16.07.2012

DRAFT

Dual Time Scale in Factory & Energy Automation
V1.02

White Paper about Industrial Requirements and Concepts @ Time Synchronization

(IEEE 802.1ASbt)

Authors and contributors:

Franz-Josef Goetz, I IA ATS, Siemens AG, (franz-josef.goetz@siemens.com)

Clemens Hoga, IC SG AE, Siemens AG, (Clemens.hoga@siemens.com)

Dr. Karl Weber, Zurich University of Applied Sciences, (Karl.Weber@ieee.org)

1 Introduction

During the last decade, lots of work has been done in order to increase the accuracy of time sync via Local Area Networks. A major step was the involvement of bridges to eliminate the variance of transmission time of delay request and delay response caused by the nature of unpredictable Ethernet traffic. Specified in IEEE 1588, the current version V2 offers comprehensive functions to fulfill almost all requirements of today's applications. Unfortunately, the increasing number of different application fields led to a number of profiles and derivates, which imply incompatibilities in detail. An additional challenge, the mandatory high performance media redundancy of such Industrial Ethernet applications increases the complexity. On the other hand, lots of successful implementations and plugfests, which have demonstrated the interoperability of the dedicated solutions, have already created the new potential fields of application for IEEE 1588. Today, it is possible to support different PTPv2 profiles in one switch or edge device, however this is not quite optimal.

Currently in IEEE 1588 a discussion about PTP version 3 is open. Within IEEE 802.1 AVB task group requirements for industrial are collected for further discussion in PTP working group.

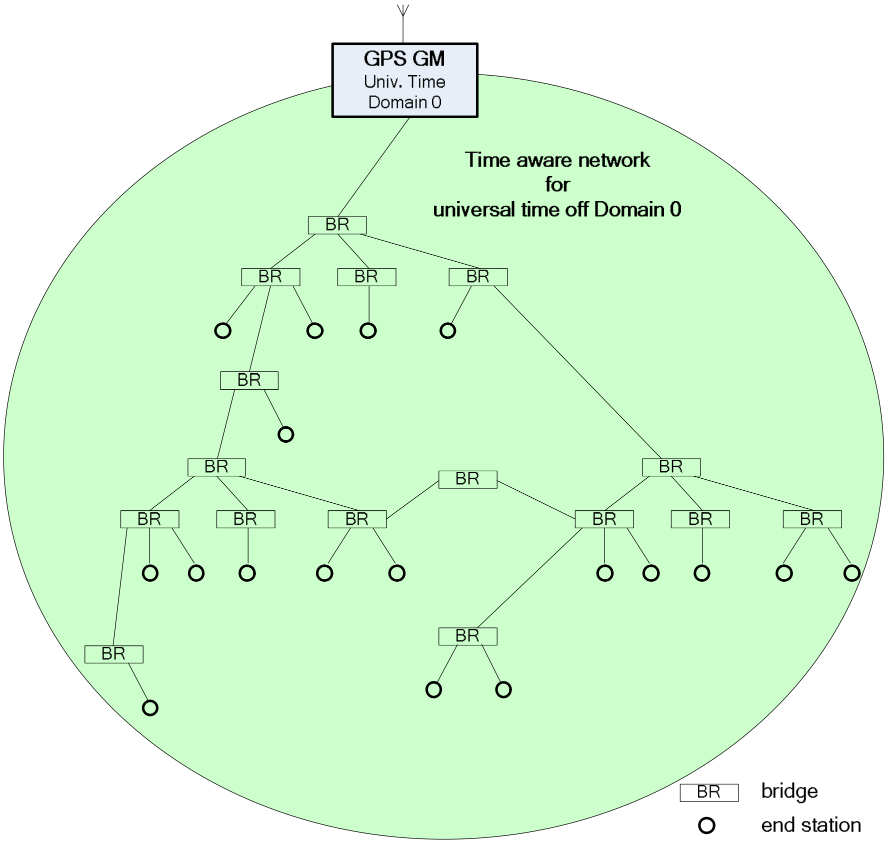
This paper elaborates the typical use cases in the various industry domains and introduces a novel way how to structure the time sync domains by introducing so called "working clocks".

This paper also describes how time synchronization is used in factory automation. Typical applications using time synchronization over Industrial Ethernet are introduced and their requirements will be described. The reasons why dual time scales may be necessary are listed.

If this paper will be officially published, I would make a note here that a reader is assumed to be familiar with IEEE1588 and 802.1AS standards. Otherwise you may need to define universal time, PTP, gPTP and other terms and abbreviations.

2 Universal time

Typically universal time is distributed by GPS satellites. To make universal time available in a factory network GPS receivers are used. The gPTP protocol is used to distribute universal time over bridged network.



