

IEEE-1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems

-A Tutorial-

John Eidson

October 10, 2005

john_eidson@agilent.com



Agilent Technologies

© Copyright 2005 Agilent Technologies, Inc

Outline

- 1. General overview of the technology and applications**
- 2. Guide to the standard- a detailed analysis of the major clauses**
- 3. IEEE 1588 interoperability/conformance topics**
- 4. Implementation topics**

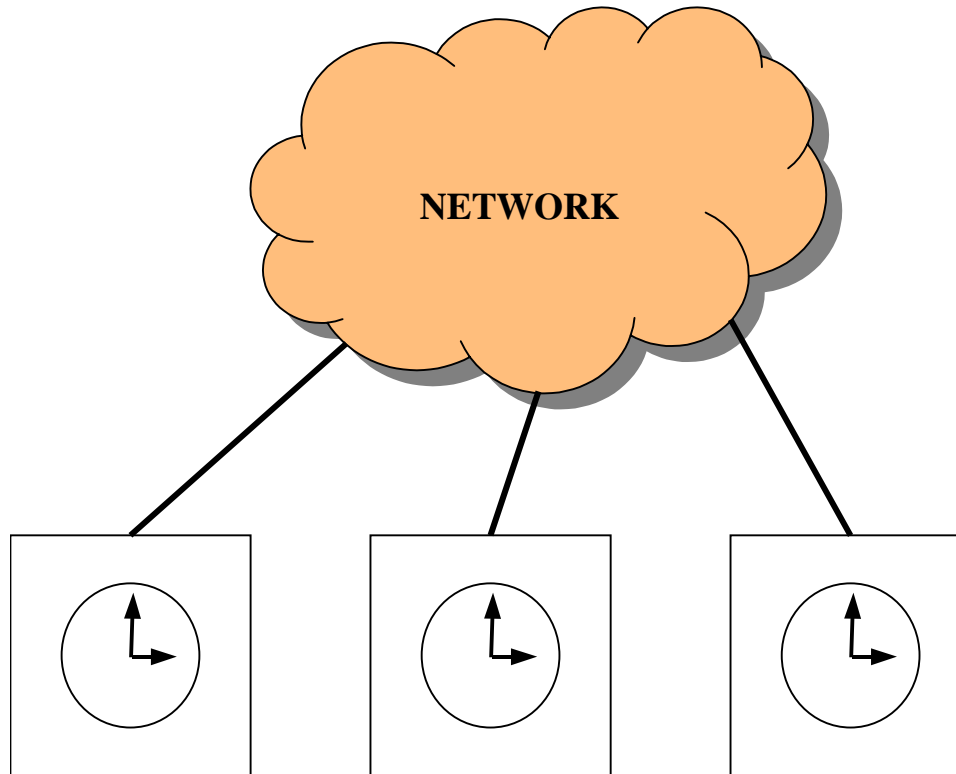


General Overview of the Technology

- a. Purpose**
- b. Status and activities surrounding IEEE 1588**
- c. Comparison to other protocols**

The Purpose of IEEE 1588

IEEE 1588 is a protocol designed to synchronize real-time clocks in the nodes of a distributed system that communicate using a network.



The Status of IEEE 1588

- **Approved by the IEEE-SA Review Committee on September 12, 2002**
- **Published as IEEE 1588-2002 on November 8, 2002**
- **Available from the IEEE <http://standards.ieee.org>**
- **Approved as IEC standard IEC 61588 on May 21, 2004**
- **Products and installations started appearing in late 2003**
- **Conferences on IEEE 1588 held in 2003, 2004, 2005**
- **P1588 committee in process of extending the standard-target completion in late 2006**
- **Current information may be found at <http://ieee1588.nist.gov>**

Comparison to Other Protocols

	IEEE-1588	NTP	GPS	TTP	SERCOS
Spatial extent	A few subnets	Wide area	Wide area	Local bus	Local bus
Communications	Network	Internet	Satellite	Bus or star	Bus
Target accuracy	Sub-microsecond	Few milliseconds	Sub-microsecond	Sub-microsecond	Sub-microsecond
Style	Master/slave	Peer ensemble	Client/server	Distributed	Master/Slave
Resources	Small network message and computation footprint	Moderate network and computation footprint	Moderate computation footprint	Moderate	Moderate



Comparison to Other Protocols (continued)

	IEEE 1588	NTP	GPS	TTP	SERCOS
Latency correction	Yes	Yes	Yes	Configured	No
Protocol specifies security	No (V2 may include security)	Yes	No	No	No
Administration	Self organizing	Configured	N/A	Configured	Configured
Hardware?	For highest accuracy	No	RF receiver and processor	Yes	Yes
Update interval	~2 seconds	Varies, nominally seconds	~1 second	Every TDMA cycle, ~ms	Every TDMA cycle, ~ms



Comparison to Other Protocols (summary)

IEEE 1588: Target is groups of relatively stable components, locally networked (a few subnets), cooperating on a set of well defined tasks.

NTP: (Network Time Protocol, RFC 1305). Target is autonomous systems widely dispersed on the Internet.

GPS: (Satellite based Global Positioning System of the US Department of Defense): Target is autonomous, widely dispersed systems.

TTP(www.ttpforum.org), **SERCOS** (IEC 61491): Target is tightly integrated, usually bus or specialized TDMA network based closed systems.

Guide to the Standard-

(A detailed analysis of the major clauses of version 1)

- a. Overview and goals of the standard**
- b. Synchronization messages and methodology**
- c. Selection of master clocks**
- d. State machine and events**
- e. Timing considerations**
- f. Management messages**

Objectives of IEEE 1588

- **Sub-microsecond synchronization of real-time clocks in components of a networked distributed measurement and control system***
- **Intended for relatively localized systems typical of industrial automation and test and measurement environments. ***
- **Applicable to local area networks supporting multicast communications (including but not limited to Ethernet)**

***indicates objectives that may be extended in version 2**

Objectives of IEEE 1588 (continued)

- **Simple, administration free installation**
- **Support heterogeneous systems of clocks with varying precision, resolution and stability**
- **Minimal resource requirements on networks and host components.**

The IEEE 1588 Standard Defines:

- **Descriptors characterizing a clock**
- **The states of a clock and the allowed state transitions**
- **IEEE 1588 network messages, fields, and semantics**
- **Datasets maintained by each clock**
- **Actions and timing for all IEEE 1588 network and internal events**

- **Critical physical specifications**
- **A suite of messages for monitoring the system**
- **Specifications for an Ethernet based implementation**
- **Conformance requirements**
- **Implementation suggestions**

Overview of the IEEE 1588 Standard

Clause	Purpose	Annex	Purpose
1	Scope	A	User Information
2	Standards References	B	Time Scales
3	Definitions	C	Subdomain Maps
4	Notation Convention	D	Ethernet UDP/IP Implementation
5	Datatypes	E	Bibliography
6	Protocol Overview		
7	Protocol		
8	Message Specifications		
9	Conformance		

WARNING

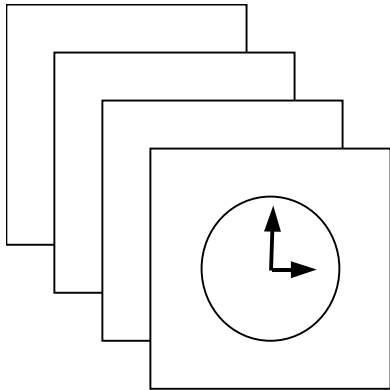
The IEEE has rather strict rules on interpreting IEEE Standards. No individual or organization can issue official interpretations or provide definitive answers to questions of interpretation. This must be done by an IEEE authorized committee. Even this committee cannot extend, correct, or change the standard- this must be done by ballot.

However-We can learn from and share our collective experience.



Clause 6: PTP Clock Synchronization Model

QUESTION: How do we take a collection of clocks, message types, clock properties, networks, etc. and produce a consistent time base in all the participating clocks?



MESSAGE TYPES	CLOCK PROPERTIES
Sync	UUID
Delay_Req	Stratum
Follow_Up	Identifier
Delay_Resp	State
Management	Variance

NETWORK COMMUNICATION FORMS:

Ethernet (UDP/IP), DeviceNet, L2 Ethernet, 802.11b, ...

Guide to the Standard-

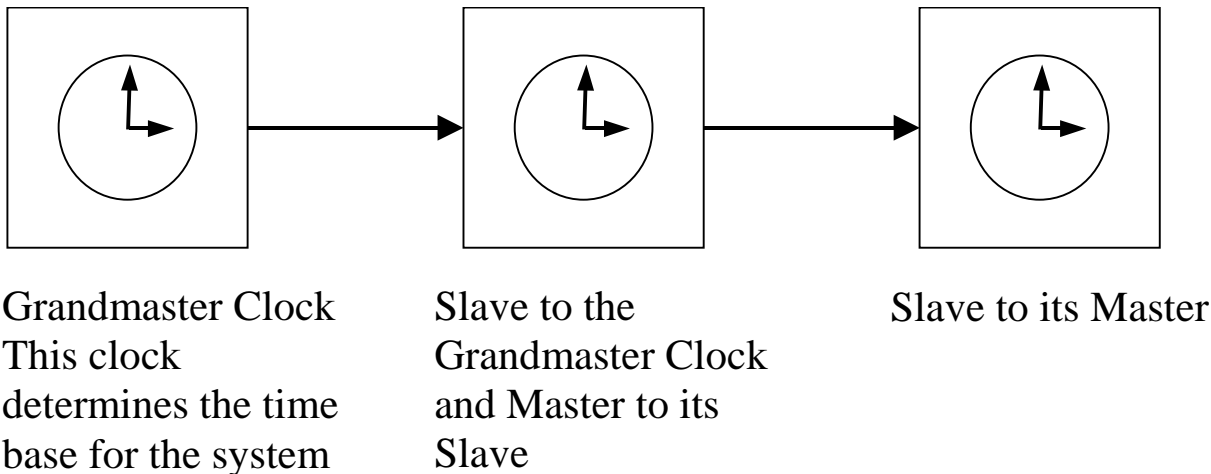
(A detailed analysis of the major clauses)

- a. Overview and goals of the standard
- b. Synchronization messages and methodology**
- c. Selection of master clocks
- d. State machine and events
- e. Timing considerations
- f. Management messages
- g. Time scales
- h. Annex D

Clause 6: IEEE 1588 Synchronization Basics

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

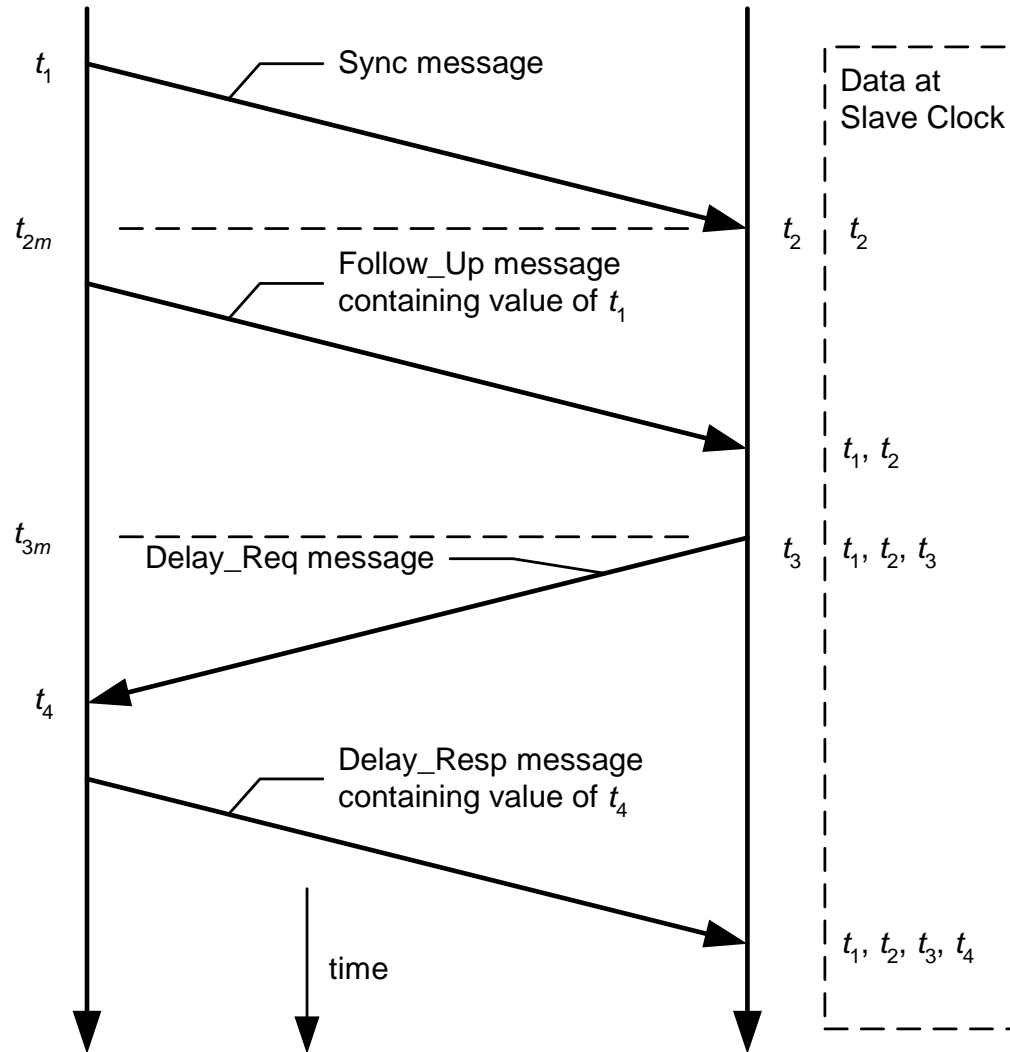
Step 2: Each slave synchronizes to its master (based on Sync, Delay_Req, Follow_Up, and Delay_Resp messages exchanged between master and its slave)



