IEEE-1588 Standard Version 2

-A Tutorial-

John Eidson
October 2, 2006
john_eidson@agilent.com

Agilent Technologies

Outline

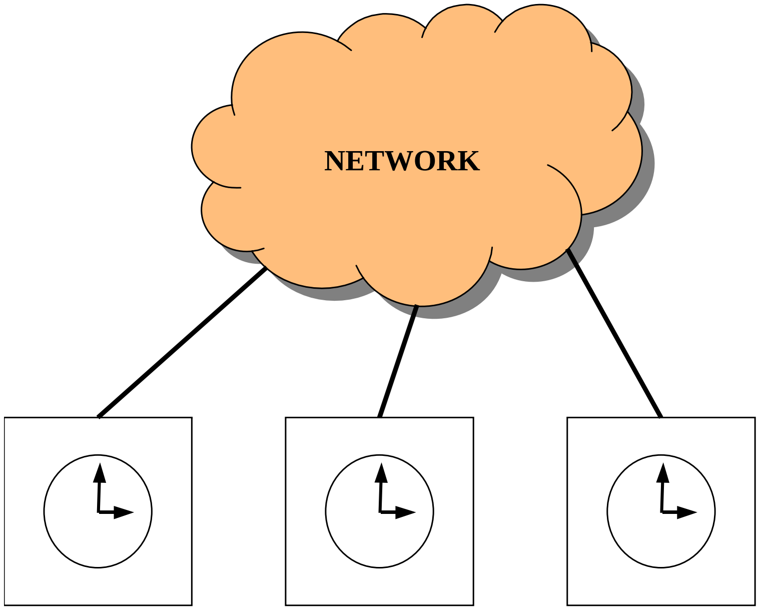
1. **What is IEEE 1588**
2. **Motivation for version 2**
3. **Version 1 Clocks**
4. **New Types of Clocks in Version 2**
5. **Comparison of clock types**
6. **Message changes in version 2**
7. **Other changes**
8. **Status**

What is IEEE 1588

- **IEEE 1588 synchronizes real-time clocks in the nodes of a distributed networked system.**
- **Enables a new methodology for measurement and control**

BASED ON TIME

NOT ON TIME-OF-RECEIPT-BASED EVENT NOTIFICATION.



Major motivation for version 2

•New application areas with new requirements: Telecom, IEEE 802.1as, power industry

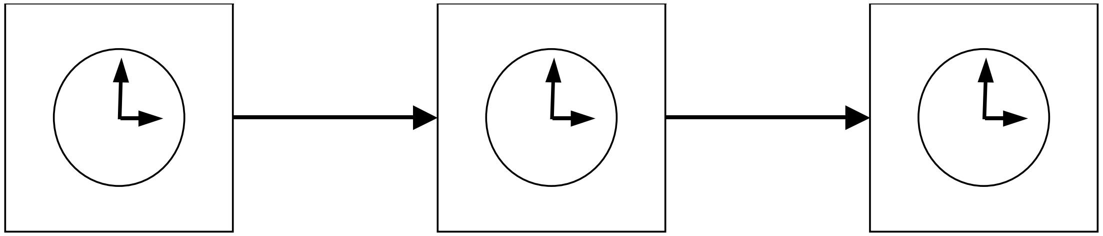
•New requirements in original applications (IA, T&M, military)

- Higher accuracy
- Varied update rates
- Linear topology (in addition to hierarchical)
- Rapid reconfiguration after network changes
- Fault tolerance

V1 Synchronization Basics

Step 1: Organize the clocks into a master-slave hierarchy (based on observing the clock property information contained in multicast Sync messages)

Step 2: Each slave synchronizes to its master (based on Sync, Delay_Req, Follow_Up, and Delay_Resp messages exchanged between master and its slave)