The increasing production speed and requirements for high product quality are among reasons for increasing requirements on higher accuracy for synchronization of universal time. These increased requirements can not met by Network Time Protocol (NTP) or Simple Network Time Protocol (SNTP).

2.1 Reasons for using gPTP (IEEE 802.1 AS-2011) to Synchronize Universal Time

- Universal time should be available over the whole network
  - One common sync domain

- Little configuration effort (plug & play)

- Inherent loop prevention mechanism (Best Master Clock algorithm used creating sync tree)

- Use COTS bridges with little hardware support and low CPU utilization
  - Announce message is used to establish port roles for sync tree with fast configuration and reconfiguration of sync tree
  - The cumulative frequency offset mechanism and the sync tree mechanism guarantees fast startup und fast reconfiguration
  - Only one reserved group multicast address for all gPTP messages, all messages are peer-to-peer messages (announce, sync, follow up, P-delay request, P-delay response, P-delay follow up response, signaling)

  - o Peer-to-peer path delay measurement is time scale independent
    (free running timer / counter are used for peer-to-peer path delay measurement)
  - o Sync messages have to follow sync tree
  - o Sync messages are only forwarded over links
    - which supports the path delay measurement and
    - where path delay measurement was successful
  - o Only one sync message per port within one sync interval (no overload) at startup

- Can cross router borderlines with gPTP capability
  (forwarding mechanism for gPTP messages is independent of L2 and L3 forwarding mechanism because it has specified own forwarding rules for announce, sync and follow up messages by best master clock algorithm (BMCA))

2.2 Typical Applications Using Universal Time for Time Stamping

- **Universal Time (wall clock)**

- **Sequence of events or events**

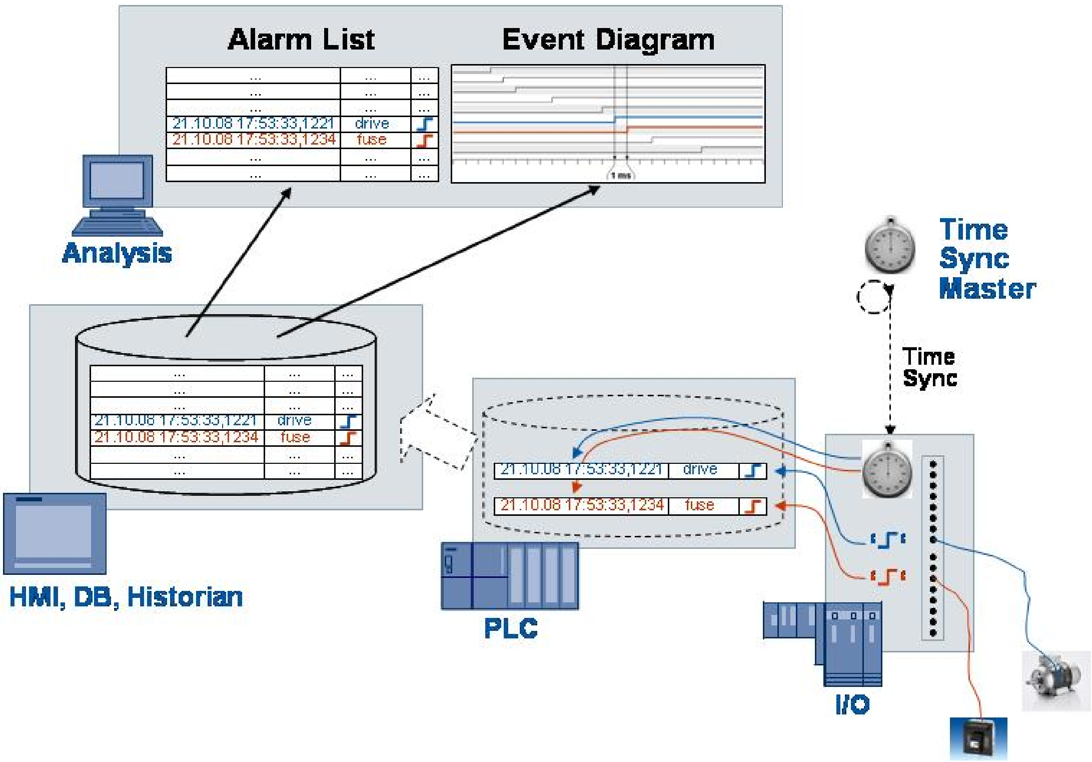
- **Latency measurement**

- **Measurement systems (sampled values)**

- **Time stamp production data**

- **…**

Detailed view for use case “sequence of events”



Distributed systems which are composed of Actuators, Sensors, PLCs and other nodes are time stamping events. All events are stored in a database. Analysis tools visualize the chronological sequence of the events.