Clause 6: Synchronization Basics (continued)

Master Clock Time

Slave Clock Time



Clause 6: Synchronization Basics (continued)

To synchronize a pair of clocks, First:

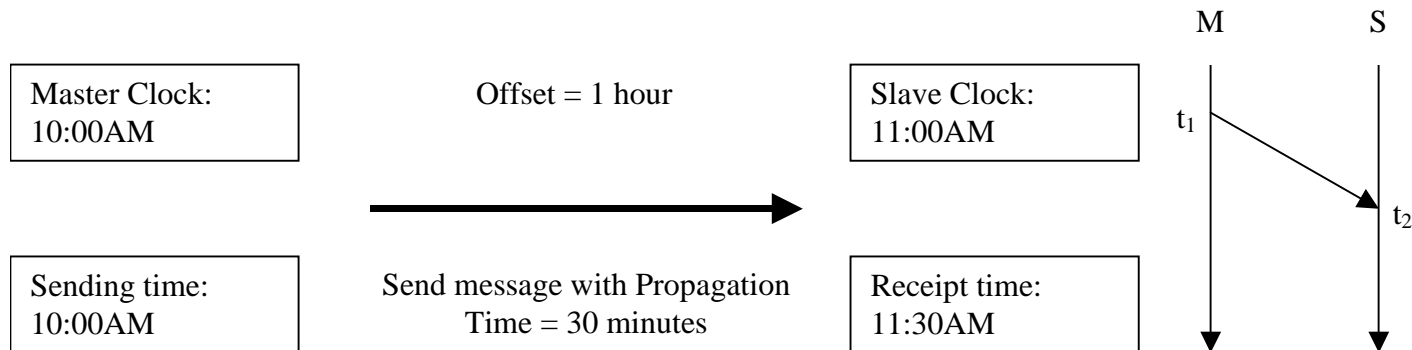
- Send a message, (Sync message), from master to slave and measure the apparent time difference between the two clocks.
MS_difference = slave's receipt time – master's sending time

$$= t_2 - t_1$$

- **MS_difference = offset + MS delay (by inspection)**

- **For example:**

MS_difference = slave's receipt time – master's sending time
90 minutes = 11:30 – 10:00



Clause 6: Synchronization Basics (continued)

Second:

- Send a message, (Delay_Req message), from slave to master and measure the apparent time difference between the two clocks.

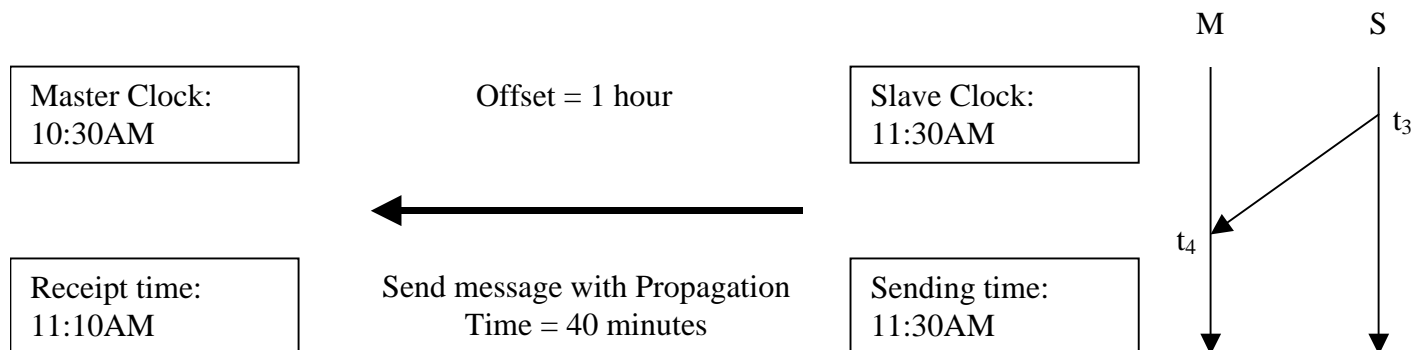
SM_difference = master's receipt time – slave's sending time

$$= t_4 - t_3$$

SM_difference = – offset + SM delay (by inspection)

- For example:

SM_difference = master's receipt time – slave's sending time – 20 minutes = 11:10 – 11:30



Clause 6: Synchronization Basics (continued)

The result is that we have the following two equations:

$$\text{MS_difference} = \text{offset} + \text{MS delay}$$

$$\text{SM_difference} = -\text{offset} + \text{SM delay}$$

With **two** measured quantities:

$$\text{MS_difference} = 90 \text{ minutes}$$

$$\text{SM_difference} = -20 \text{ minutes}$$

And **three** unknowns:

offset , MS delay, and SM delay

Clause 6: Synchronization Basics (continued)

Rearranging the two equations:

$$\text{MS_difference} = \text{offset} + \text{MS delay}$$

$$\text{SM_difference} = -\text{offset} + \text{SM delay}$$

We get:

$$\text{offset} = \{(\text{MS_difference} - \text{SM_difference}) - (\text{MS delay} - \text{SM delay})\}/2$$

$$\text{MS delay} + \text{SM delay} = \{\text{MS_difference} + \text{SM_difference}\}$$

ASSUME: $\text{MS delay} = \text{SM delay} = \text{one_way_delay}$

Then:

$$\text{offset} = \{\text{MS_difference} - \text{SM_difference}\}/2$$

$$\text{one_way_delay} = \{\text{MS_difference} + \text{SM_difference}\}/2$$

Clause 6: Synchronization Basics (continued)

$$\text{offset} = \{\text{MS_difference} - \text{SM_difference}\}/2$$

$$\text{one_way_delay} = \{\text{MS_difference} + \text{SM_difference}\}/2$$

In our example using the two measured quantities:

$$\text{MS_difference} = 90 \text{ minutes}$$

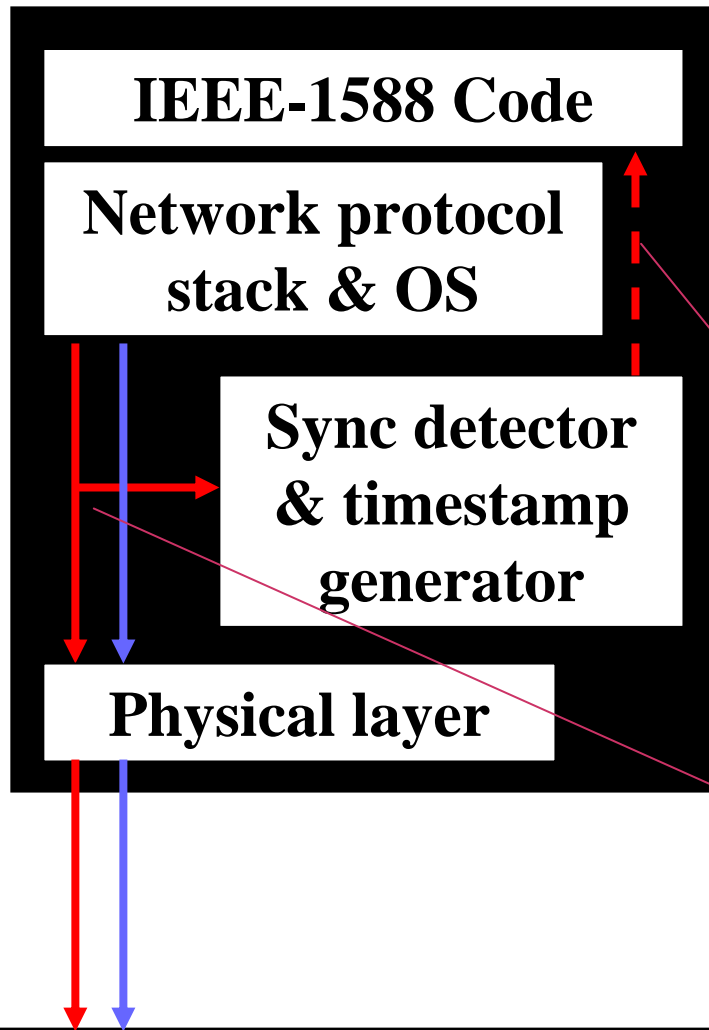
$$\text{SM_difference} = -20 \text{ minutes}$$

We get:

$$\text{offset} = \{90 - (-20)\}/2 = 55 \text{ minutes (not actual 60)}$$

$$\text{one_way_delay} = \{90 + (-20)\}/2 = 35 \text{ minutes (not 30 or 40)}$$

Synchronization Details (clauses 6 & 7)



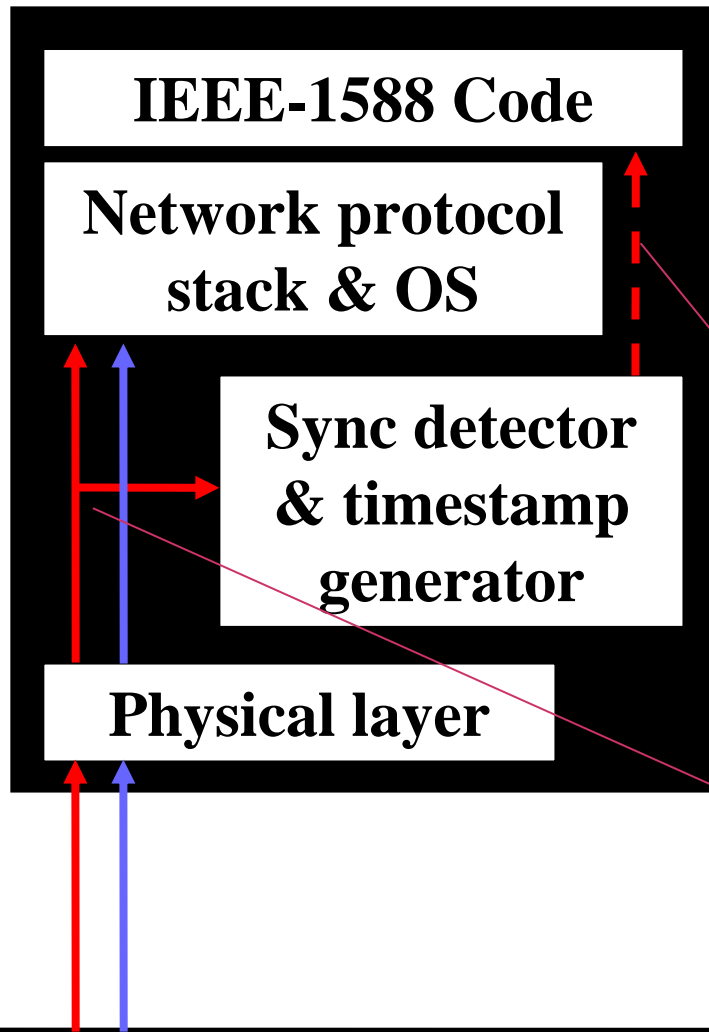
Master clock sends:

1. Sync message
2. Follow_up message

Time at which a Sync message passed the Timestamp Point (t_1)

Timestamp Point

Synchronization Details (continued)



Slave clock receives:

1. Sync message
2. Follow_up message

Time at which a Sync message passed the Timestamp Point (t_2)

Timestamp Point

Synchronization Details (continued)

Sync messages:

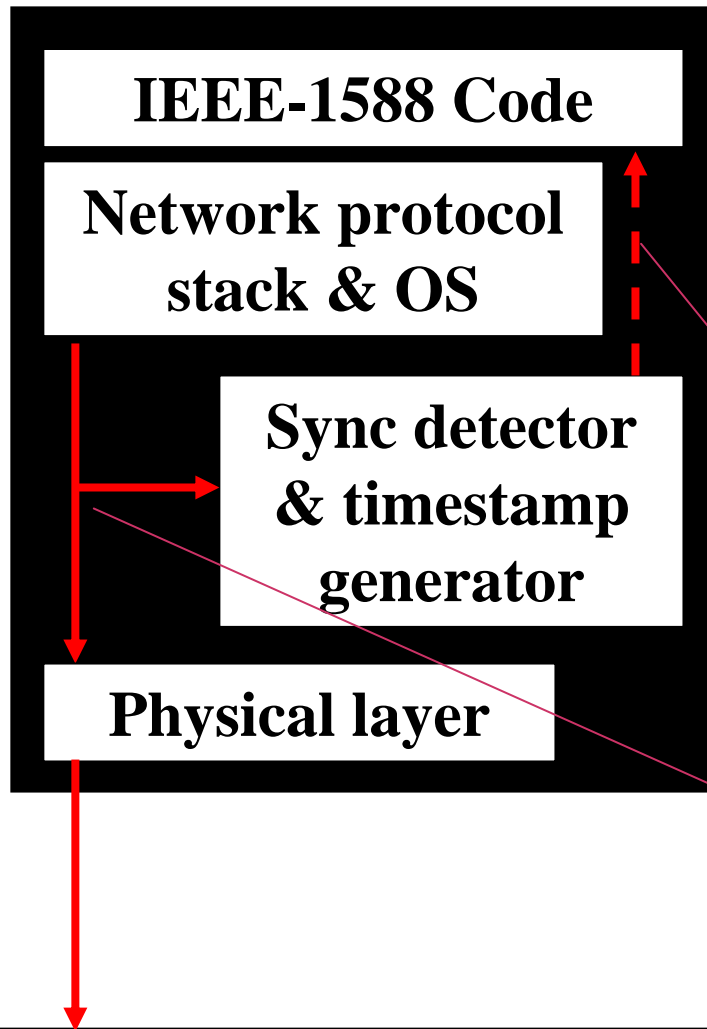
- Issued by clocks in the 'Master' state
- Contain clock characterization information
- Contain an **estimate** of the sending time ($\sim t_1$)
- When received by a slave clock the receipt time is noted
- Can be distinguished from other legal messages on the network
- For best accuracy these messages can be easily identified and detected at or near the physical layer and the precise sending (or receipt) time recorded

Synchronization Details (continued)

Follow_Up messages:

- Issued by clocks in the ‘Master’ state
- Always associated with the preceding Sync message
- Contain the ‘**precise sending time= (t_1)**’ as measured as close as possible to the physical layer of the network
- When received by a slave clock the ‘precise sending time’ is used in computations rather than the estimated sending time contained in the Sync message

Synchronization Details (continued)



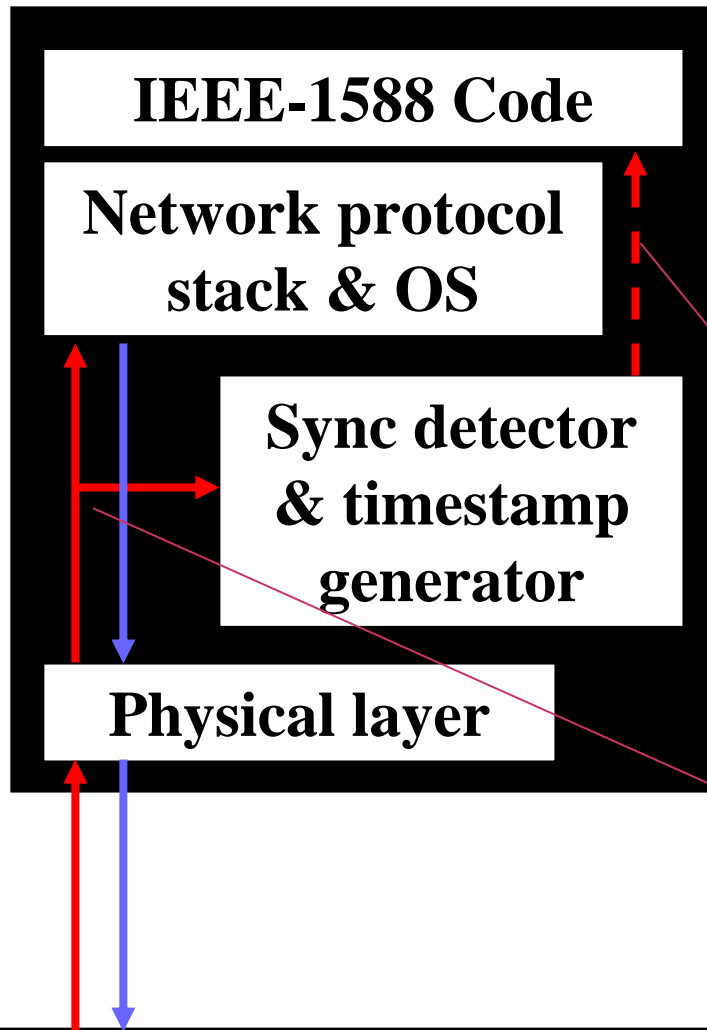
Slave clock sends:

- **Delay_Req message**

Time at which a Delay_Req message passed the Timestamp Point (t_3)

Timestamp Point

Synchronization Details (continued)



Master clock receives:

- **Delay_Req** message

Master clock sends:

- **Delay_Resp** message

Time at which a Delay_Req message passed the Timestamp Point (t_4)

Timestamp Point

Delay_Req messages:

- **Issued by clocks in the ‘Slave’ state**
- **The slave measures and records the sending time (t_3)**
- **When received by the master clock the receipt time is noted (t_4)**
- **Can be distinguished from other legal messages on the network**
- **For best accuracy these messages can be easily identified and detected at or near the physical layer and the precise sending (or receipt) time recorded**

Delay_Resp messages:

- **Issued by clocks in the ‘Master’ state**
- **Always associated with a preceding Delay_Req message from a specific slave clock**
- **Contain the receipt time of the associated Delay_Req message (t_4)**
- **When received by a slave clock the receipt time is noted and used in conjunction with the sending time of the associated Delay_Req message as part of the latency calculation**

Synchronization Details (continued)

Synchronization computation (in the Slave clock):

offset = receipt time – precise sending time – one way delay
(for a Sync message)

one way delay = {master to slave delay + slave to master delay}/2 (**assumes symmetric delay**)

master to slave delay = receipt time – precise sending time
(for a Sync message)

slave to master delay = Delay_Req receipt time - precise sending time (of a Delay_Req message)

From this **offset the slave corrects its local clock!**

From This Offset the Slave Corrects its Local Clock!

BUT: The standard says nothing about how to do this.

(more later)

Guide to the Standard-

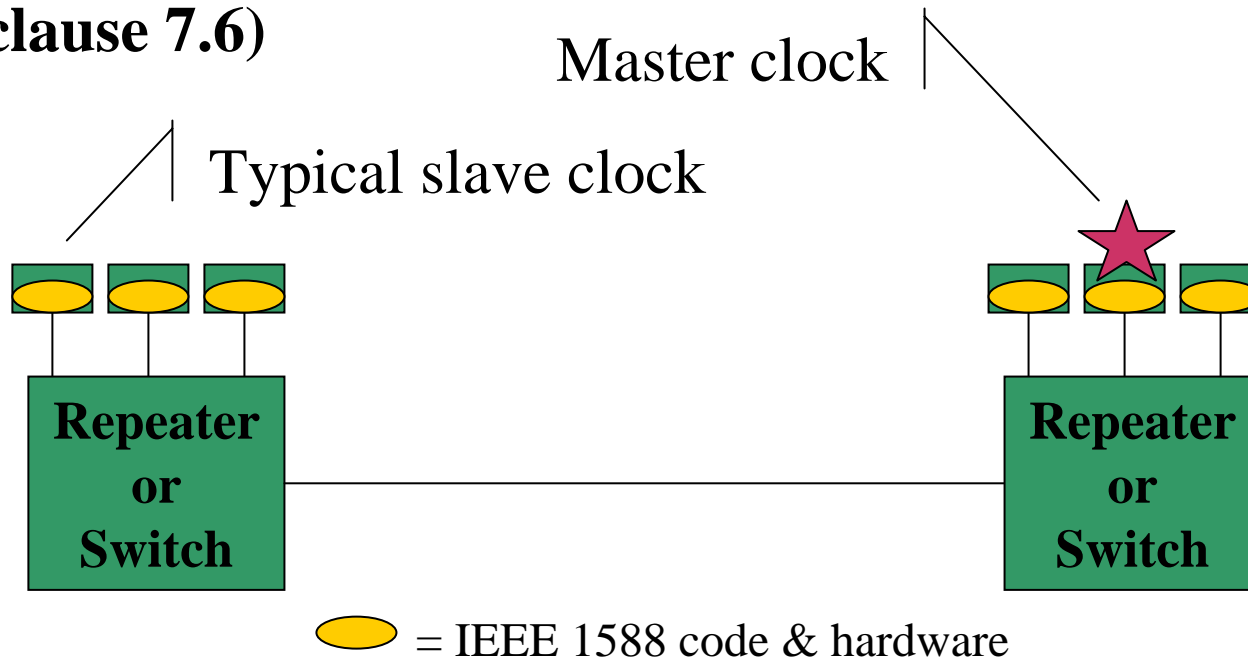
(A detailed analysis of the major clauses)

- a. Overview and goals of the standard
- b. Synchronization messages and methodology
- c. Selection of master clocks**
- d. State machine and events
- e. Timing considerations
- f. Management messages
- g. Time scales
- h. Annex D

Selecting a Master Clock-Single Subnet

Self-configuring based on clock characteristics and network topology

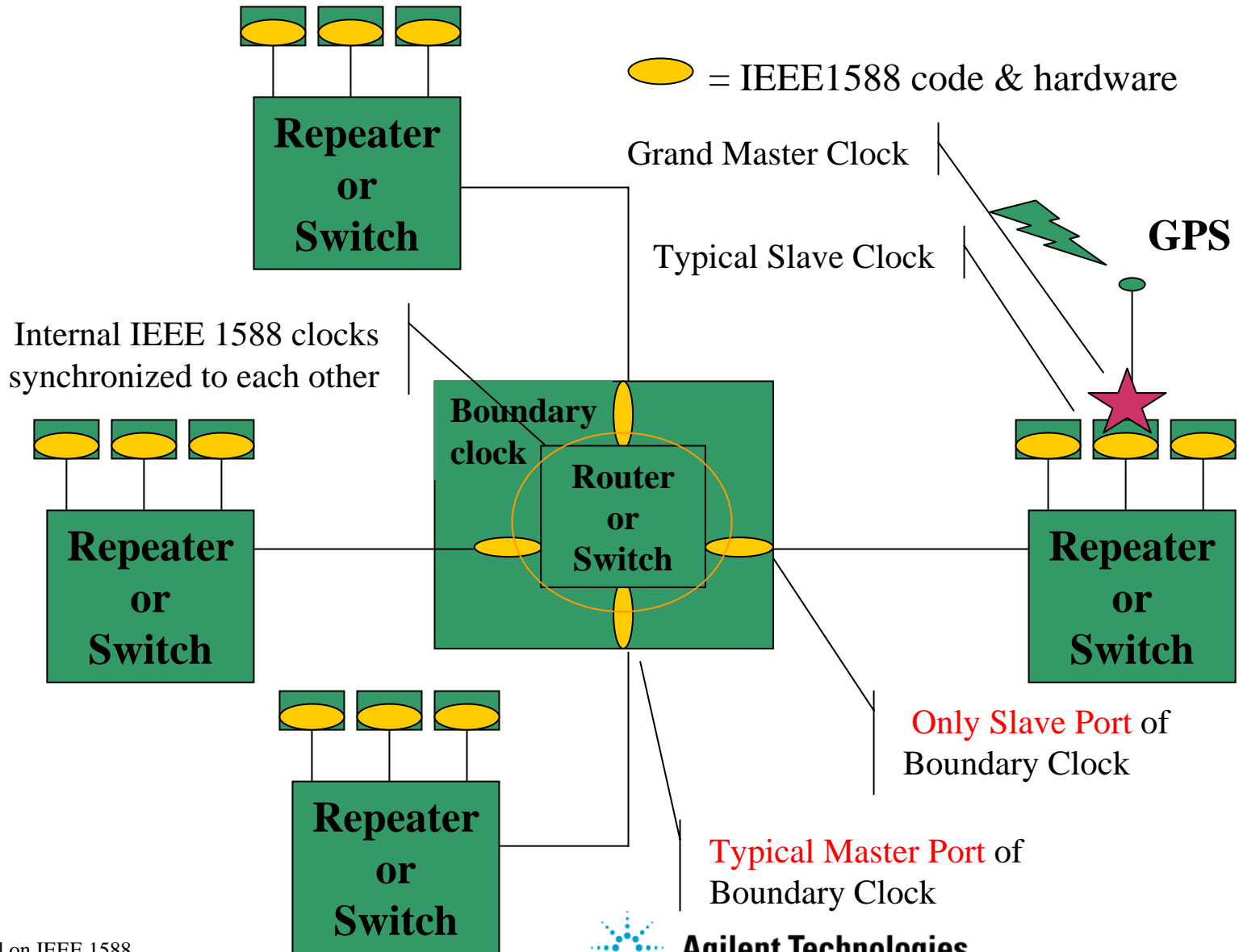
- Based on information contained in ‘Sync’ messages
- All clocks run an identical ‘**Best Master Clock**’ algorithm (clause 7.6)