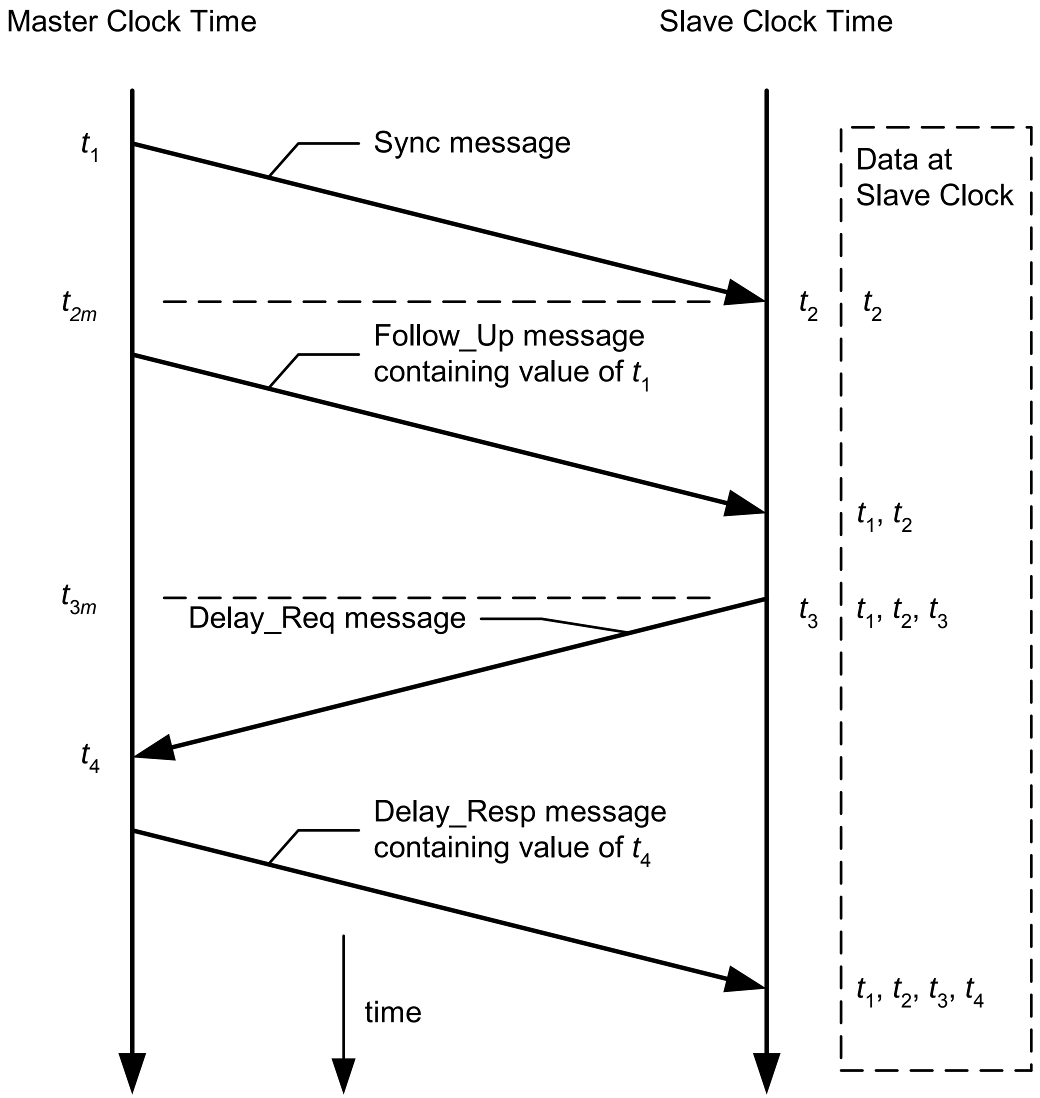


Grandmaster Clock This clock determines the time base for the system

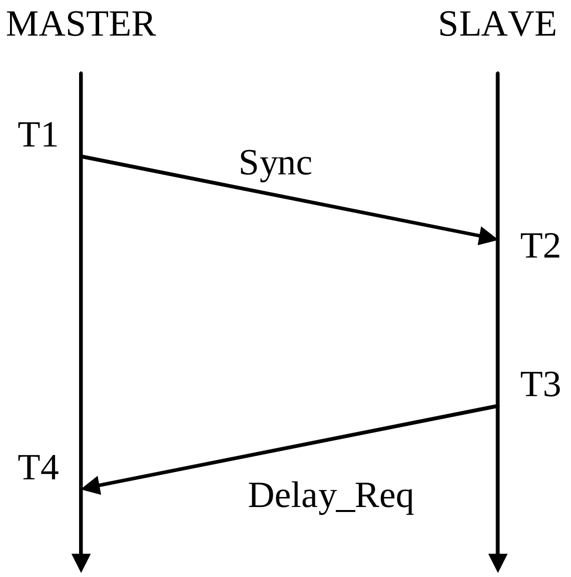
Slave to the Grandmaster Clock and Master to its Slave

