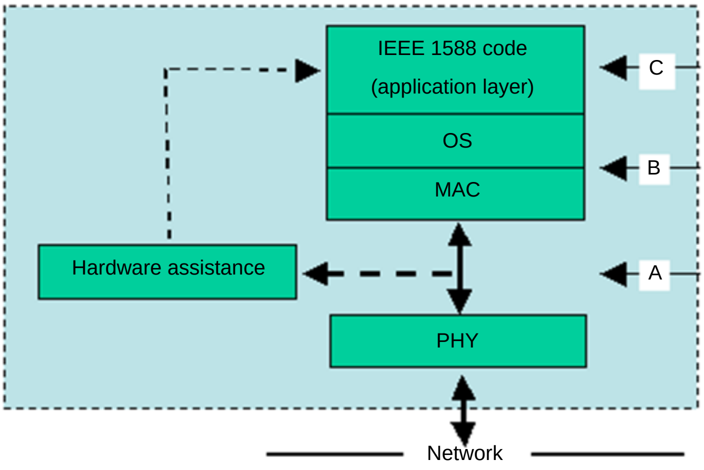
The above figure shows how to implement time synchronization using 1588v2 messages. On a 1588v2 synchronization system, the path delay and residence time must also be considered in time synchronization.

2.5 1588v2 Clock Timestamp Generation

When a PTP message traverses the protocol stack on a node, the timestamp of the message is selected among the reference points defined by the protocol stack (A, B, C in the following figure). The closer a reference point is to the actual physical connection point, the smaller the timing error is. Point A in the following figure is the best reference point. According to 1588v2, timestamps are added between the MAC layer and the PHY layer. This helps effectively prevent uncertain delays introduced by the protocol layer and achieve nanosecond synchronization precision, thereby meeting the high-precision time synchronization requirements of telecommunications networks.

The 1588v2 clock timestamp records the time of the 1588v2 event messages passing the timestamp point of the device physical port. The time is indicated by an 80-bit value, including 48-bit seconds and 32-bit nanoseconds.

Figure 2-12 Timestamp position



2.6 Master-Slave Hierarchy Establishment

The clock synchronization process consists of three phases:

1. Establish the master-slave hierarchy, select the GMC, and negotiate the master/slave state of ports.
2. Synchronize the frequency of the slave node with that of the master node.
3. Synchronize the time of the slave node with that of the master node.

2.6.1 BMC Implementation

The BMC algorithm, defined in 1588v2, is used to determine the master-slave hierarchy among clocks on a network. The BMC algorithm divides clocks on a network into master clocks and slave clocks. Slave clocks trace the frequency or time of master clocks. If the network or a clock source attribute changes, the best master clock is re-selected using the BMC algorithm so that clocks on the entire network are synchronized.

In terms of networks, the BMC algorithm establishes the master-slave hierarchy of clocks on a network, which is like a tree with the grandmaster as the root. The grandmaster is the best clock source on the entire network.

In terms of nodes, the BMC algorithm determines the master clock for each clock node. The BMC algorithm determines the best master clock by comparing the clock signals contained in Announce messages received from different ports and the local clock signals. If the local clock is selected as the best master clock, the local clock functions as the grandmaster. If an external clock is selected as the best master clock, the local clock traces the master clock.

The BMC algorithm compares the following data set information contained in Announce messages to select the best master clock and determine the PTP port states:

- Priority1: indicates clock priority 1. The value is configurable and ranges from 0 to 255. A smaller value indicates a higher priority.
- ClockClass: defines the capability to trace the clock time or frequency International Atomic Time (TAI).
- ClockAccuracy: defines the precision of a clock. A smaller value indicates higher precision.
- OffsetScaledLogVariance: defines the stability of a clock.
- Priority2: indicates clock priority 2. The value is configurable and ranges from 0 to 255. A smaller value indicates a higher priority.

When the BMC algorithm is used by a PTP device for best master clock selection, Priority1 of each candidate clock source is compared first, then ClockClass, ClockAccuracy, OffsetScaledLogVariance, and Priority2. If Priority1 of candidate clock sources is the same, ClockClass is compared, and so on. The clock source with the highest priority, class, and precision is selected as the best master clock.

You can change the preceding attributes to influence the way the PTP master clock is selected, and finally select the clock signals to receive. The BMC algorithm can allocate PTP clock sources and implement protection switching.