Selecting a Master Clock-Simplified (clause 7.6.1)

- **A clock at startup listens for a time SYNC_RECEIPT_TIMEOUT**
- **A master clock (clock in the PTP_MASTER state) issues periodic Sync messages (period is called the sync_interval)**
- **A master clock may receive Sync messages from other clocks (who for the moment think they are master) which it calls ‘foreign masters’**
- **Each master clock uses the Best Master Clock algorithm to determine whether it should remain master or yield to a foreign master.**
- **Each non-master clock uses the Best Master Clock algorithm to determine whether it should become a master.**

IEEE 1588 Multiple Subnet Topology



Multiple Subnet Synchronization & Master Clock Selection (more details)

- **Boundary clocks do NOT pass Sync, Follow_Up, Delay_Req, or Delay_Resp messages. Boundary clocks thus segment the network as far as IEEE 1588 synchronization is concerned.**
- **Within a subnet a port of a boundary clock acts just like an ordinary clock with respect to synchronization and best master clock algorithm**
- **The boundary clock internally selects the port that sees the ‘best clock’ as the single slave port. This port is a slave in the selected subnet. All other ports of the boundary clock internally synchronize to this slave port.**



- **Boundary clocks define a parent-child hierarchy of master-slave clocks.**
- **The best clock in the system is the Grand Master clock.**
- **If there are cyclic paths in the network topology the best master clock algorithm reduces the logical topology to an acyclic graph.**

Best Master Clock Algorithm-overview (clause 7.6)

1. A **master** clock 'A' can receive Sync messages from other **potential master** clocks- 'B', 'C',...
2. Clock 'A' decides:
 - a. Which of the clocks 'B', 'C' ,... is the 'best' clock
 - b. Whether clock 'A' is better than the best of 'B', 'C' ,...
3. Using the Best Master Clock algorithm, **BMC**, it does this by **pair wise comparisons of the data sets** describing each of the clocks.
4. Based on the results of this comparison the BMC returns a recommended clock state: in simple situations either **master** or **slave**.
5. All clocks operate on the same information and therefore arrive at consistent results.
6. Data for these comparisons logically is maintained by each clock in one of several **data sets**.



Datasets Maintained by Each Clock (clause 7.4)

The following are PER CLOCK data sets:

- **Default data set:** Properties of the local clock that determine its behavior and performance when it is the grandmaster clock
- **Global time properties data set:** Time base properties
- **Current data set:** Current synchronization and topological operational properties

The following are PER CLOCK PORT data sets:

- **Parent data set:** Properties of the parent and grandmaster
- **Port configuration data set:** Clock port properties
- **Foreign master data set:** Identification of Sync messages from potential master clocks-part of a qualification scheme to reduce thrashing

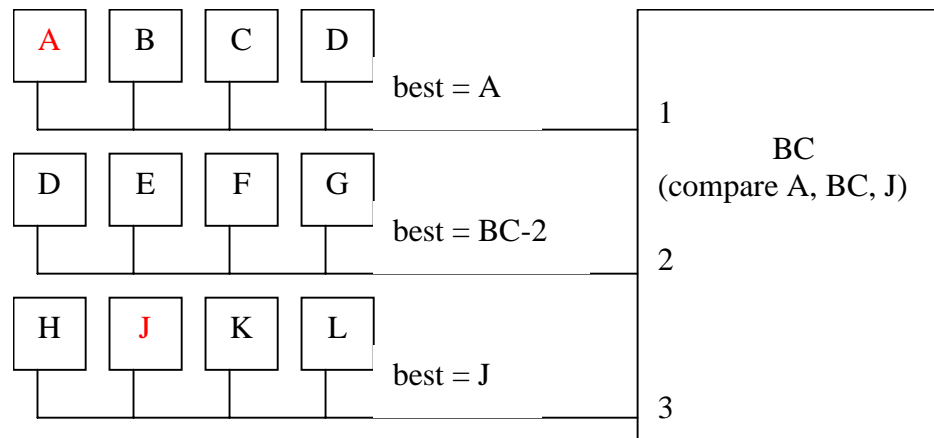
“It does this by pair wise comparisons of the data sets...”

•On a subnet an ordinary clock sees itself and others on the same subnet: default and foreign master data sets:

e.g. ‘A’ sees B,C,D, and BC port 1

•A boundary clock sees itself and all clocks on its several subnets: default and the foreign master set for each port:

e.g. BC sees all ordinary clocks, A,...L, plus itself.



IEEE 1588 Characterization of Clocks (clause 6)

The following are the principal items used by the BMC.

- **Based on primary source of time, e.g. GPS, local oscillator...**
- **Accuracy**
- **Variance**
- **Preferred set membership**
- **Type: Boundary clock (spans subnets) or ordinary clock**
- **UUID**

Best Master Clock Algorithm-details clause 7.6

The BMC algorithm consist of two sub algorithms:

- 1. State decision algorithm:** using the results of comparisons of all pairs of relevant data sets this produces a recommended state.
- 2. Data set comparison algorithm:** a binary relation using specific information from the data sets of the two clock ports being compared:
 - a. Select the clock that derives its time from the better grandmaster
 - b. If the grandmasters are equivalent choose the 'closest' grandmaster
 - c. If the above fail to indicate a choice use tie-breaking (UUID)



Data Set Comparison Algorithm (clause 7.6.4)

Standard has a big flow chart (figures 17, 18, 19 & 20) and a table (20) defining this algorithm. The net effect is to define a hierarchy of choices of which the first one satisfied determines which of the two data sets represents the ‘better’ clock (port). The hierarchy is:

- 1.Preferred: (designates a set from which GM is selected)**
- 2.Stratum: (clause 6.2.4.3- primary or secondary standard)**
- 3.Identifier: (6.2.4.5- accuracy of clock’s time base)**
- 4.Variance: (6.2.4.8- stability and noise of clock)**
- 5.‘Closest’: minimum spanning tree algorithm (key to understanding mechanism is Table 21)**
- 6.UUID (tiebreaker)**

State Decision Algorithm (clause 7.6.3 & figure 16)

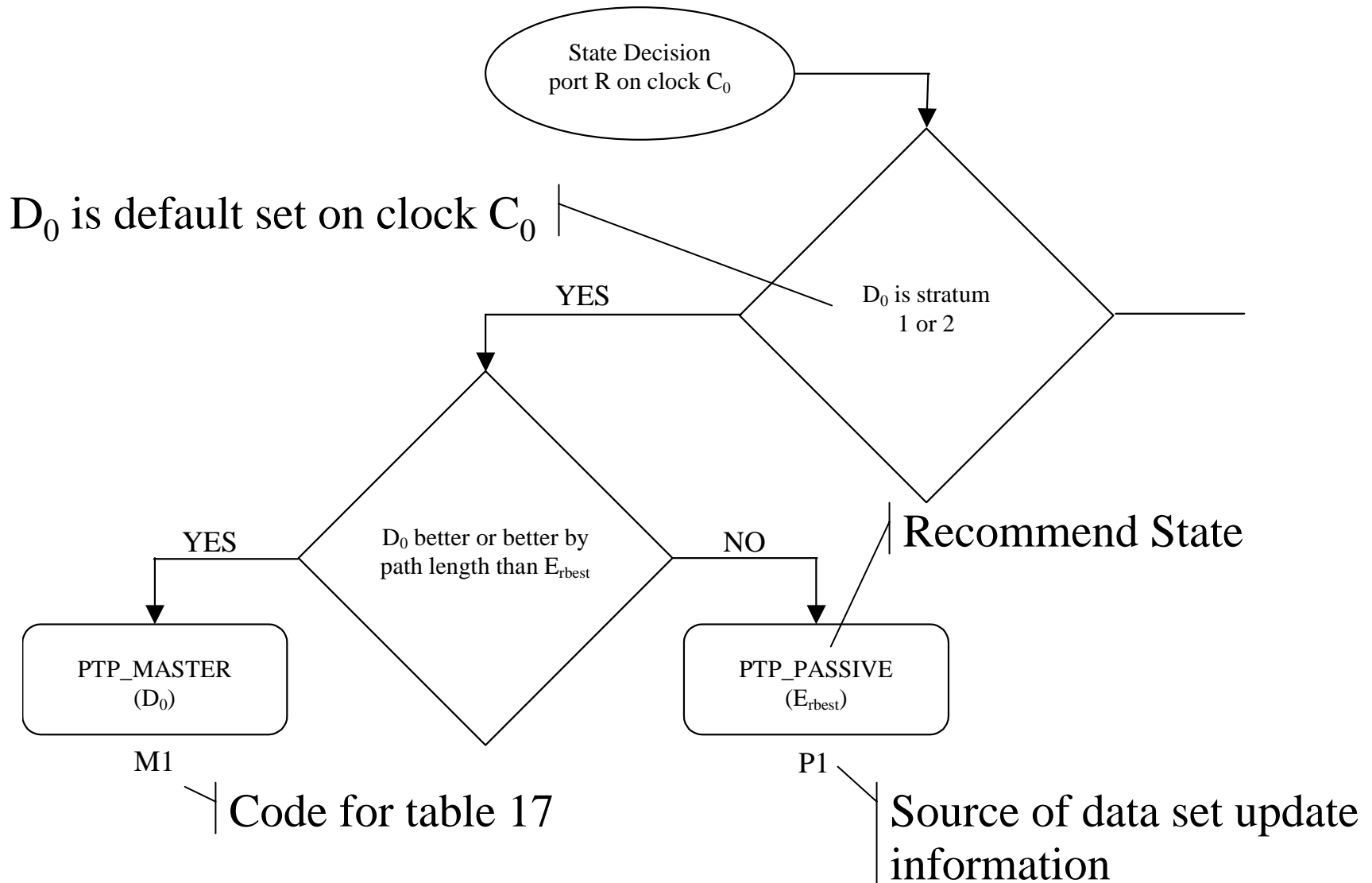
The result of this algorithm is:

- **A ‘recommended state’: drives the state machine of 7.3**
- **Update specification for data sets**