Slave to its Master

Timing diagram



IEEE 1588



MASTER

SLAVE

T1

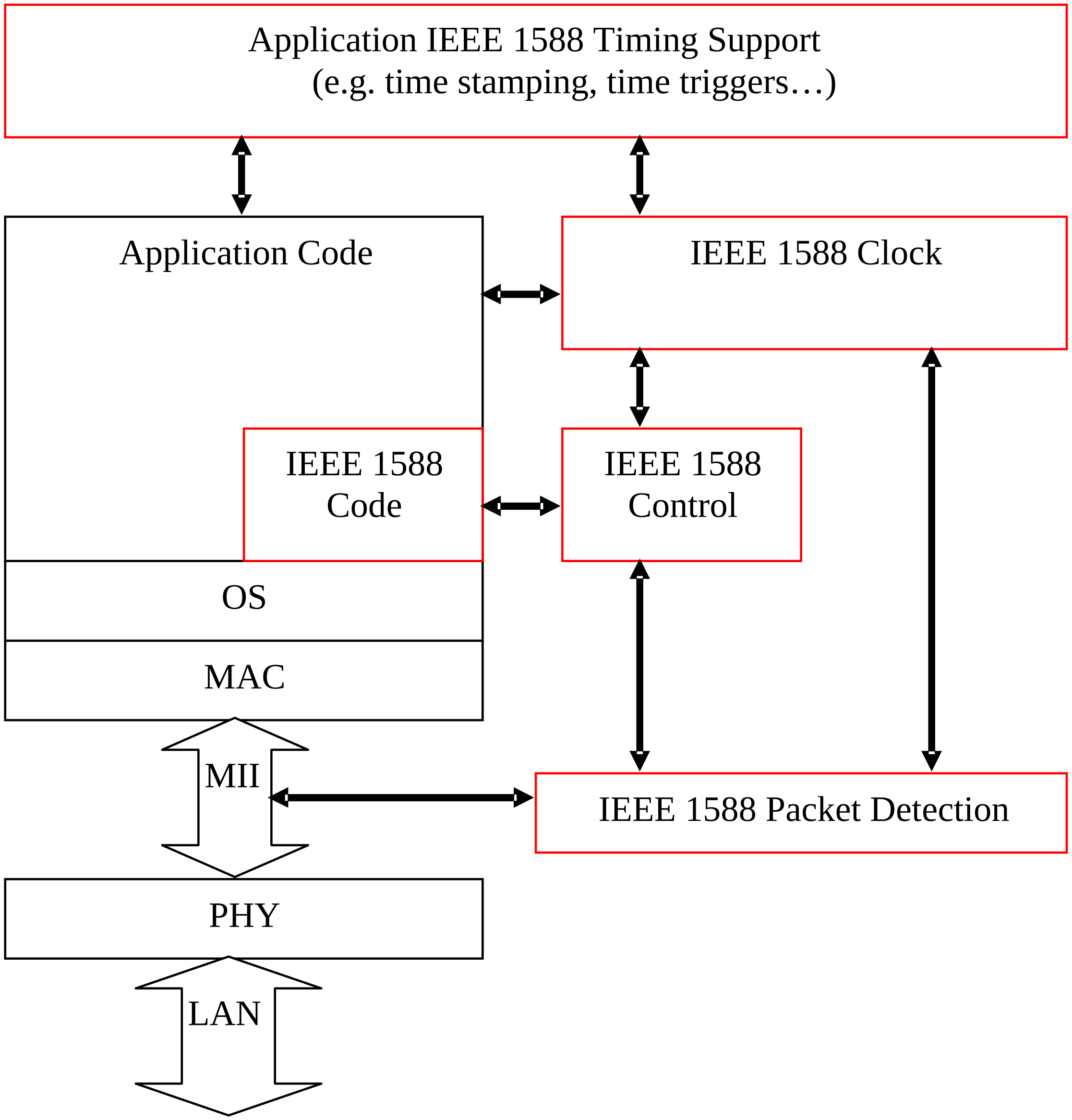
Sync

T2

T3

T4

Delay_Req



Application IEEE 1588 Timing Support
(e.g. time stamping, time triggers…)