2.6.2 Process of Establishing a Master-Slave Hierarchy

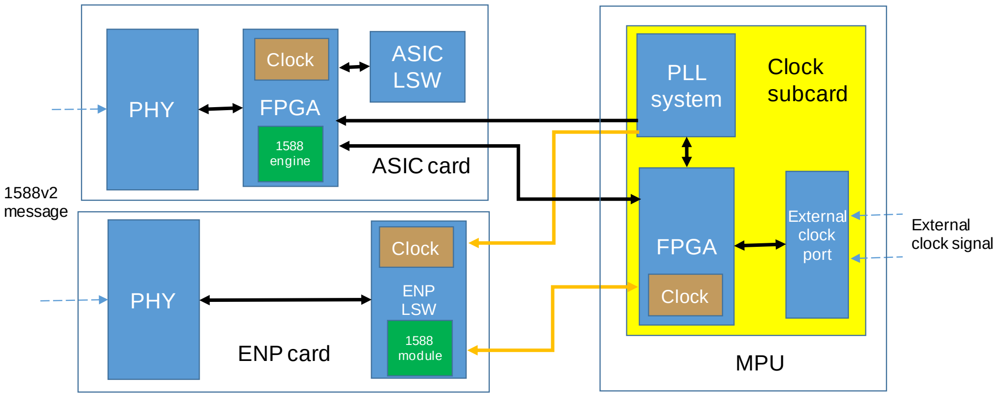
In a PTP system, ports receive and send Announce messages to exchange data sets and port state machine information. The BMC algorithm compares data sets and port state machine information to establish the master-slave hierarchy. The establishment process is as follows:

1. A port receives and processes the Announce message from the peer port.
2. The local device uses the BMC algorithm to determine the best master clock and recommended state of the local port, namely, master, slave, or passive.
3. The local device updates data sets based on the recommended port state.
4. The local device determines the actual port status based on the port state machine and establishes the master-slave hierarchy based on the recommended state and the state decision event. The state decision event is generated when an Announce message is received or when no Announce message is received within the specified timeout period. When the interval at which a port receives an Announce message is longer than the Announce message timeout period, the state of the port is set to master.

2.7 Switch Capability

Figure 2-13 shows the hardware structure of a modular switch that supports 1588v2. A modular switch can support 1588v2 only when it is configured with a main processing unit (MPU) that provides a clock subcard and a line card that provides the clock function. Cards can be classified into application-specific integrated circuit (ASIC) cards and Ethernet Network Processor (ENP) cards. An ASIC card provides a field programmable gate array (FPGA) logical module to process 1588v2 messages. An ENP card directly uses an ENP chip to process 1588v2 messages.

Figure 2-13 Hardware structure of a switch that supports 1588v2



The 1588v2-based time synchronization process is as follows:

1. A 1588v2 message is sent to the PHY.
2. The 1588v2 engine of the logical FPGA module on the ASIC card or the 1588v2 module on the ENP card identifies the 1588v2 message, and adds a timestamp to the message based on the local clock time.
3. The timestamp is sent to the FPGA logical module on the clock subcard through the channel between the line card and the clock subcard. The FPGA module resolves the timestamp and sent the timestamp to the phase-locked loop (PLL) system.
4. The PLL system calculates the time difference between master and slave devices based on the timestamp information, and sends the time difference information to the local FPGA logical module.
5. The local FPGA logical module calibrates the local time after receiving the time difference information sent by the PLL system.

6. The PLL system sends the time information to the FPGA logical module on the ASIC card or to the ENP card.
7. The FPGA logical module or the ENP card updates the local clock time after receiving the time information, implementing time synchronization between the network element (NE) clock with the external clock.

If the external clock signal is input, the device will work in the master state. The time synchronization process is as follows:

1. The external time information is sent to the FPGA logical module on the clock subcard through the external clock port.
2. The clock subcard updates its local clock time based on the external time.
3. The clock subcard sends the time information to the FPGA logical module on the ASIC card or to the ENP card.
4. The FPGA logical module or the ENP card updates the local clock time after receiving the time information, implementing time synchronization between the NE clock with the external clock.

Currently, modular switches but not fixed switches support 1588v2. The modular switches must be configured with an MPU that provides a clock subcard and a line card that provides the clock function.

There is only one clock subcard (CKM) model. An MPU that supports this clock subcard needs to be configured. If a device uses double MPUs, clock subcards need to be inserted into both of the MPUs.