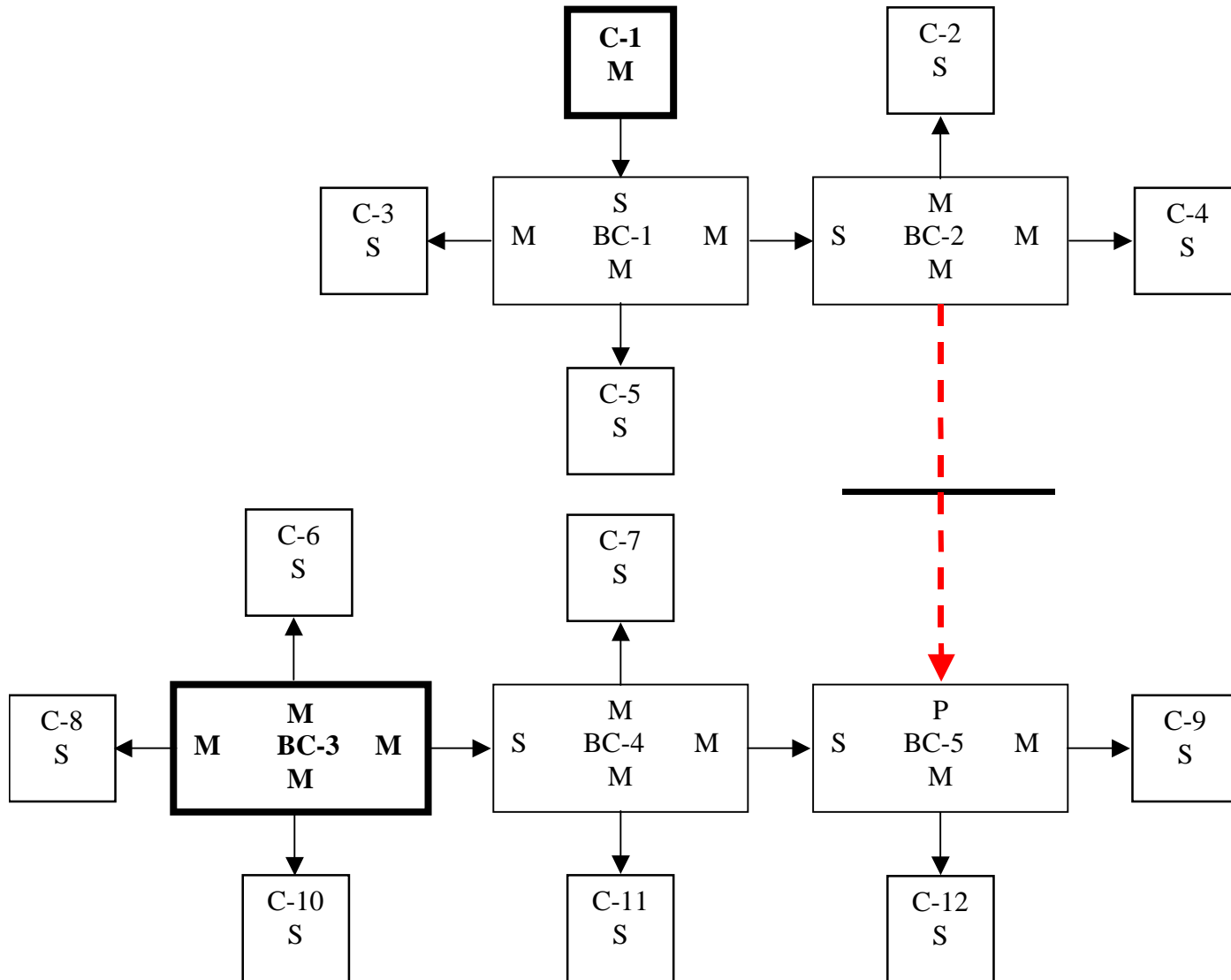
A CAREFUL study of figure 16 reveals all. For example:



State Decision Algorithm (clause 7.6.3 & figure 16)



The Result of the State Decision Algorithm:



Guide to the Standard-

(A detailed analysis of the major clauses)

- a. Overview and goals of the standard
- b. Synchronization messages and methodology
- c. Selection of master clocks
- d. State machine and events**
- e. Timing considerations
- f. Management messages
- g. Time scales
- h. Annex D

State Machine (clause 7.3)

There are 9 states defined in IEEE 1588:

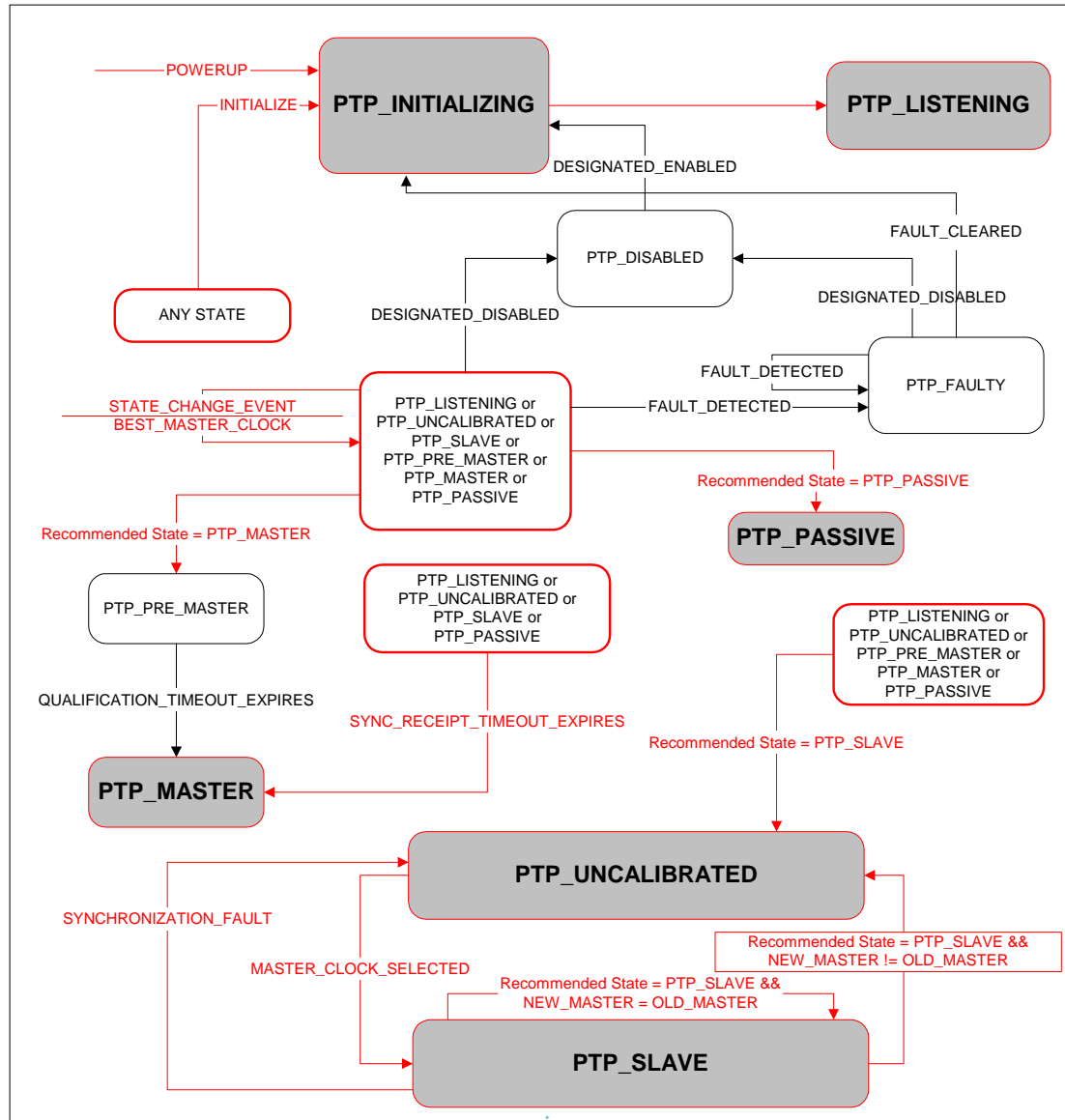
1. **PTP_INITIALIZING**: Initialization- data sets, hardware
2. **PTP_FAULTY**: fault state
3. **PTP_DISABLED**: allows removal of a clock
4. **PTP_LISTENING**: orderly addition of clocks to net
5. **PTP_PRE_MASTER**: transitions in complex topologies
6. **PTP_MASTER**: clock is source of time to its slaves
7. **PTP_PASSIVE**: used to segment network
8. **PTP_UNCALIBRATED**: transition state to slave
9. **PTP_SLAVE**: synchronizing to it's master

State Machine Events (clause 7.5)

There are several events that MAY lead to a state change:

1. **Initialization**
2. Receipt of any message
3. **STATE_CHANGE_EVENT**: clause 7.5.8 (at least once/sync interval !)
4. Transmission of a message
5. **SYNC_RECEIPT_TIMEOUT_EXPIRES**
6. Sync interval timeout expires
7. **QUALIFICATION_TIMEOUT_EXPIRES**
8. **BMC completes**
9. Detection of an internal fault
10. Synchronization changes in a local clock
11. Events related to an external timing signal

State Machine (simplified, clause 7.3, fig 9)



Guide to the Standard-

(A detailed analysis of the major clauses)

- a. Overview and goals of the standard
- b. Synchronization messages and methodology
- c. Selection of master clocks
- d. State machine and events
- e. Timing considerations**
- f. Management messages
- g. Time scales
- h. Annex D

Timing Considerations (clause 7.11)

- **IEEE 1588 timing is centered around the sync interval**
- **Clause 7.11 specifies the rates at which events and messages must be processed by the local clock**
- **The most complex specification deals with how often slave clocks issue Delay_Req messages:**
 - **Randomized to reduce network and master clock processing loads**
 - **Randomization is first over multiple sync intervals and second within the selected interval.**

Guide to the Standard-

(A detailed analysis of the major clauses)

- a. Overview and goals of the standard
- b. Synchronization messages and methodology
- c. Selection of master clocks
- d. State machine and events
- e. Timing considerations
- f. Management messages**
- g. Time scales
- h. Annex D

Management Messages (clause 7.12 & 6.2.2.1)

- **Management messages provide external visibility to the several data sets maintained within each clock**
- **Management messages provide a mechanism to modify certain parameters within these data sets, e.g. `sync_interval`, `subdomain_name`**
- **Management messages provide a mechanism to drive certain state changes. For example initialization, disabling, setting the time in the grand master, ... can be forced using a management message.**



Guide to the Standard-

(A detailed analysis of the major clauses)

- a. Overview and goals of the standard
- b. Synchronization messages and methodology
- c. Selection of master clocks
- d. State machine and events
- e. Timing considerations
- f. Management messages
- g. Time scales**
- h. Annex D

IEEE 1588 Time Scales (Annex B)

- **The time base in an IEEE 1588 system is the time base of the Grandmaster Clock. The **epoch** and **rate** is determined by the grandmaster.**
- **All other clocks synchronize (perhaps via boundary clocks) to the grand master.**
- **The Grandmaster Clock time base is implementation and application dependent.**
- **If the Grandmaster Clock maintains a UTC time base, the IEEE 1588 protocol distributes leap second information to the slaves if it is available.**



IEEE 1588 Time Scales (Annex B): EPOCH

Time Base	IEEE 1588 User Defined	IEEE 1588 UTC	GPS UTC	NTP UTC
Epoch	User Defined	0:00:00 1 January 1970	0:00:00 6 January 1980	0:00:00 1 January 1900
Rollover Frequency	~9x10 ⁶ years	~9x10 ⁶ years	1024 weeks (~2019)	136 years (~2036)
Linear Time Base	Yes	Yes (offset TAI)	Yes (offset TAI)	No (leap second discontinuity)
Civil Calendar Events	Hard leap seconds	Hard leap seconds	Hard leap seconds	Easy
Duration & Relative Events	Easy	Easy	Easy	Hard leap seconds

Guide to the Standard-

(A detailed analysis of the major clauses)

- a. Overview and goals of the standard
- b. Synchronization messages and methodology
- c. Selection of master clocks
- d. State machine and events
- e. Timing considerations
- f. Management messages
- g. Time scales
- h. Annex D**

ANNEX D: IEEE 1588 on UDP/IP ETHERNET

- 1. Defines the message time stamp point: The start of the first bit of the octet following the start of frame delimiter**
- 2. Defines relevant fields in Ethernet header**
- 3. Defines the mapping of clause 8 messages onto Ethernet frame user space**
- 4. Defines IEEE 1588 uuid_field when using Ethernet: Based on Ethernet MAC address**

ANNEX D: IEEE 1588 on UDP/IP ETHERNET

(continued)

5. Defines IEEE 1588 addressing when using UDP/IP on Ethernet

Port category	IANA Name	Value
EventPort	Ptp-event	319
GeneralPort	Ptp-general	320

Address name	IANA Name	Value
DefaultPTPdomain	PTP-primary	224.0.1.129
AlternatePTPdomain1	PTP-alternate1	224.0.1.130
AlternatePTPdomain2	PTP-alternate2	224.0.1.131
AlternatePTPdomain3	PTP-alternate3	224.0.1.132

AND!

Everything covered so far exists within a scope.

The scope is defined by the value of the subdomain_name parameter of the default data set. (clauses 6.2.5 & 7.4.2)

All activity such as messages, time base, state machines, etc. in one subdomain is completely independent of similar activity in another subdomain, even on the same network medium.



IEEE 1588 Interoperability/Conformance Topics

- 1. Interoperability and conformance are NOT the same thing!**
- 2. Clause 9 defines three levels of clock conformance and a minimal set of system conformance requirements.**
- 3. Individual clock conformance:**
 - a. Fully conformant: meets all aspects of IEEE 1588 standard**
 - b. Slave only: Always defers to E_{best} (clause 7.6) as selected by BMC algorithm. Never issues Sync, Delay_Resp, or Follow_Up messages**
 - c. Management only: Only issues management messages.**
- 4. System: Conformant clocks, One fully conformant clock, common system parameters, no non-specified transport links**

IEEE 1588 Interoperability/Conformance Topics

IEEE 1588 network messages represent the critical interface to an IEEE 1588 clock port.