Application Code

IEEE 1588 Clock

IEEE 1588
Code

IEEE 1588
Control

OS

MAC

MII

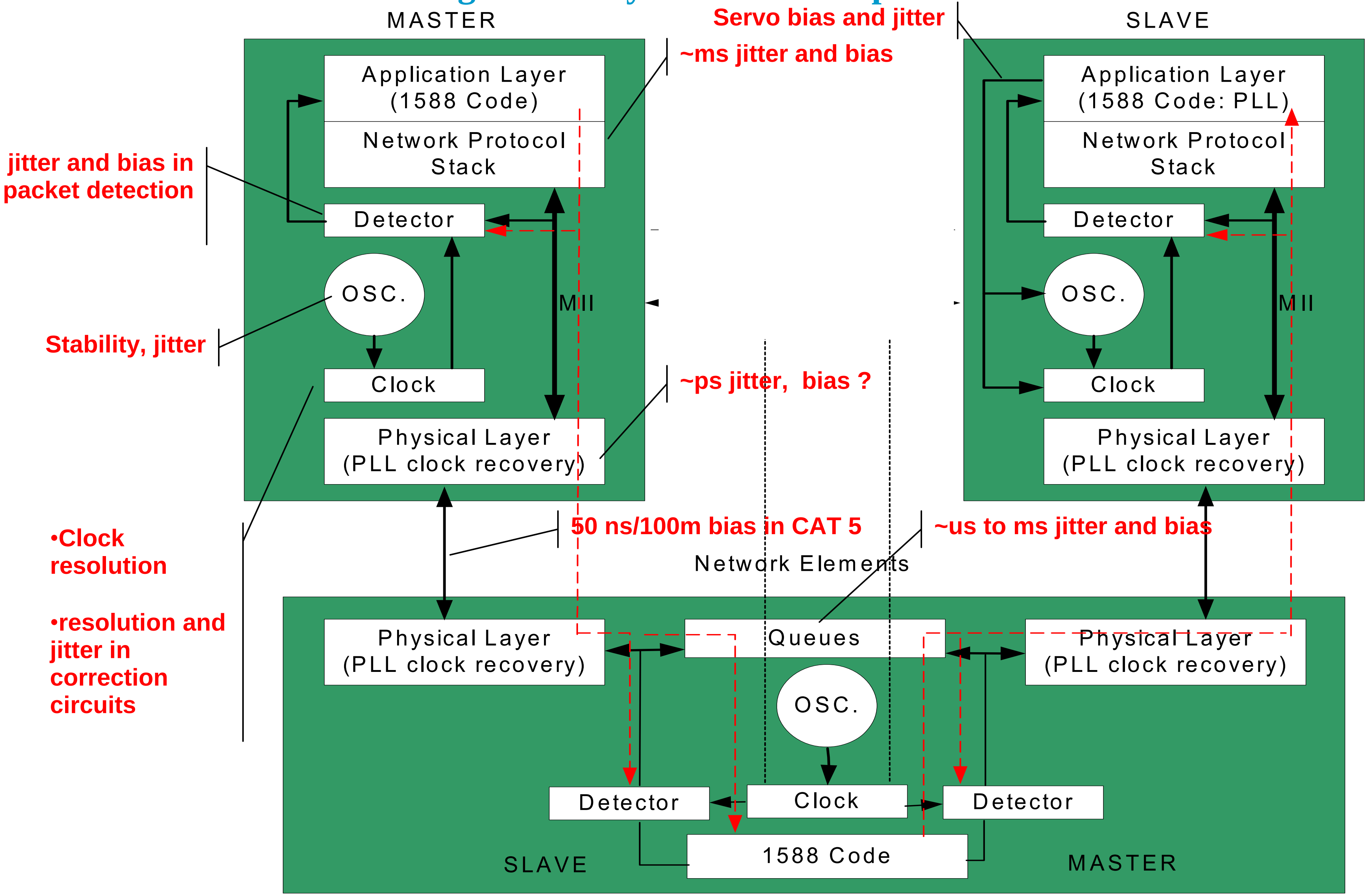
IEEE 1588 Packet Detection

PHY

LAN

Background

High accuracy IEEE 1588 implementation



MASTER

Servo bias and jitter

SLAVE

~ms jitter and bias

Application Layer (1588 Code)

Network Protocol Stack

Application Layer (1588 Code: PLL)

Network Protocol Stack

jitter and bias in packet detection

Detector

Detector

OSC.

OSC.

MII

MII

Stability, jitter

Clock

Clock

~ps jitter, bias ?

Physical Layer (PLL clock recovery)

Physical Layer (PLL clock recovery)

- Clock resolution
- resolution and jitter in correction circuits

50 ns/100m bias in CAT 5

~us to ms jitter and bias

Network Elements

Physical Layer (PLL clock recovery)

Queues

Physical Layer (PLL clock recovery)

OSC.