ASIC cards that provide the clock function include only the S24CEA0. ENP cards that provide the clock function include multiple series, including X1E, X2E, and X2S series cards.

A modular switch in a stack does not support 1588v2.

3 Typical Networking

Frequency synchronization is a precondition for time synchronization and can be implemented through SyncE and 1588v2. There are two combination solutions for time synchronization:

1. SyncE + 1588v2: SyncE implements frequency synchronization and 1588v2 implements time synchronization.
2. 1588v2 + 1588v2: 1588v2 implements both frequency and time synchronization.

1588v2-based time synchronization can be implemented in BC and TC modes. The TC mode is not recommended because of its disadvantages:

1. Poor reliability: TC nodes do not use the BMC algorithm. Therefore, 1588v2 path protection can only be ensured through service path protection. In addition, TC nodes do not support automatic path switching based on the link status and quality level.
2. Complex maintenance:
    - Protocol fault: TC nodes do not resolve messages and cannot detect protocol faults. They only provide transmission channels and calculate the residence time.
    - Performance fault: TC nodes calculate only the residence time. However, the residence time is randomly changing. Whether the calculated residence time is accurate cannot be measured. When a fault occurs on a downstream BC or OC node, the faulty NE cannot be located.
3. High bandwidth load: Messages sent by TC nodes need to be replicated and transmitted. All Delay_Req/Delay_Resp messages sent by E2E TC nodes are aggregated to the upstream OC or BC node, consuming high bandwidth on the OC or BC node. This causes service message loss.

The SyncE + 1588v2 time synchronization solution has the following advantages:

1. Good compatibility and mature SyncE: SyncE evolves from synchronous digital hierarchy (SDH) and has been widely deployed on live networks. In addition to base stations, fixed network operators or integrated operators need to provide the synchronization service for other businesses or applications. However, these businesses may require only frequency synchronization but not time synchronization. Moreover, the number of SyncE-capable devices is larger than that of 1588v2-capable devices.
2. Higher reliability: In SyncE + 1588v2 networking mode, if the 1588v2 clock source or a 1588v2 link fails, NEs can use SyncE to maintain time precision. In 1588v2 + 1588v2 networking mode, if the 1588v2 clock source fails, NEs have to rely on themselves to maintain time precision. However, the frequency precision of NEs is far less than the frequency synchronization precision provided by SyncE.

The 1588v2 + 1588v2 time synchronization solution (BC mode) has the following advantages:

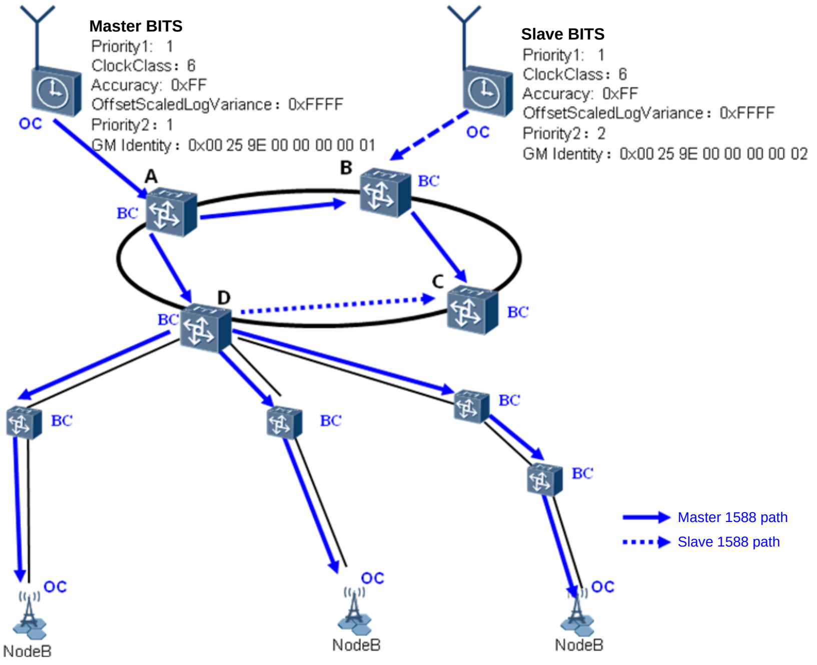
1. Easy network planning: 1588v2 can implement automatic synchronization path calculation through BMC, avoiding loops.
2. Easy maintenance: The SyncE + 1588v2 solution uses different synchronization protocols to achieve frequency and time synchronization. In this solution, frequency and time synchronization paths are different. The 1588v2 + 1588v2 solution uses the same protocol to achieve frequency and time synchronization. In this solution, frequency and time synchronization paths are the same.