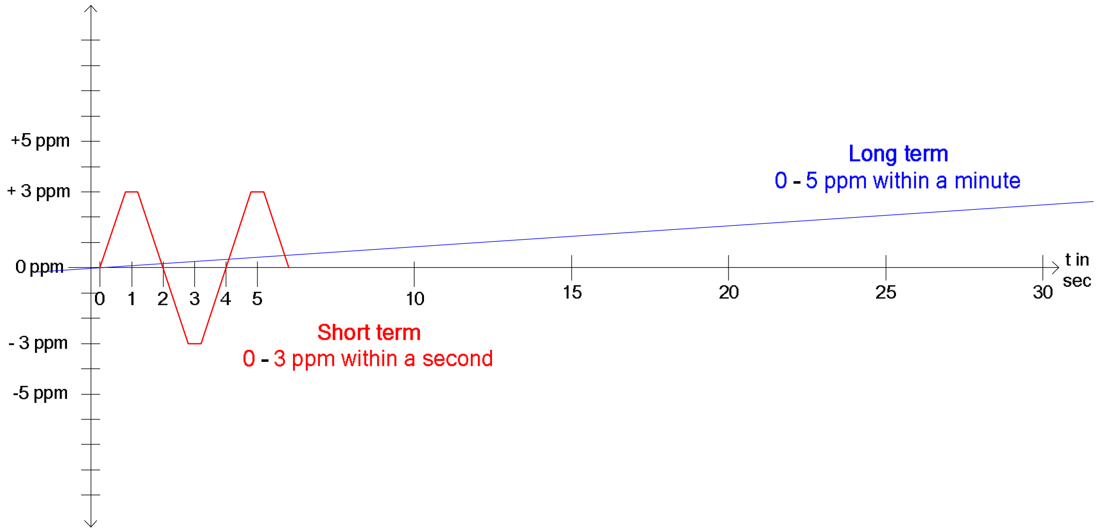
2.3 Requirements for Synchronizing Universal Time

Typically only few components (e.g. GPS receiver) within a network can distribute traceable universal time.

1) Accuracy
- Accuracy <100µs over 128 hops @ industrial automation
- Accuracy <1µs over 16 hops @ energy automation (IEEE C37.238-2011 standard)

2) Interval for sync messages is 125ms (default for gPTP)

3) Frequency change (Δf / f) / +Δt @ industry
- Short term 0 – 3 ppm within a second
- Long term 0 – 5 ppm within a minute



4) End-to-End GM rate measurement (to follow frequency GM change very quick)

5) Plug & play

6) Usage of low cost oscillators in end stations and bridges
(typically +/- 25ppm deviation form nominal frequency)

7) Guaranteed latency for sync messages to minimize PLL reaction time (10ms / hop)

8) Open standard (e.g. IEEE)

9) Independent loop prevention mechanism

10) Media independent and also long distance
- o ***Wired***
  - Long distance with fiber optic (multi mode, single mode)
  - Polymeric optical fiber
  - Copper
  - …
- o ***Wireless***
  - Wi-Fi (Wireless LAN, IEEE 802.11)
  - WPAN (Wireless Personal Area Networks IEEE 802.15)
  - …

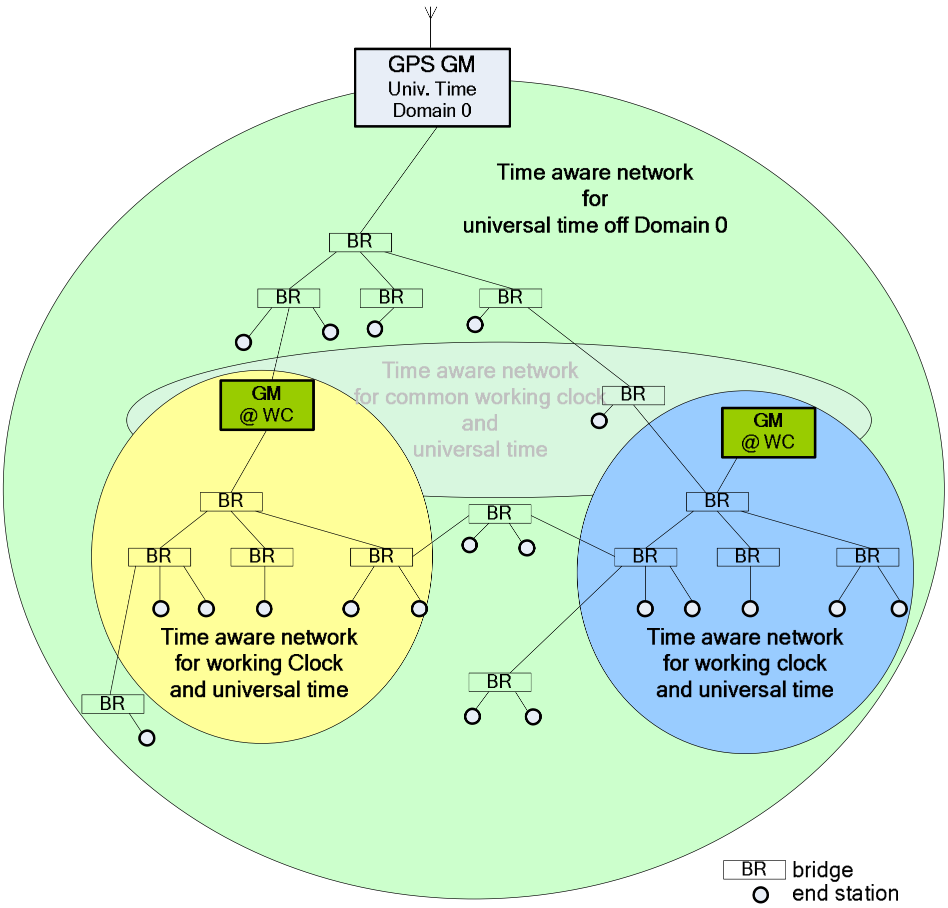
11) When different network parts are joined to one network automatically reconfiguration for synchronization is expected