Detector

Clock

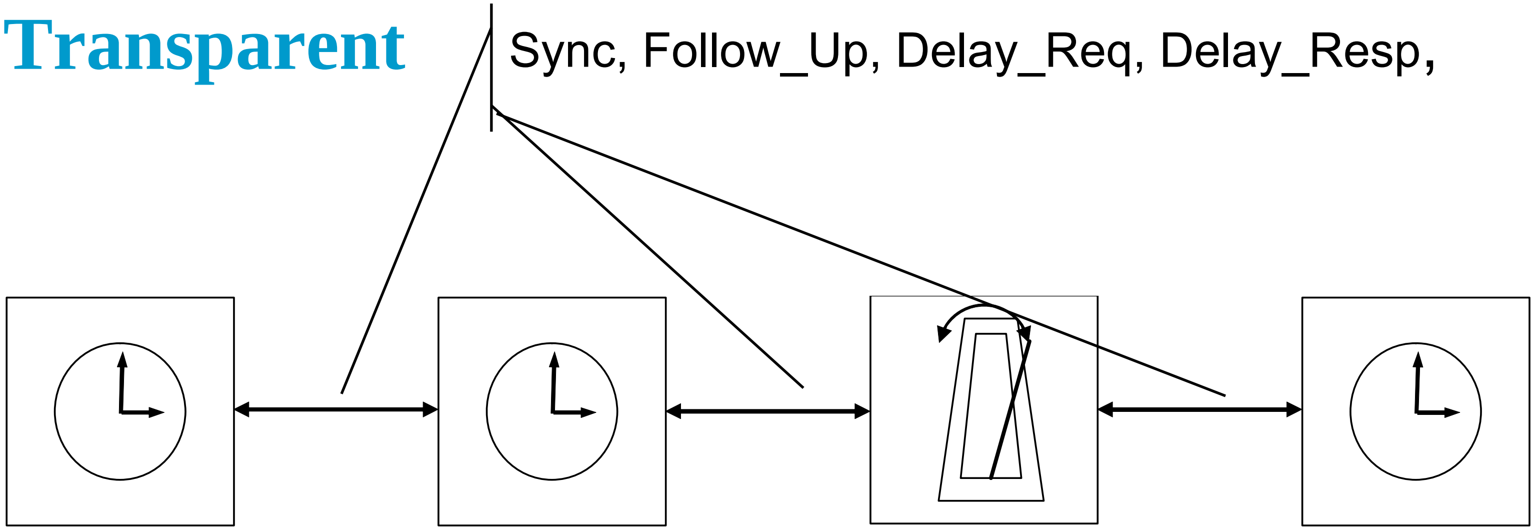
Detector

1588 Code

SLAVE

MASTER

V2 Synchronization Basics + End-to-end Transparent



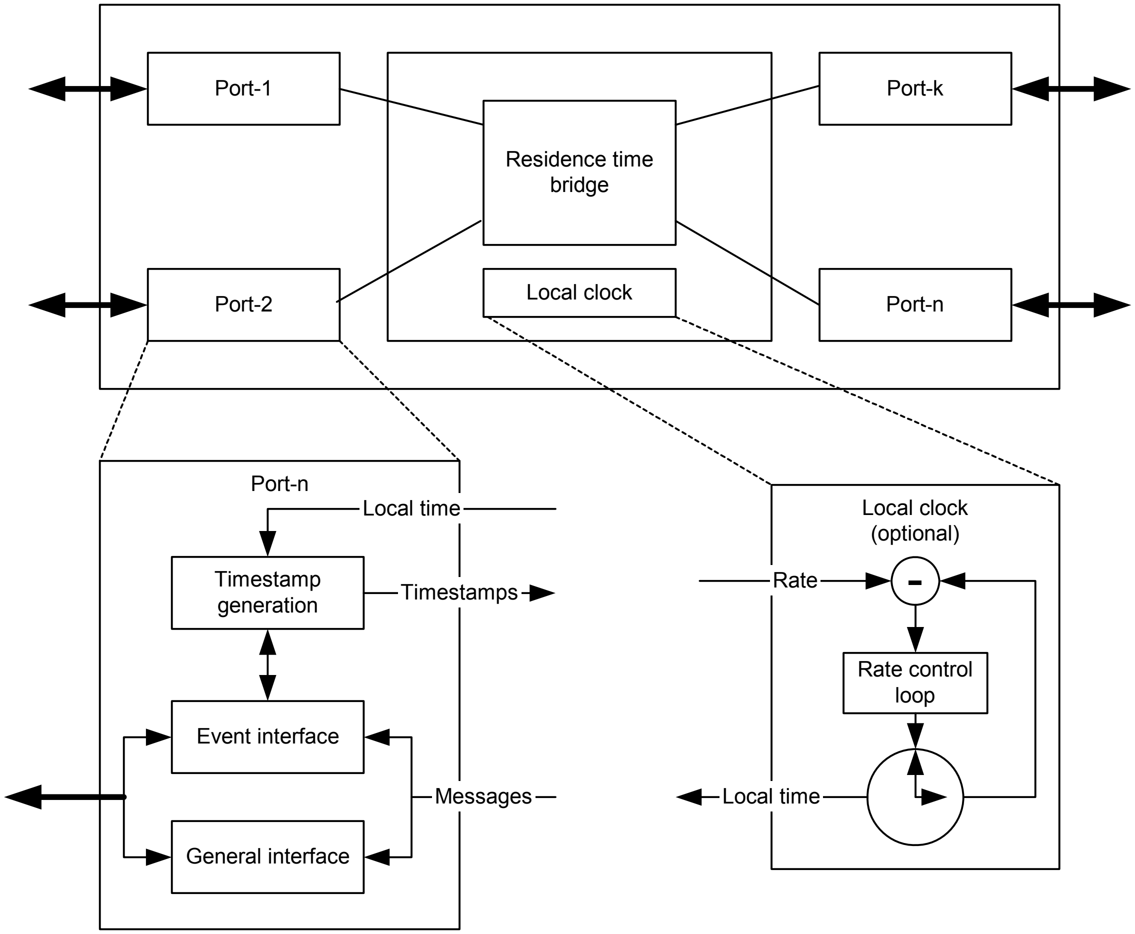
Grandmaster Clock