3.1 1588v2-based Frequency and Time Synchronization (BC Mode)

Figure 3-1 shows how 1588v2 implements frequency and time synchronization on a network.

- Two building integrated timing supply (BITS) devices function as OCs to inject the clock source to switches. The BMC algorithm is used to determine the GMC.
- All the devices on the synchronization path must support the 1588v2 function.
- The switches work in the BC mode and recover time hop by hop. Base stations function as OCs and obtain time information from the switches.
- Only the BMC algorithm is used for synchronization. Therefore, frequency and time synchronization paths are the same.
- The minimum rates of sending Sync, Delay_Req/Delay_Resp, and Announce messages are 128/s, 128/s, and 8/s, respectively.

Figure 3-1 1588v2-based network-wide synchronization

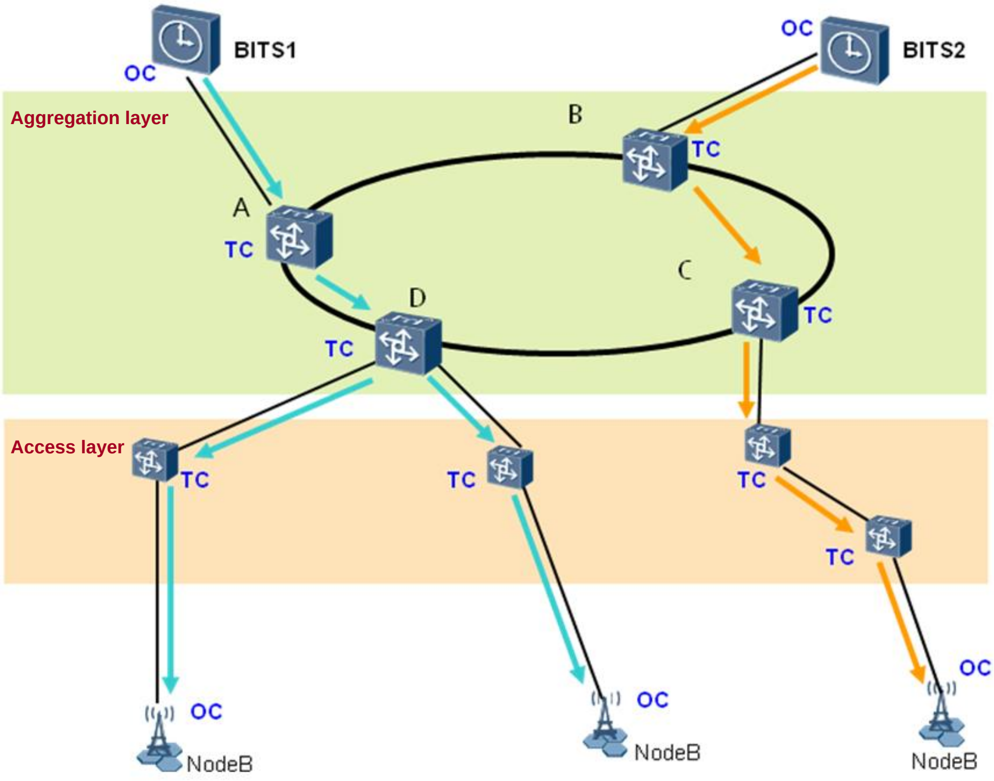


BC synchronization on the entire network is advantageous in that every site recovers time, and the synchronization precision can be measured hop by hop, which facilitates maintenance and fault locating. In addition, the BMC algorithm is used on the BCs to ensure rapid protection switching between BITS reference sources and tracing paths.

3.2 1588v2-based Frequency and Time Synchronization (TC Mode)

In the transparent transmission scenario, bearer network devices act as TCs and do not require time recovery. They only calculate the residence time that 1588 messages take to traverse the TCs. That is, in TC mode, time of bearer network devices is not synchronous with that of the BITSs. Base stations which act as slave OCs recover time through 1588 messages and synchronize the time to the BITSs.