Detailed network independent specifications on the fields, meanings, data types, etc. for each of the 5 defined IEEE 1588 messages are given in clause 8.

Specific mappings of the message specifications onto a particular network transport are defined in Annexes to the standard.

Currently the only such mapping is to UDP/IP on Ethernet.

Implementation Topics

- 1. Minimal implementations**
- 2. Accuracy issues**
- 3. Application level support**

Minimal Implementations

IEEE 1588 specifies very few optional features:

- **Slave only nodes (conformance clause 9.2.2)**
- **Follow_Up capable (clause 6.2.4.6). This is tied to the issue of hardware assist in time stamping Sync and Delay_Req messages.**
- **External timing signal (clause 7.5.20)**
- **Burst mode (clause 7.5.5, 7.5.9, 7.5.11)**
- **Parent statistics (clause 7.4.4.8)**

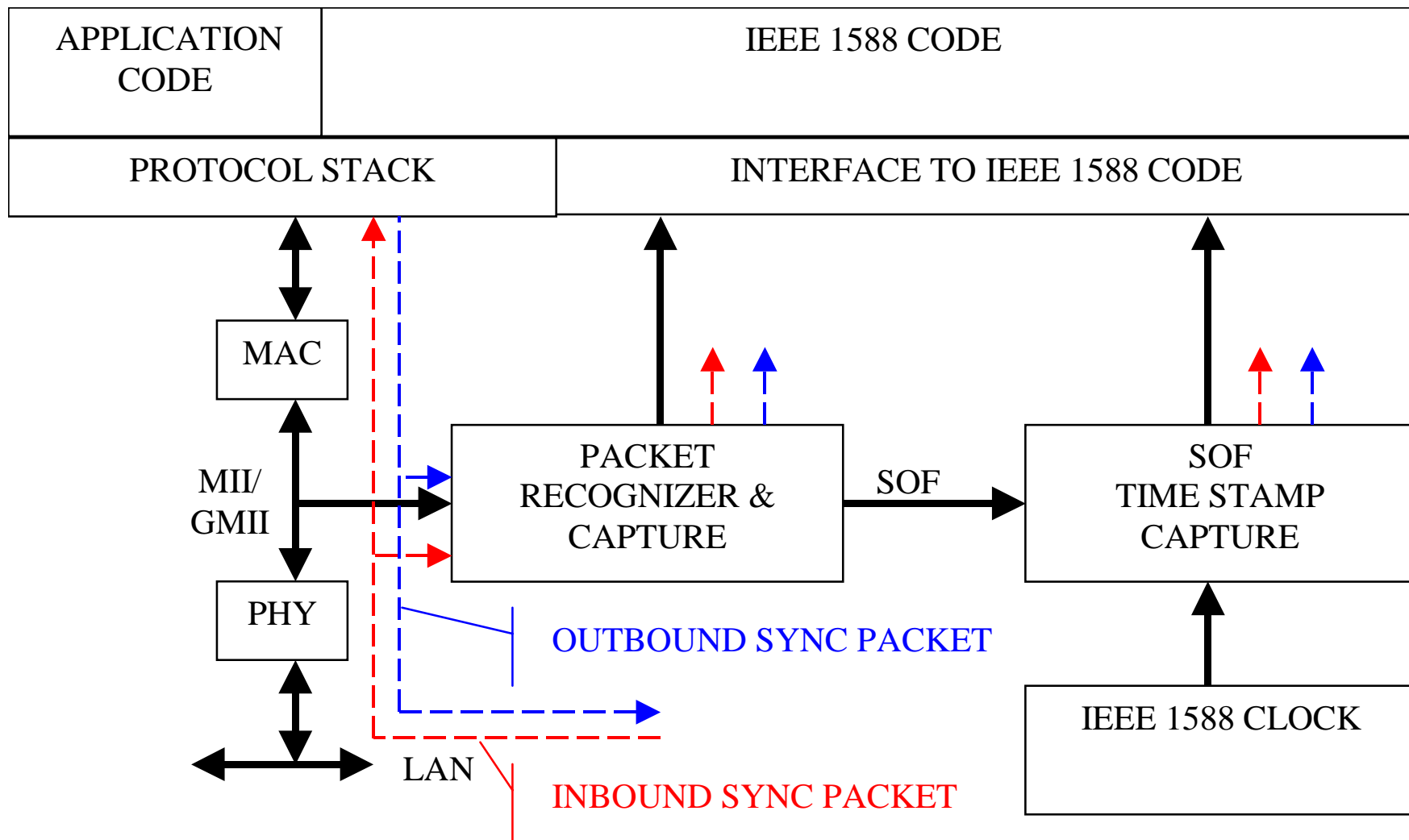
Minimal Implementations (Follow_Up capable)

1. **Clocks generate a **time stamp** when a Sync message is sent or received.**
2. **Can be done in hardware (e.g. at MII and is the most accurate), ISR or kernel level, or at application level (least accurate)**
3. **Can be communicated:**
 - a. **In Sync message: Requires on-the-fly message modification**
 - b. **In a Follow_Up message: Easy to insert but requires IEEE 1588 code to keep track of pairs of messages.**

Accuracy Issues (hardware assist)

- 1. Hardware assisted generation of time stamps is potentially the most accurate.**
- 2. Requires attention to latency (clause 6.2.4.9) and message time stamp point (clause 6.2.2.3)**
- 3. In addition to capturing the time stamp enough information must be captured to enable IEEE 1588 code to associate the time stamp with the correct Sync message**
- 4. Must differentiate between IEEE 1588 Sync (or Delay_Req) messages and other traffic.**

Accuracy Issues (hardware assist)



Accuracy Issues (oscillators)

- 1. IEEE 1588 is all about reducing timing fluctuations:**
 - a. In the protocol stacks: hardware assist**
 - b. In network components: boundary clocks**
- 2. The final reduction technique is statistics:**
 - a. Pre-filtering of raw clock offset data**
 - b. Design of the servo in the slaves**
- 3. Clocks must be sufficiently stable to support the statistic given sync_interval, fluctuation level, and desired accuracy.**

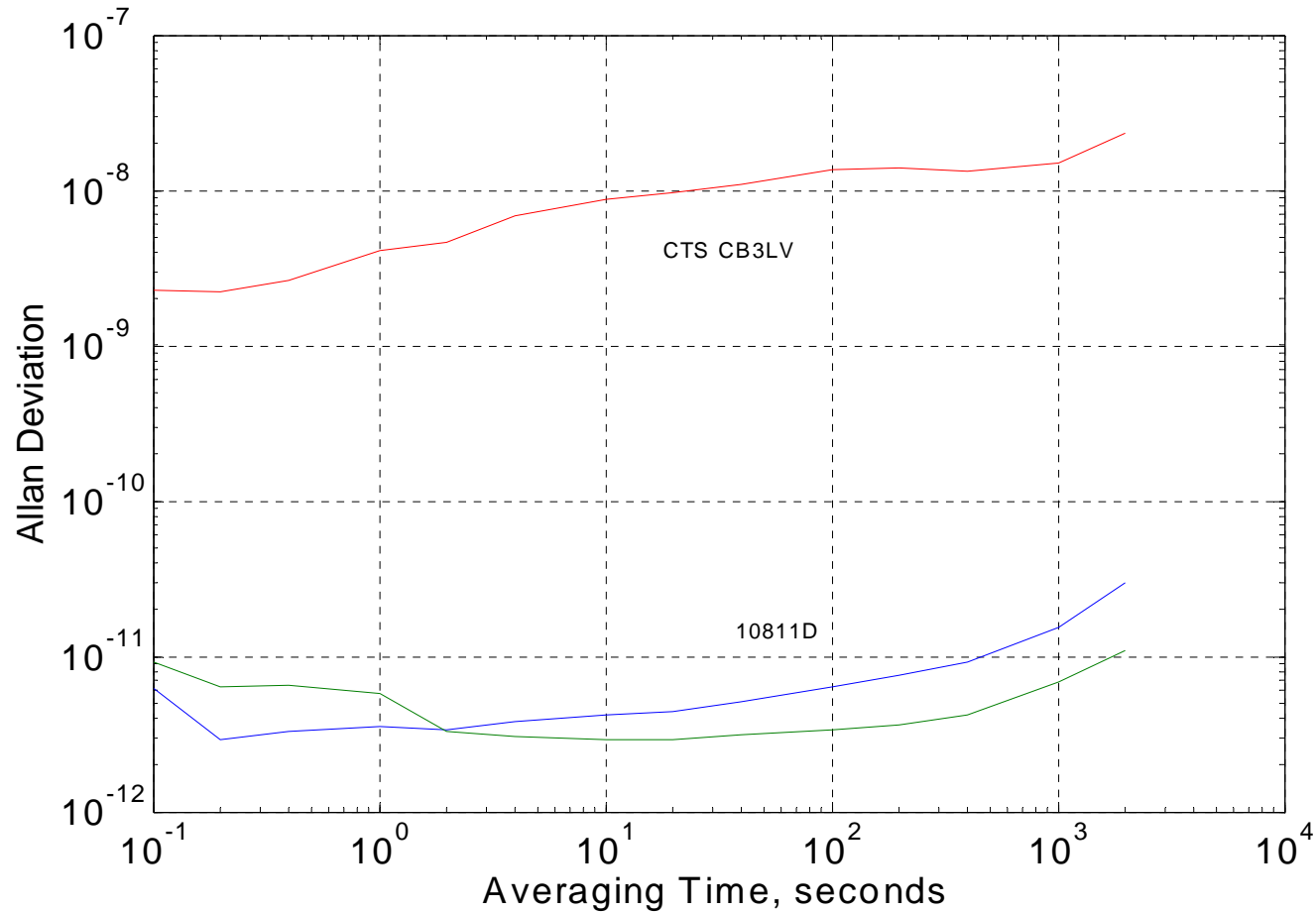
Accuracy Issues (oscillators)

Experience has shown:

- **Accuracy ~100 ns level is achievable with 2 second updates, inexpensive oscillators, compact topologies with lightly loaded switches, and simple PI servos for averaging.**
- **Accuracy <20 ns will require some combination of faster sampling, better oscillators, boundary clocks, sophisticated statistics and servo algorithms and careful control of environment **especially temperature.****



Allan Frequency Deviations for Two Oscillators



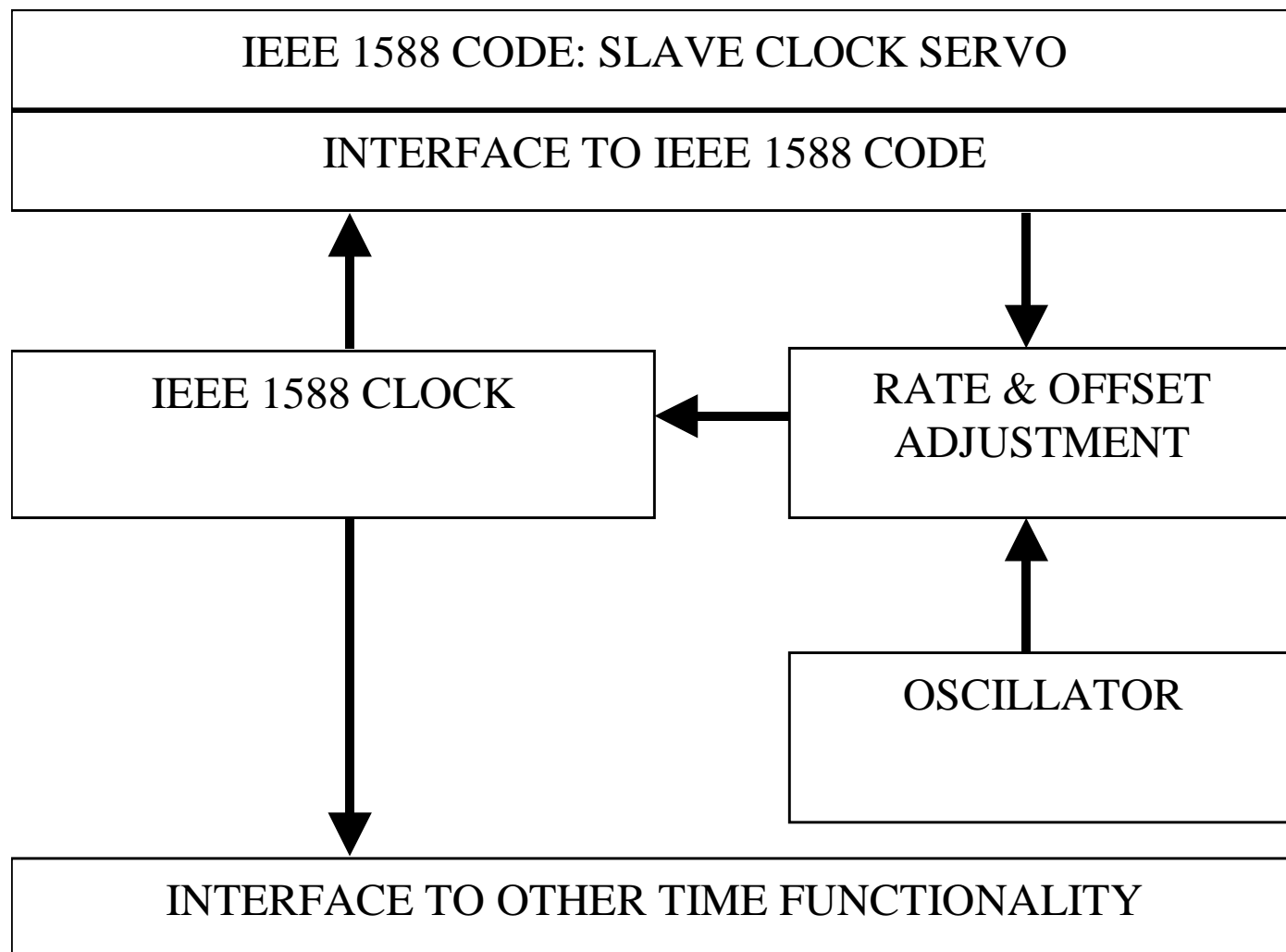
Accuracy Issues (asymmetry)