This clock
determines the time
base for the system

Slave to the
Grandmaster Clock
and Master to its
Slave

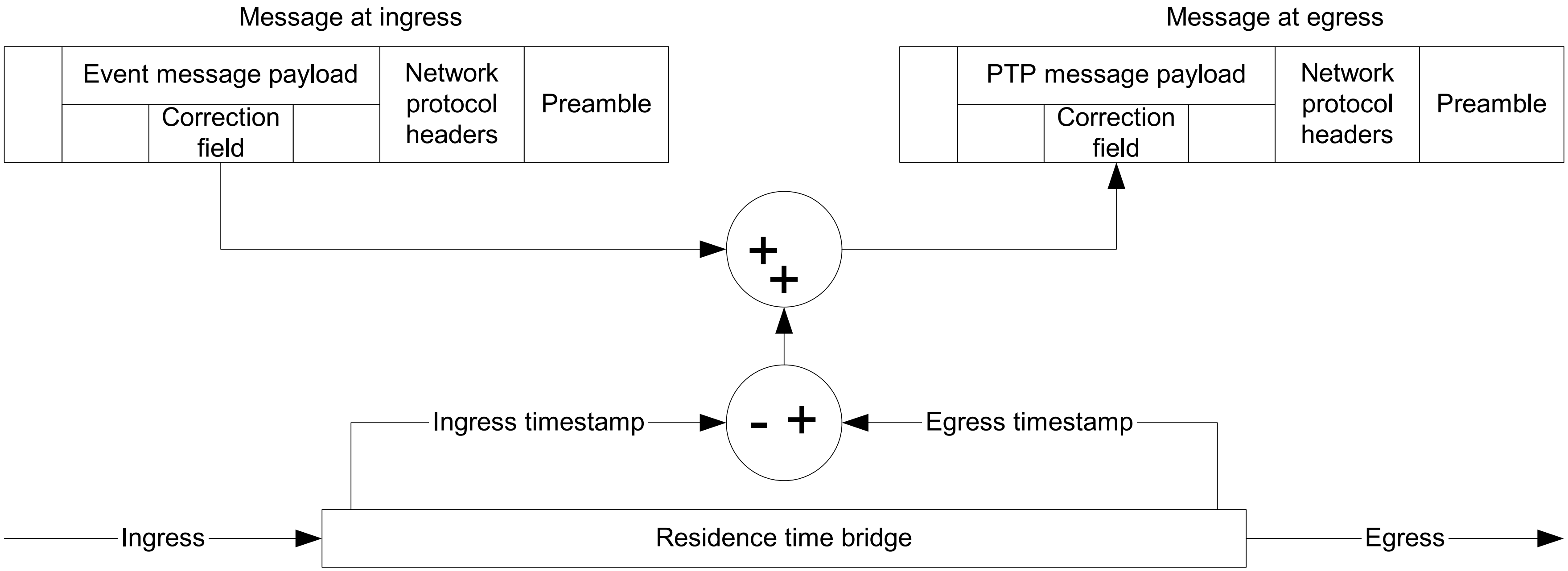
End-to-end
Transparent Clock
(syntonizes to
master)

