SyncE/1588v2 Basics

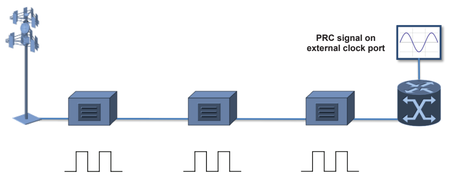
SyncE

ITU-T G.826x Recommendations

The standards related to timing and synchronization in Packet Networks are:

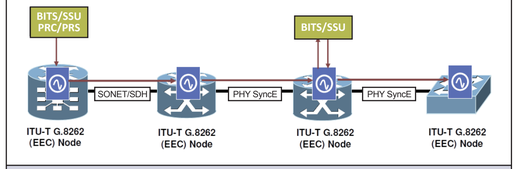
G.8261 (04/2008)	Timing and Synchronization Aspects in Packet Networks provides an architecture and presents Sync-E requirements
G.8262 (10/2010)	Timing Characteristics of a Synchronous Ethernet Equipment Slave Clock specifies clock requirements
G.8264 (10/2010)	Distribution of Timing Information Through Packet Networks outlines the Synchronous Status Message (SSM), which is used to report the quality level of the transmitting clock to other Network Elements (NEs)
G.8265 (10/2010)	Architecture and Requirements for Packet Based Frequency Delivery
G.8265.1 (10/2010	Precision Time Protocol Profile for Frequency Synchronization

SyncE Functionality



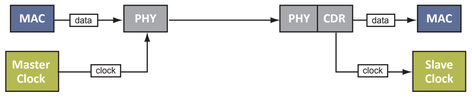
- Similar to traditional TDM or SONET/SDH timing distribution
- Frequency information is transmitted via the Physical Interface/Layer (PHY)
- Quality of clock distribution is “indifferent” to Layer 2 phenomena such as traffic load, jitter, packet delay, and packet loss
- The PHY clock is used to generate the clock signal from the “bit stream”
- Considered to be an End-to-End synchronization technique

Physical Layer Synchronization



- Equivalent to SDH/SONET Synchronization Architecture
  - Able to maintain G.803 synchronization chain
- Like SDH/SONET, SyncE uses a Physical Layer (PHY) Synchronization method
  - G.8261 defines Synchronous Ethernet clock performance limits
- Extend previous ITU-T (and Telcordia) node clock recommendations
  - G.8262 defines synchronous Ethernet Equipment Clock (EEC)
- Ethernet Slow Protocol to extend the SSM traceability function
  - G.8264 defines ESMC (Ethernet Synchronization Messaging Channel) to support SSM

Timing Overview

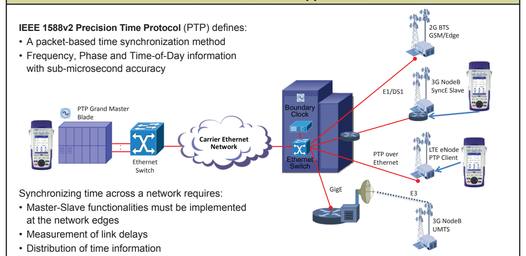


- Synchronization is based on a Primary Reference Clock (PRC) injected at the Tx end
- All nodes must have a clock source traceable to a Primary Reference Clock (PRC)
  - The Clock source can be derived from incoming data OR an independent clock source
  - An external Phase Lock Loop (PLL) can be used for frequency correction on each local node
- Clock is distributed by way of downstream node clock recovery
- OAM PDUs (protocol data units) pass SSM (synchronization status message)
- 8B/10B encoding facilitates clock recovery in optical links
- Phase Lock Loop recovers clock from received data and helps reduce jitter propagation in noisy network environments

IEEE 1588v2

IEEE 1588v2/PTP Test Application

IEEE 1588v2 Precision Time Protocol (PTP) defines:
- A packet-based time synchronization method
- Frequency, Phase and Time-of-Day information with sub-microsecond accuracy

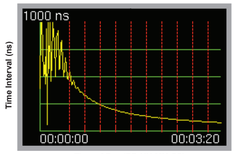


Synchronizing time across a network requires:
- Master-Slave functionalities must be implemented at the network edges
- Measurement of link delays
- Distribution of time information

PDV Analysis

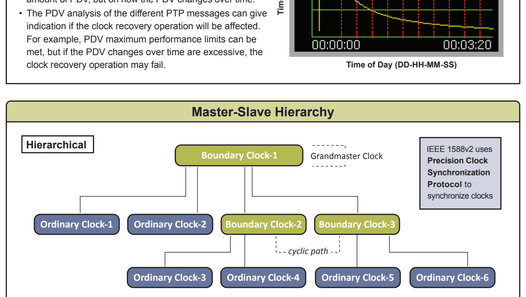
Packet Delay Variation (PDV) is the difference in packet delay from the master to the slave, from one packet to the next.

- PDV is caused by a combination of distance between the master and slave, and the queue delay of the network elements affected by traffic.
- Clock recovery with 1588v2 depends not only on the total amount of PDV, but on how the PDV changes over time.
- The PDV analysis of the different PTP messages can give indication if the clock recovery operation will be affected. For example, PDV maximum performance limits can be met, but if the PDV changes over time are excessive, the clock recovery operation may fail.