Figure 3-2 1588v2-based transparent transmission of time



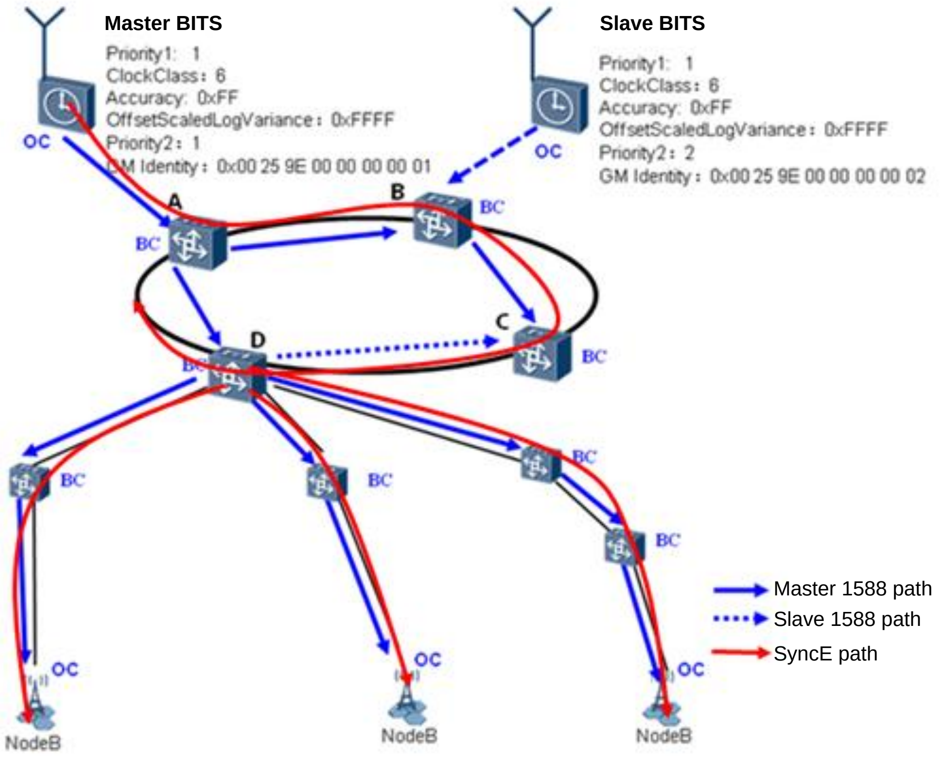
In TC mode, bearer network devices can transparently transmit time of different domains. The BITS1 and BITS2 shown in Figure 3-2 can belong to different carriers or time domains. Time of the two paths can be different. Bearer network devices do not need to support the BMC algorithm, which raises a low requirement for device software. However, because the TC nodes do not recover time and only calculate the residence time which is variable, it is difficult to determine whether the calculated residence time is correct. Therefore, maintenance and fault locating become difficult. Because TC nodes do not use the BMC algorithm, they cannot automatically switch to another path when the original TC path on the bearer network is faulty. A service-layer protection protocol other than the 1588 protocol must be used for TC path protection.

3.3 SyncE–based Frequency Synchronization and 1588v2-based Time Synchronization (BC Mode)

Figure 3-3 shows how SyncE implements frequency synchronization and 1588v2 implements time synchronization on a network.

- The BITSs obtain the clock source from the GPS or BDS.
- Devices requiring frequency synchronization obtain the reference clock source from the deployed SDH network or SyncE network.
- Devices requiring time synchronization obtain the reference clock source from the BITSs.
- Because Synchronization Status Message (SSM) and BMC are different algorithms, the frequency and time synchronization paths are not completely the same.
- When a BITS fails, the network can use SyncE to maintain time precision.

Figure 3-3 SyncE-based frequency synchronization and 1588v2-based time synchronization



A Acronyms and Abbreviations

Acronyms and Abbreviations	Full Spelling
OC	Ordinary Clock
BC	Boundary Clock
TC	Transparent Clock
P2P	Peer to Peer
E2E	End to End
BMC	Best Master Clock
PTC	Precision Time Protocol
CUT	Coordinated Universal Time
PRC	Prime Reference Clock
GPS	Global Positioning System
PPM	Parts per Million
PLL	Phase Lock Loop
RTC	Real Time Clock
GMC	Grandmaster Clock
SSM	Synchronization Status Message