12) Universal time shall be able to cross IP router borderlines

13) Security concept for universal time synchronization

3 Working Clock

Typically working clock is distributed by PLC's in factory automation. PLC's are used as clock source to distribute local time as working clock within a working clock domain. A working clock domain covers only a restricted area of a factory network. Within a factory network there can exist multiple independent working clock domains but also the hierarchical (max. two hierarchical level) working clock domains are conceivable.



Unlike the universal time domain, a working domain is typically engineered and configured (PLC's, sensors and actuators which belong to a working clock domain). Clock source for working clock is a local oscillator..

3.1 Reasons for Using Working Clock

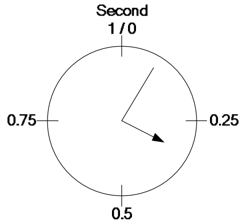
- Synchronization of scheduled control data traffic
  - o Time aware traffic shaper in end stations
  - o Time aware blocking shaper in bridges (if required)

- Synchronization for data sampling
  - o Input system (e.g. sensors of an Energy Automation Process Buss IEC 61850-9-2)

- Synchronization of actuators
  - o Output System

- Synchronization of applications
    - o Motion control loop

- Different cyclic time-scales for working clock (e. g. 1 second in energy automation)

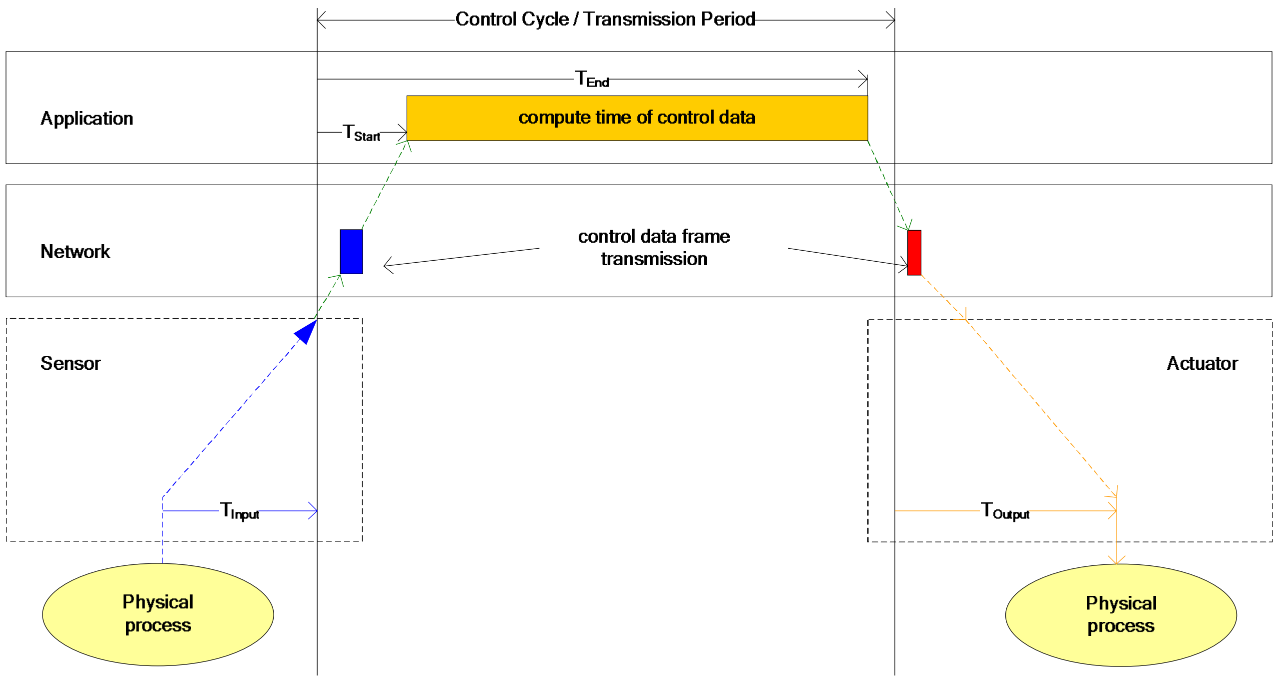


Some applications e.g. energy automation for working clock they do have no need for time of day information. Working clock wrap around at 1 second is sufficient.

3.2 Use Case for Working Clock

3.2.1 Motion Control Application

The following figure shows a typical traffic pattern for motion control applications. Motion control applications are closed control loops. Within each control cycle before a motion control application can compute new output data for actuators sensor data must be exchange over network.



- Synchronized measurement of sensor control data
- Scheduled transmission of control data traffic (simultaneously input and output control data)

3.2.2 Scheduled Control Data Traffic

Control data traffic specifies a time sensitive traffic class for control data with guaranteed quality of service (QoS). In industrial automation control data are exchange between PLC's, actuators and sensors.

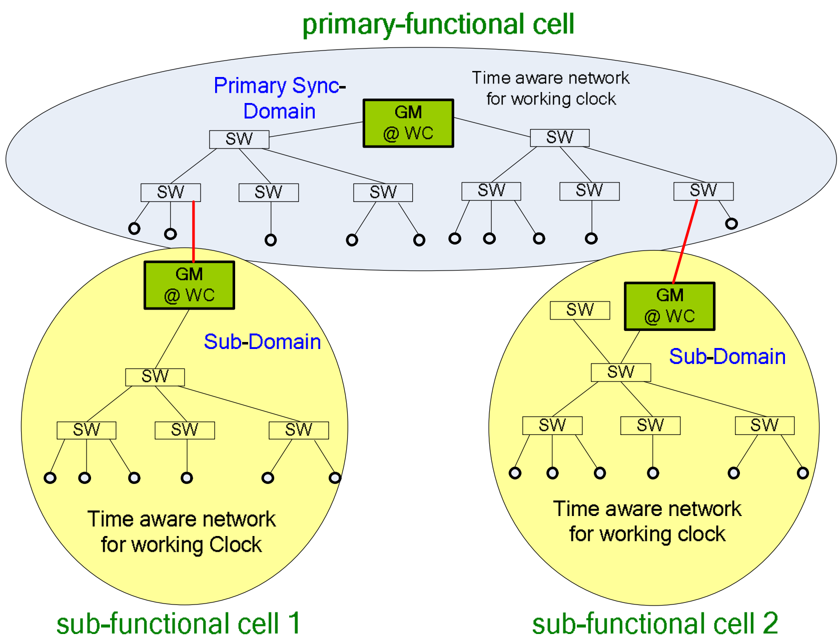
- To avoid packet lost in bridges,

- to guarantee latency control data traffic and
- to minimize time for exchange a certain amount of control data,

In convergent networks control data traffic are scheduled (transmission time and transmission) in end stations. Time based control data transmission in end stations helps to minimize make span and resources for control data within bridges.