Slave to its Master

V2 End-to-end Transparent Clock

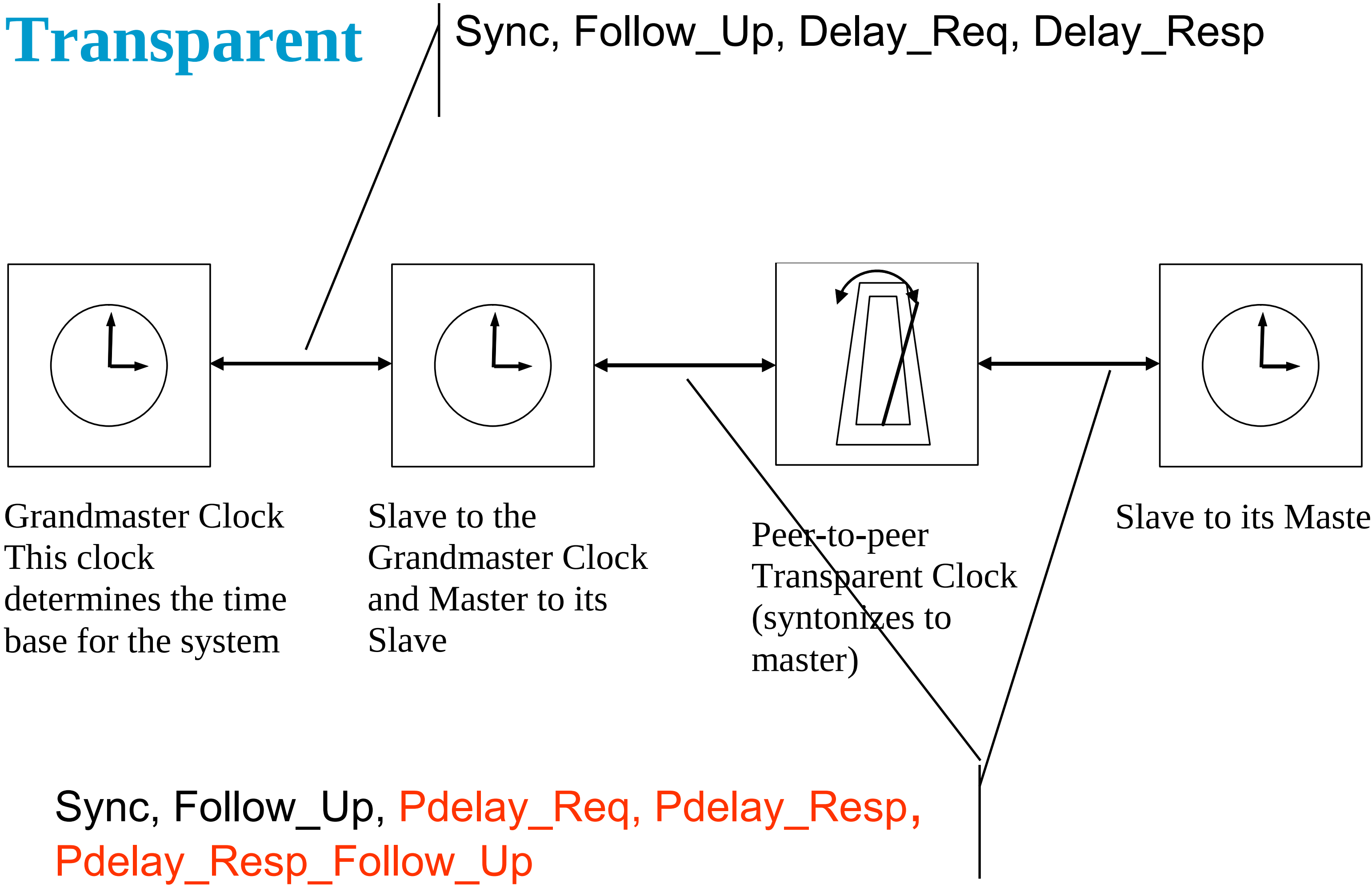


V2 End-to-end Transparent Clock Corrections



Agilent Technologies

V2 Synchronization Basics + Peer-to-peer Transparent



Sync, Follow_Up, Delay_Req, Delay_Resp

Grandmaster Clock
This clock
determines the time
base for the system

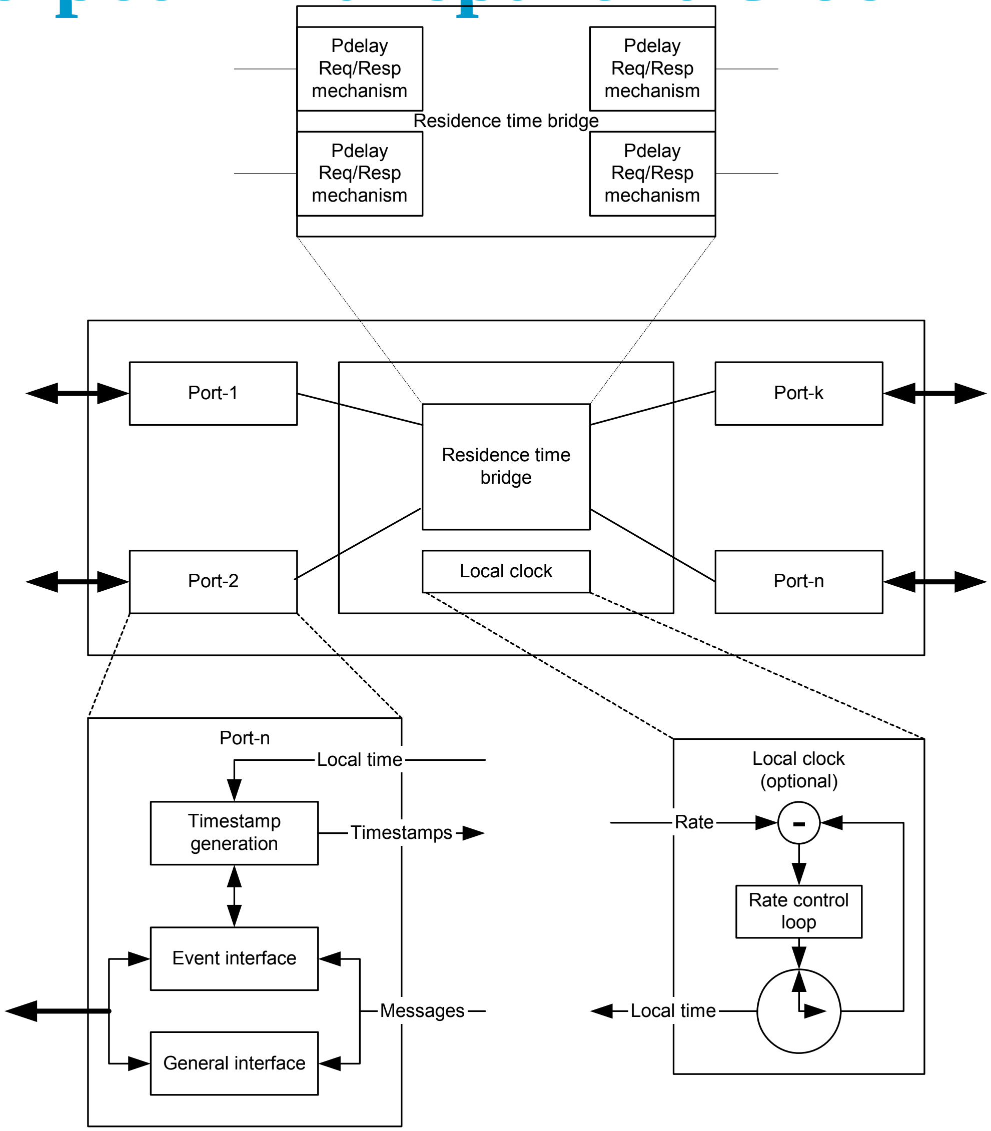
Slave to the
Grandmaster Clock
and Master to its
Slave

Peer-to-peer
Transparent Clock
(syntonizes to
master)