- 1. Path asymmetry introduces offset errors**
- 2. Whether asymmetry needs to be considered depends on the network topology and implementation and the desired accuracy**
- 3. The major source of asymmetry in a complex network is different path lengths in the master to slave and slave to master directions. This can result from queuing differences in switches/routers or in actual routing differences.**
 - a. Control routing**
 - b. Measure and correct for delay**

Accuracy Issues (asymmetry)

- 4. Physical media can also be asymmetric**
 - a. CAT5 cable asymmetry is nominally 25-50ns/100m**
 - b. Measure and correct for delay**

Accuracy Issues (clock design)



Clock Design Issues

1. IEEE 1588 Clock

- a. **Width:** how many bits of seconds, resolution, rollover
(**Y2K**)
- b. **Representation:** binary, BCD, sec/ns vs. ns

2. Rate & offset adjustment

- a. **Rate range:** must allow for maximum offset specification on oscillators ($\pm 0.01\%$)
- b. **Minimum correction:** Consistent with desired accuracy, e.g. 1 part in 10^9
- c. **Offset correction:** Must allow gross error correction on transients

3. Interface to IEEE 1588 code

4. Interface to other time functionality

Clock Design Issues

IEEE 1588 Code: **Slave clock servo**

- 1. Servo input is the ‘offset’ computed from time stamps of Sync and Delay_Req messages exchanged between master and slave**
- 2. Typical implementations of the servo use a PI (proportional integral) control strategy**
- 3. Usual issues of servo stability, parameters, wind-up, outliers in error input,...**



Clock Design Issues

IEEE 1588 Code: **Grandmaster clock**

1. **The grandmaster clock determines the time base for the entire system.**
2. **The grandmaster clock MAY itself synchronize to a source of time EXTERNAL TO THE IEEE 1588 system:**
 - a. **Application time base (within the tolerance of the IEEE 1588 system)**
 - b. **GPS, NTP, or other recognized UTC time base. In all cases but especially with NTP a ‘flywheel’ will be needed to average out fluctuations of the source to the desired accuracy of the IEEE 1588 system.**

Clock Design Issues

IEEE 1588 Code: Grandmaster clock (continued)

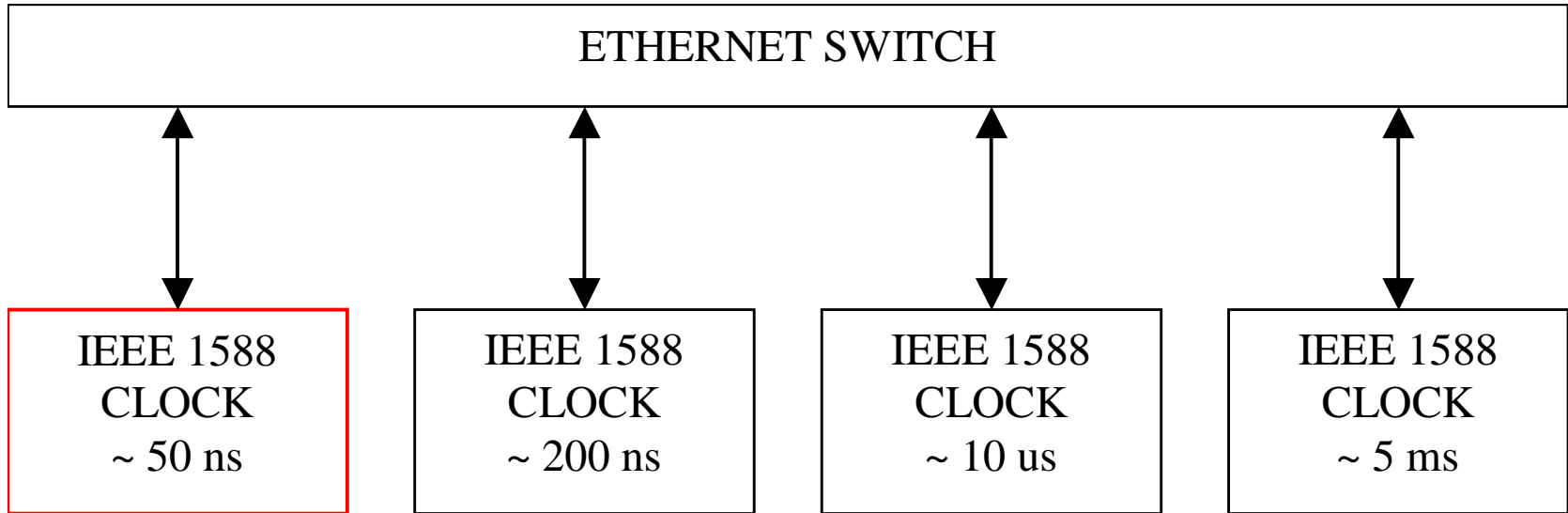
Synchronization to an external source can be implemented using the clock servo normally used when in the slave state by:

- 1. Generating an error signal representing the offset between the IEEE 1588 grandmaster clock and the external source. External sources, e.g. GPS, typically provide a 1 PPS signal useful for this purpose.**
- 2. Applying the error signal to the grandmaster clock servo.**



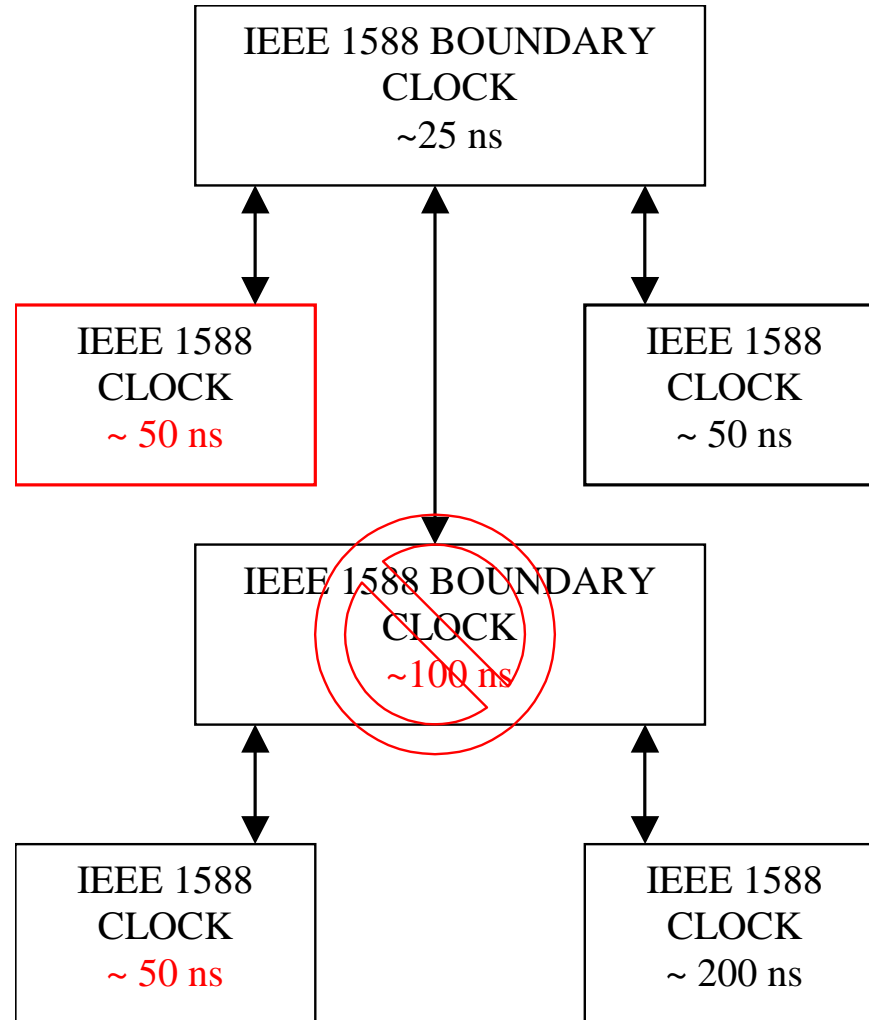
Accuracy Issues (topology)

Single subnet: no problem



Accuracy Issues (topology)

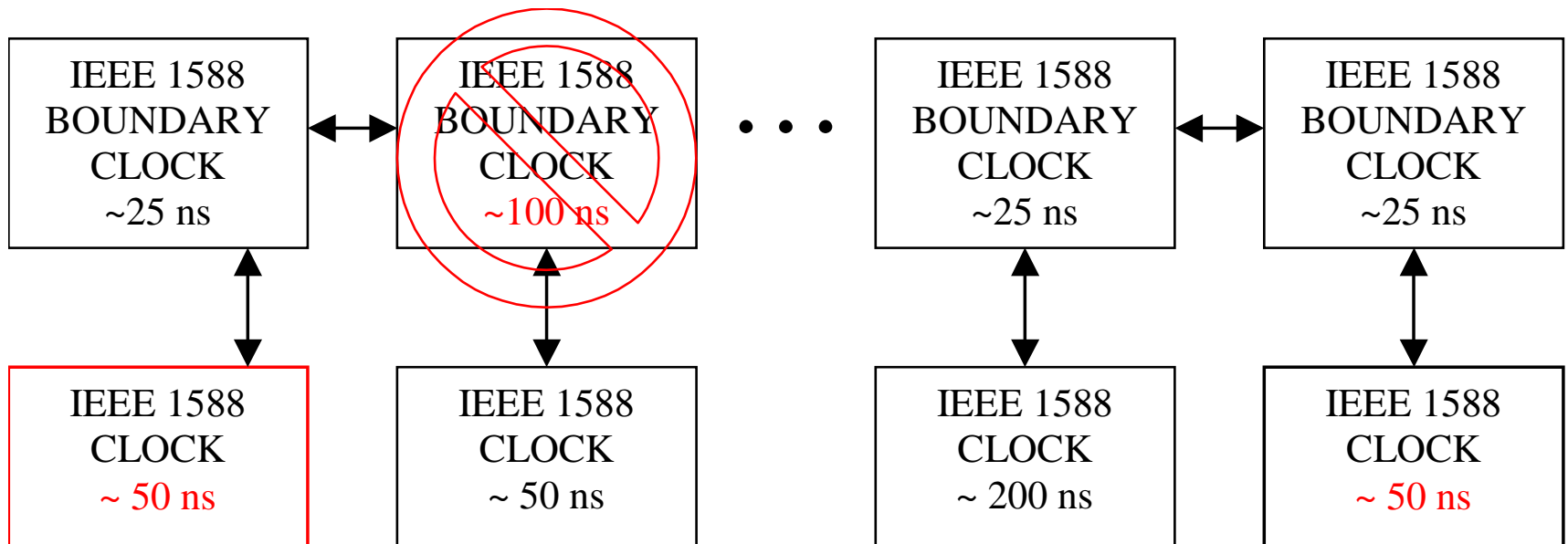
Hierarchy: Be Careful!



Accuracy Issues (topology)

Linear: Be Careful!

1. Cascaded devices accumulate servo error & quantization errors
2. Low accuracy intermediate devices dominate error budget of chain



Application Level Support

How do applications interface to an IEEE 1588 clock?