3.2.3 Joining and Separating Synchronization Islands

3.2.3.1 Industrial Automation

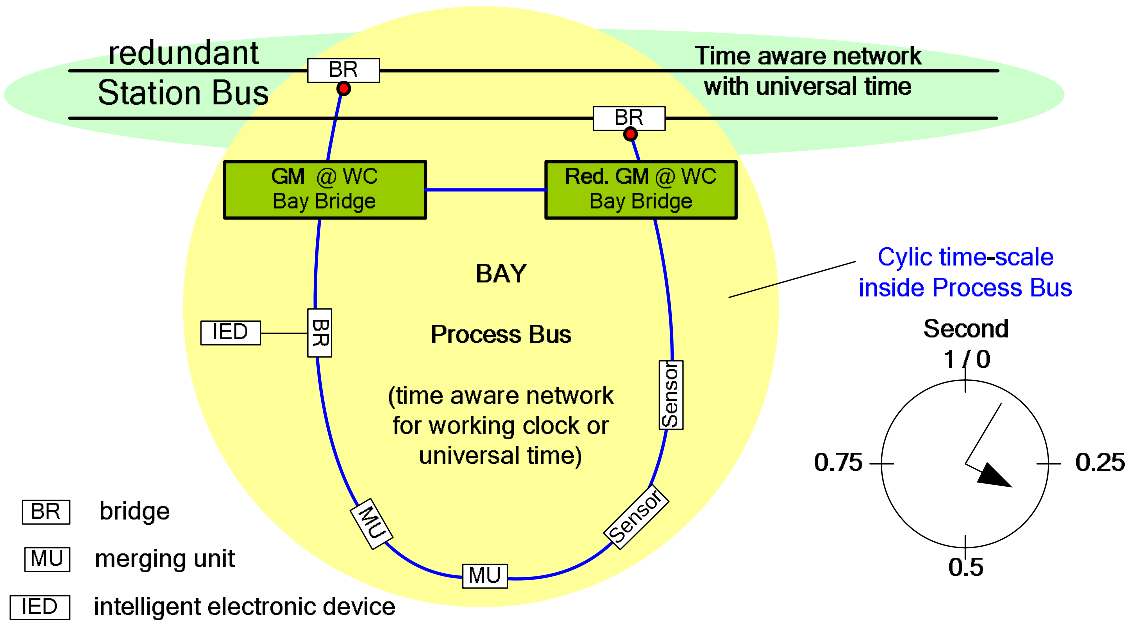


When independent synchronized sub-functional cells are joined to a primary functional cell, merging to one working clock domain should happen manually and be driven without reconfiguration for synchronization. As long as an operator has not approved merging to one working clock domain, synchronized sub-functional cells should work independently from each other.

Typical use case:

- Pre-commissioning for functional cells
- Printing machines with multiple printing and folding units
- Production lines which consists of a lot of different components

3.2.3.2 Energy Automation



The typical cyclic time-scale for working clock in energy automation's Process Bus is one second. A Process Bus is a subnet in a substation, typically allocated to branch of the substation (a BAY) with its voltage and current sensors. Currents and voltages are typically sampled with 4 kHz, each sample tagged by a sample counter. This sample counter will be reset every second and incremented every sample. If a protection relay (IED) needs the input of different sensors, it can explicitly identify the different values sampled at the same point of time.

As long as BAY's are not connected to the station bus, they are operating independently. While operating independently, all components inside a BAY are synchronized with working clock (local time).

When a BAY is connected to the station bus, which provides universal time, all components within a BAY shall be synchronized to universal time (Some protection functions need sensor data from various BAYs). Synchronization of a BAY to universal time is the manually driven procedure and should happen by the following steps:

1. BAY is synchronized with local time (working clock, cyclic time-scale of 1 second)
2. Slewing to universal time (Clock Boundary functionality)
3. When synchronized to the second of universal time, switch role to time aware system
4. All nodes (Station Bus & Process Bus) are synchronized by GM over universal time

(Comment: More details must be specified. Overall slow time can take about 1 hour.)

3.3 Requirements for Synchronizing Working Clock

1) Accuracy
- < 1µs over 64 hops@ industrial automation and energy automation
- < 100ns over 8 hops for special use cases like frequency converter

2) Interval for sync messages in a range of 15,625ms - 31,25ms

3) Usage of low cost oscillators in end stations and bridges

4) Frequency change (Δf / f) / Δt @ industry

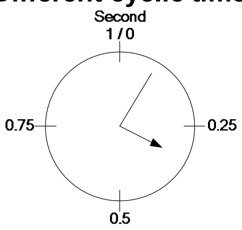
- short term 0 – 3 ppm within a second
- long term 0 – 5 ppm within a minute

5) Low latency for sync messages to minimize PLL reaction time (1ms / hop)

6) End-to-End GM rate measurement (to follow frequency GM change very quick)

7) Media independent and also long distance (e.g. production line)
- o ***Wired***
  - Long distance with fiber optic (multi mode, single mode)
  - Polymeric optical fiber
  - Copper
  - …
- o **Wireless**
  - .11
  - Bluetooth

8) Different cyclic time-scales for working clock



(e. g. 1 sec in energy automation)

9) Clock source for working clock is typically local time and not traceable to TAI (option)

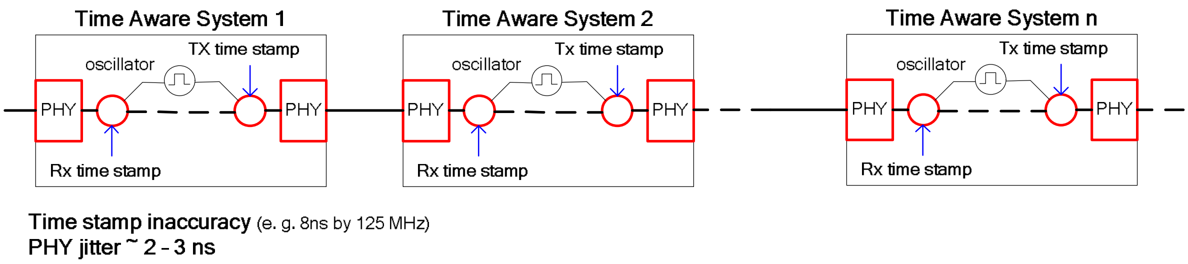
10) Guaranteed seamless working clock operation
- *Grandmaster change*
  - o guaranteed take over time < 200ms
  - o switch over time for slaves < 250ms
- *Path change*
  - o Guaranteed path reconfiguration time