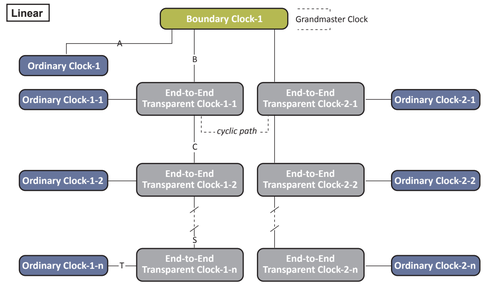
Master-Slave Hierarchy

Hierarchical

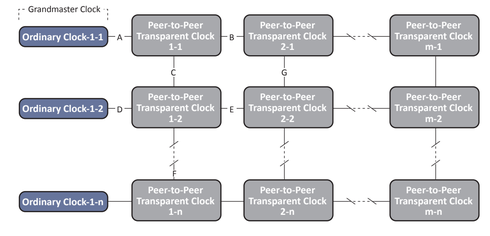


IEEE 1588v2 uses **Precision Clock Synchronization Protocol** to synchronize clocks

Linear



Multiply Connected

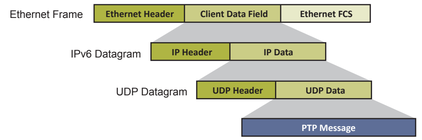


PTP Device Types

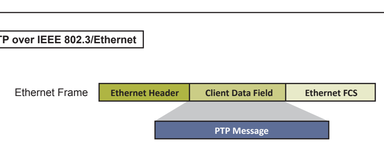
Ordinary Clock	A single port device that can be a Master or Slave clock
Boundary Clock	A multi-port device that can be a Master or Slave clock
End-to-end Transparent Clock	A multi-port device that is not a Master or Slave clock but a bridge between the two. Forwards and corrects all PTP Messages. Correction achieved by addition of the bridge residence time into a correction field within the header of the message.
Peer-to-peer Transparent Clock	A multi-port device that is not a Master or Slave clock but a bridge between the two. Forwards and corrects Sync and Follow_Up message only. Correction achieved by addition of the bridge residence time + the peer-to-peer link delay, into a correction field within the header of the message.
Management Node	A device that configures and monitors clocks

Transport Protocols

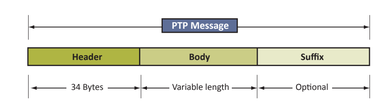
PTP over UDP over IPv4 or IPv6 over Ethernet



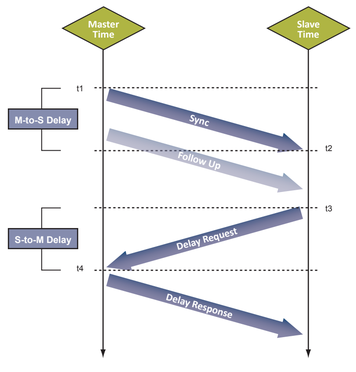
PTP over IEEE 802.3/Ethernet



Basic PTP Message Format



Clock Synchronization



t1 = Time at which the Master sends a Sync message to the slave
t2 = Time at which the Slave receives the Sync message
t3 = Time at which the Slave sends a Delay Response message to the Master
t4 = Time at which the Master receives the Delay Response message from the slave

PTP Messages

Event Messages	General Messages
Sync	Announce
Delay_Req	Follow_Up
Pdelay_Req	Delay_Resp
Pdelay_Resp	Pdelay_Resp_Follow_Up
	Management
	Signalling

PTP Message Header Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
transportSpecific				messageType				1	0
Reserved				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
Reserved								1	5
Flags								2	6
correctionField								8	8
Reserved								4	16
sourcePortIdentity								10	20
sequenceID								2	30
controlField								1	32
logMessageInterval								1	33

Announce Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header								34	0
originTimestamp								10	34
currentUtcOffset								2	44
Reserved								1	46
grandmasterPriority1								1	47
grandmasterClockQuality								4	48
grandmasterPriority2								1	52
grandmasterIdentity								8	53
stepsRemoved								2	61
timeSource								1	63

Sync Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header								34	0
originTimestamp								10	34

Delay_Req Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header								34	0
originTimestamp								10	34

Signalling Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header								34	0
targetPortIdentity								10	34
One or more TLVs								N	44

Pdelay_Resp Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header								34	0
receiveReceiptTimestamp								10	34
requestingPortIdentity								10	44

Management Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header								34	0
targetPortIdentity								10	34
startingBoundaryHops								1	44
boundaryHops								1	45
Reserved				actionField				1	46
Reserved								1	47
managementTLV								M	48

Pdelay_Resp_Follow_Up Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header								34	0
responseOriginTimestamp								10	34
requestingPortIdentity								10	44

Follow_Up Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header								34	0
preciseOriginTimestamp								10	34

Delay_Resp Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header								34	0
receiveTimestamp								10	34
requestingPortIdentity								10	44

Pdelay_Req Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header								34	0
originTimestamp								10	34
reserved								10	44

Management TLV Field Format

Bits								Octets	TLV Offset
7	6	5	4	3	2	1	0		
tlvType								2	0
lengthField								2	2
managementID								2	4
dataField								N	6

Message Exchange Mechanism

A) The master sends a Sync message to the slave and notes the time t1 at which it was sent.

1) Embedding the timestamp t1 in the Sync message, in the case of a one-step clock.
2) Embedding the timestamp t1 in a Follow_Up message, in the case of a two-step clock.

B) The slave receives the Sync message and notes the time of reception t2.
C) The slave sends a Delay_Req message to the master and notes the time t3 at which it was sent.
D) The master receives the Delay_Req message and notes the time of reception t4.
E) The master conveys to the slave the timestamp t4 by embedding it in a Delay_Resp message.

At the conclusion of this exchange of messages, the slave possesses all four timestamps. These timestamps may be used to compute the offset of the slave’s clock with respect to the master and the mean propagation time of messages between the two clocks.

VeEX
www.veexinc.com