1. Time stamp events

2. Generate events

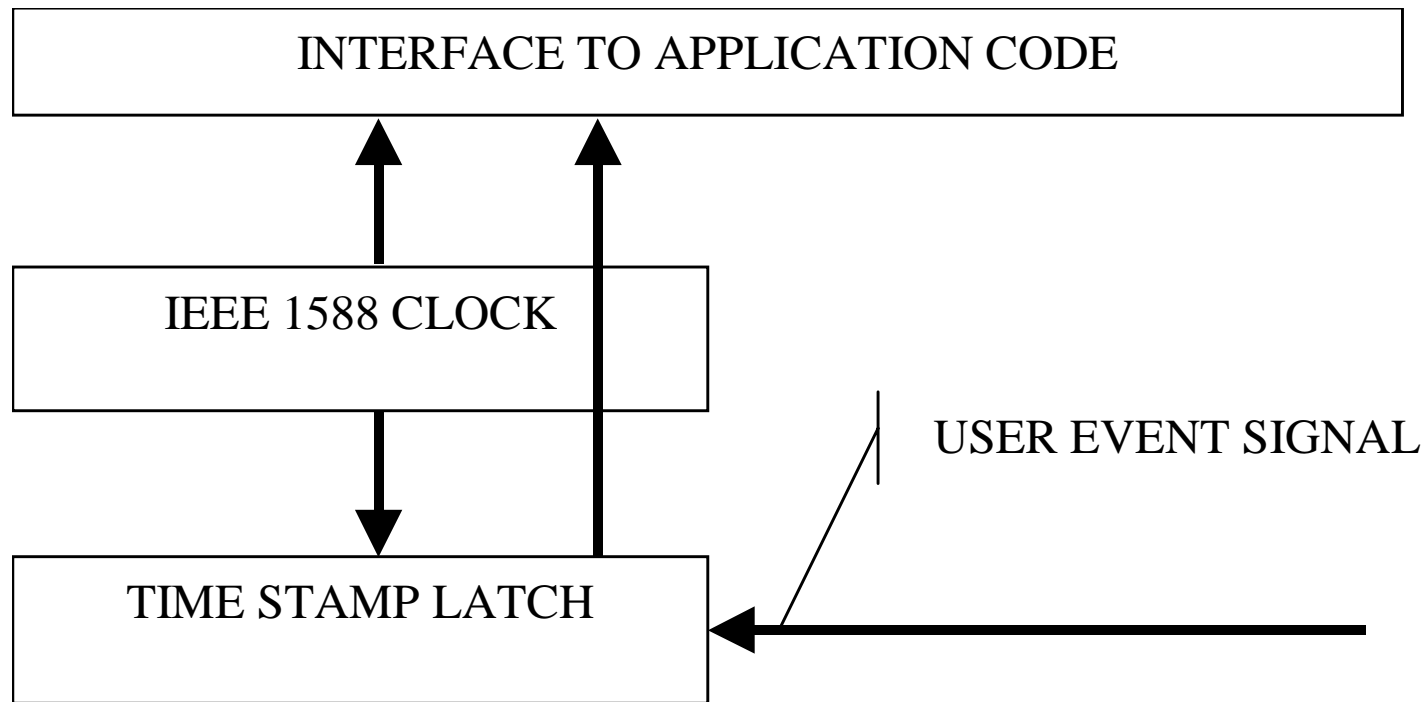
3. Generate waveforms

In each case the application signals in one device will be correlated in time with those in other devices within the synchronization accuracy of the underlying IEEE 1588 clocks.



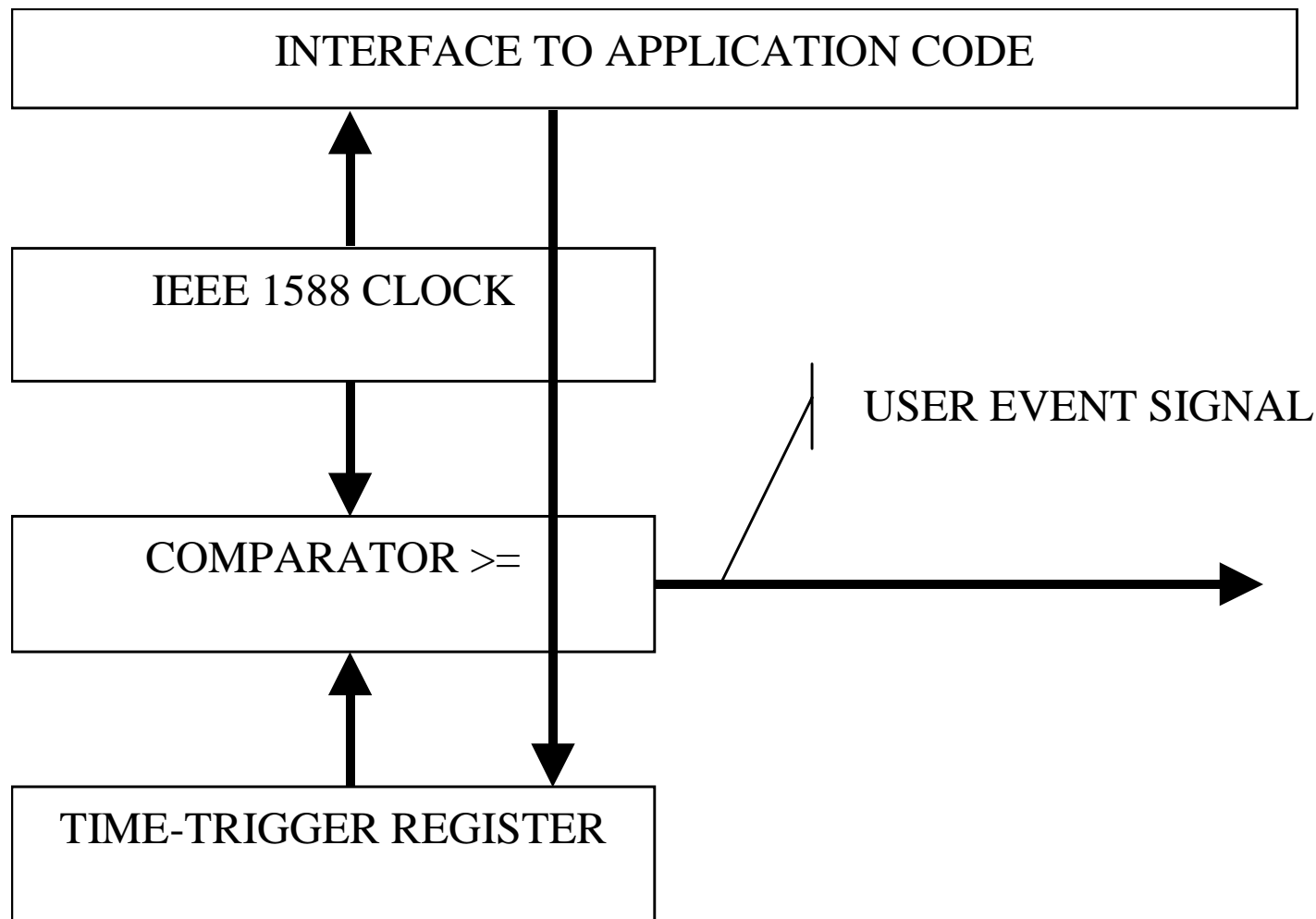
Application Level Support

How to time stamp events:



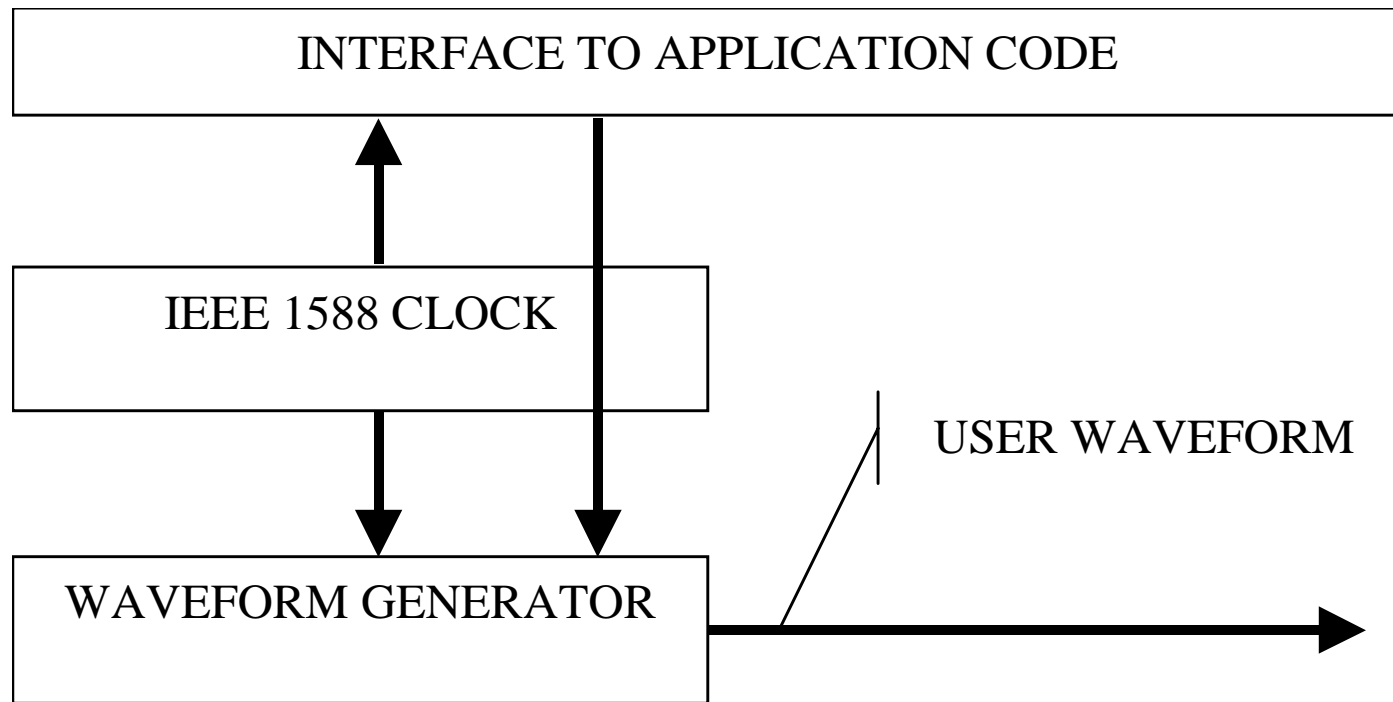
Application Level Support

How to generate events (time-triggers):



Application Level Support

How to generate waveforms:



Extensions to IEEE 1588 version 1 in PAR of the P1588 Committee



P1588 PAR Topics

- 1. Resolution of known errors:** A list of these and recommended solutions is posted on the IEEE 1588 web site. <http://ieee1588.nist.gov> These are not expected to have appreciable impact on existing implementations.
- 2. Conformance enhancements:** 1 PPS or equivalent signal, management message or extension fields to make internal time stamps visible.
- 3. Enhancements for increased resolution and accuracy:**
 - Extension fields to allow sub-nanosecond time stamps,
 - **shorter sync_intervals** allowed.
- 4. Increased system management capability:** Additional management messages, perhaps SNMP

(items in red may substantially impact version 1 operation or compatibility)

P1588 PAR Topics (continued)

5. **Mapping to DeviceNet: Few if any changes required in body of standard**
6. **Annex D modifications for variable Ethernet headers: Likely additions are tagged frames and IPV6. These could impact existing packet recognition designs and protocol stacks.**
7. **Prevention of error accumulation in cascaded topologies: New clock type (transparent clock), topology and system design guidelines.**
8. **Rapid network reconfiguration: Path delay measurements and correction of timestamps.**
9. **Ethernet layer 2 mapping**



P1588 PAR Topics (continued)

- 10. Optional shorter frame:** Must resolve needs of industrial and telecommunication applications.
- 11. Extensions to enable implementation of redundant systems:**
 - **Master clock failure and network failure.**
 - **Redundant grandmaster clocks, and/or**
 - **Slave selection of grandmaster clocks.**
- 12. Security extensions: authentication of grandmaster,...**
- 13. Extension mechanism: Uniform way of **extending** fields/messages.**

IEEE Procedures to Revise/Update the Standard

- 1. IEEE sponsor (Kang Lee for TC-9 of I&M Society) appoints chair of working group.**
- 2. Solicit membership in working group.**
- 3. Draft and submit PAR (project authorization request) to the IEEE**
- 4. PAR approval (March, 2005)**
- 5. Develop revised standard (12-18 months)**
- 6. Submit to IEEE ballot process (~ 3 months)**
- 7. Revise/re-ballot if necessary**
- 8. Editorial/publish process with IEEE (~ 3 months)**



Questions?