=> Deterministic failure behavior for seamless working clock operation is required

11) High availability of working clock to handle single point of failure (robustness) and guaranteed take over time

Synchronization with multiple sync messages (forwarded over disjoint path) from one grand master to avoid offset jumps after sync tree reconfiguration (long daisy chains)

Impact of sync path change on accuracy



Time stamp accuracy causes an error in path delay measurement on each link which causes offset error. When receiving sync message, which is transmitted over one path, the offset error can not make visible. Only when receiving multiple sync messages from the grandmaster, which are forwarded over disjoint path, an offset error can make visible.

The effect can be measured when doing synchronization with PTP over large number of hope counts and long distances.

12) Working clock domains can be located anywhere in the network

13) GM of working clock domains can be located anywhere in a working clock domain

14) Each working clock GM capable device has mostly the same clock quality which fulfil the clock source quality requirements for working clock grand master

- The “active” GM has highest priority
- GM changes only triggered by failure and ***not*** by source clock quality
- Only a few numbers (typical 2) of GM capable within a working clock domain

15) Multiple (in)dependent working clock domains within one network

16) Maximum two hierarchical levels for working clock domains (see figure in chapter 3.2)

17) Manually driven merging to one working clock domain of two independent synchronized functionally cells without reconfiguration

18) While configuring a working clock domain synchronization of universal time shall not be disturbed

19) Topology independent

20) Security concept for working clock synchronization?

3.4 Clock Boundary Function and Alternate Timescale TLV

Clock Boundaries are required when a synchronized sub-domain is joined to a synchronized primary time domain and,

a) syntonization to universal time

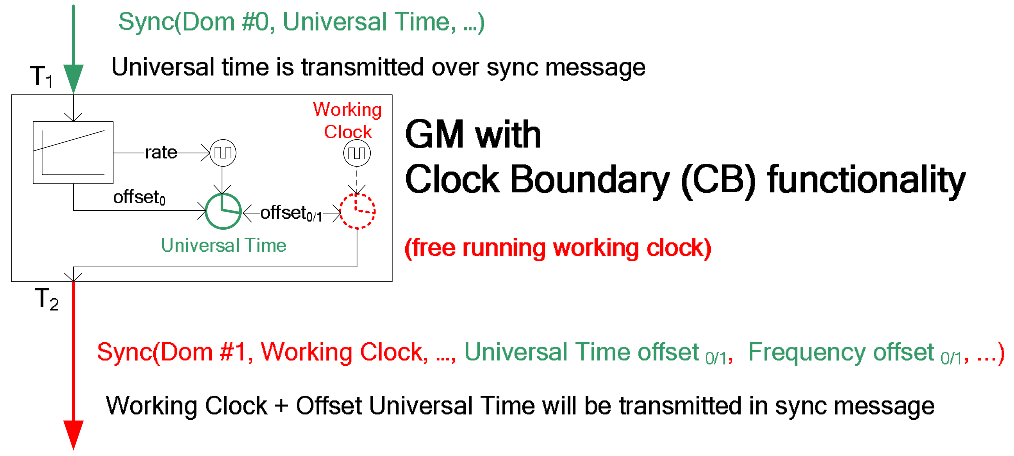
b) or synchronization to universal time (for e.g. short cycle tines-scale of 1 second inside the working clock domain) is required.

Time jumps within the working clock domain must be avoided. A mechanism for slewing to primary time domain is required. Only grandmaster capable nodes shall support clock boundary functionality.

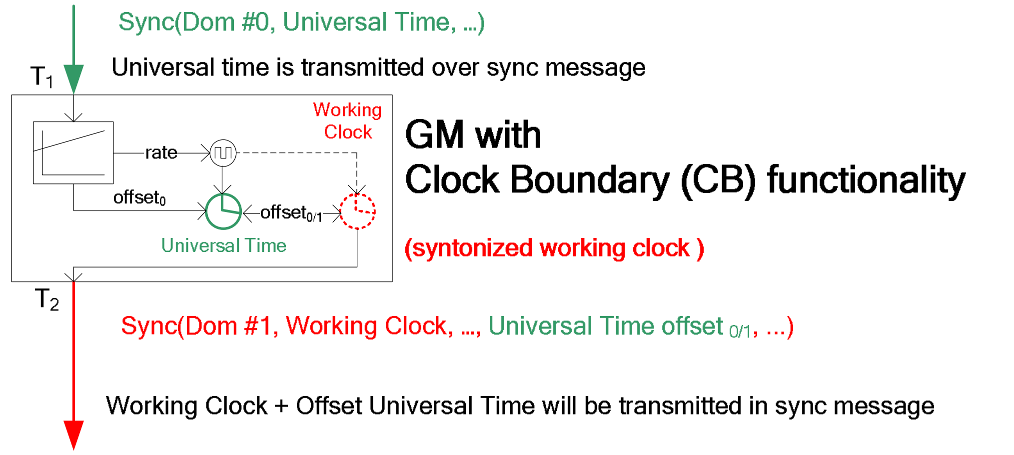
Also a mechanism, which supports manually driven joining operation to one common sync domain, is required.