Slave to its Master

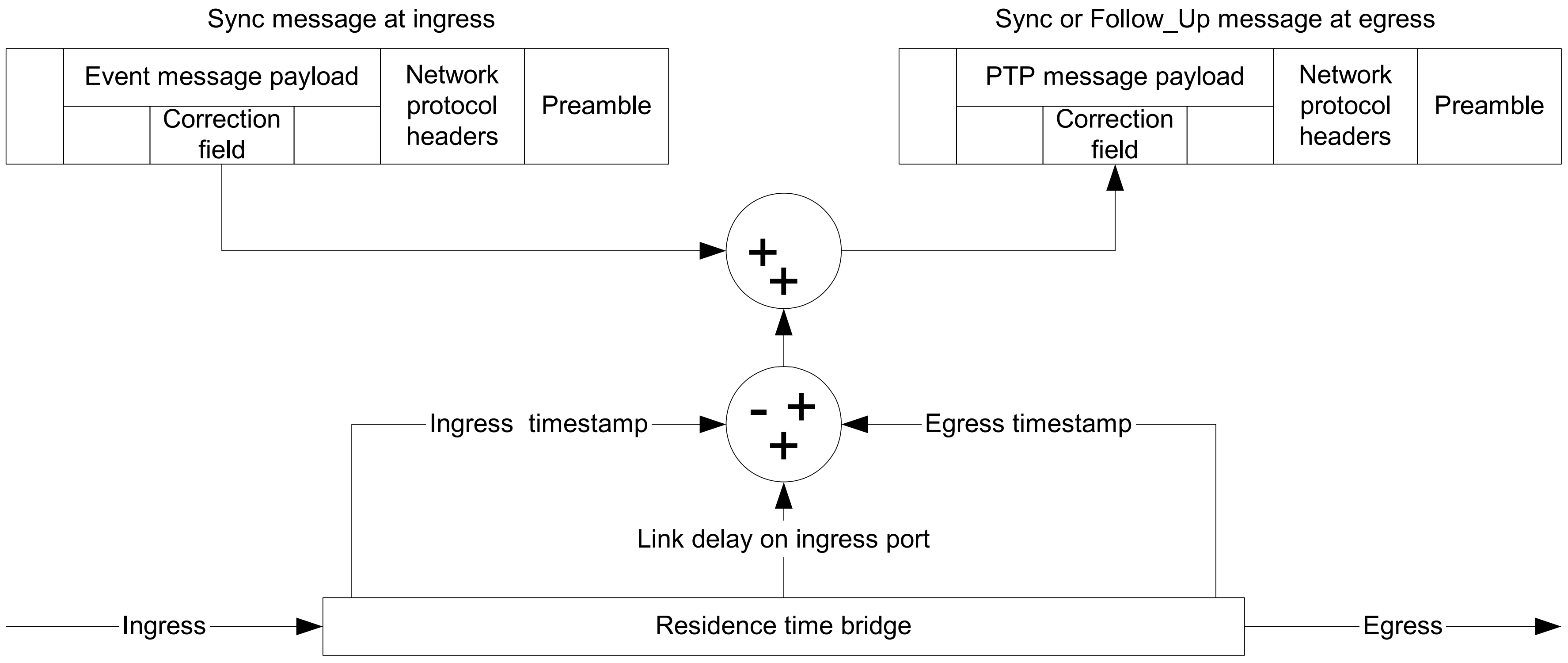
Sync, Follow_Up, Pdelay_Req, Pdelay_Resp,
Pdelay_Resp_Follow_Up

V2 Peer-to-peer Transparent Clock



Agilent Technologies

V2 Peer-to-peer Transparent Clock Corrections



V2 1588 Switches

1. Boundary clocks
   a) + Good for hierarchical systems
   b) + Scale well with the number of devices
   c) – Poor for linear systems (large number of daisy chained clocks)
   d) + Can resolve 1:N introduced by ordinary switches or end-to-end TC
   e) – Must maintain state for scaling and 1:N
   f) + Can translate between different media.
2. End-to-end transparent clocks
   a) + Can be used for hierarchical systems
   b) – Scale poorly with the number of devices (master sees all slaves)
   c) + Good for linear systems (eliminates cascaded servos)
   d) – Can introduce 1:N topology

V2 1588 Switches (con’t)

3. **Peer-to-peer transparent clocks**

   a) **+ Can be used for hierarchical systems**

   b) **+ Scale well with the number of devices**

   c) **+ Good for linear systems (large number of daisy chained clocks)**

   d) **+ Rapid recovery with changes in network topology**

   e) **— Cannot resolve 1:N introduced by ordinary switches, wireless, or end-to-end TC**

   f) **— Must maintain per port path length state and measuring mechanisms.**

   g) **— Only used in homogeneous P2P systems. Requires a boundary clock at the edges.**

   h) **— 6 (vs. 4) measurements per link introduces slight degradation in accuracy**

Split 'timing' and 'master-slave hierarchy determination'

V1 Sync message (165 octets)

- **Network headers**
- **Version, subdomain, type**
- **Source identification**
- **Control, flags, update rates**
- **Origin timestamp**
- **Grandmaster information**
- **Local clock information**
- **Parent clock information**

V2 Sync message (46 octets)

- **Network headers**
- **PTP common: type, version, domain, CORRECTION FIELD, source identification, update rates, *control***
- **Origin timestamp**

Split ‘timing’ and ‘master-slave hierarchy determination’

V1 Sync message (165 octets)

- Network headers
- Version, subdomain, type
- Source identification
- Control, flags, update rates
- Origin timestamp
- Grandmaster information
- Local clock information
- Parent clock information

V2 Announce message (88 octets)

- Network headers
- PTP common: type, version, domain, CORRECTION FIELD, source identification, update rates, *control*
- *Origin timestamp*
- Grandmaster information
- Local clock information
- Parent clock information

Provision for high accuracy implementations & transparent clocks

Correction field:

•Integer64

•Correction in nanoseconds x 2^{+16}

•e.g. 2.5 ns = 0x0000000000028000

Other v2 Changes

•Wider choice of 'Sync' (timing) and 'Announce' (M-S hierarchy) update rates

•Fixed v1 errors (and hopefully did not introduce too many new ones)

•TLV extension mechanism

•Profiles to accommodate different market requirements

•WIP: Fault tolerance, Unicast option, Security, Alternate timescales, SNMP MIB

•WIP: mappings to-L2, Profinet, DeviceNet

Status

- Original ballot target fall 2006
- Probable ballot spring 2007