The following three figures shows three different clock boundary types:

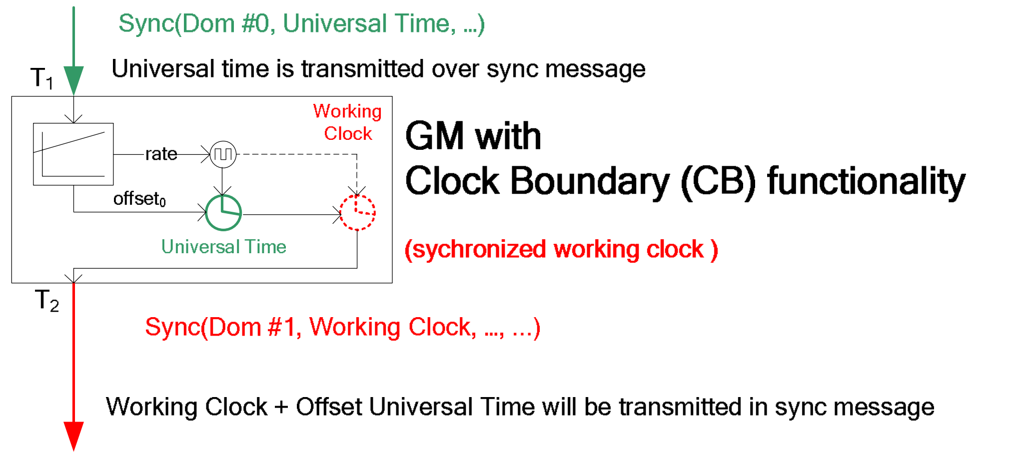
1. Clock boundary with free running working clock



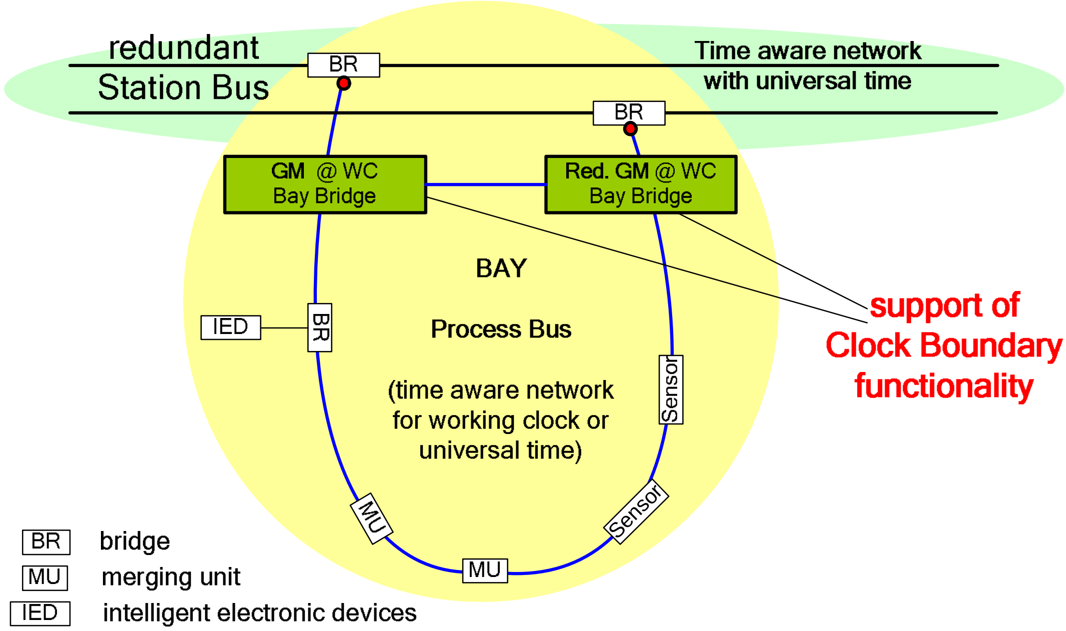
2. Clock boundary with syntonized working clock



3. Clock boundary with synchronized working clock



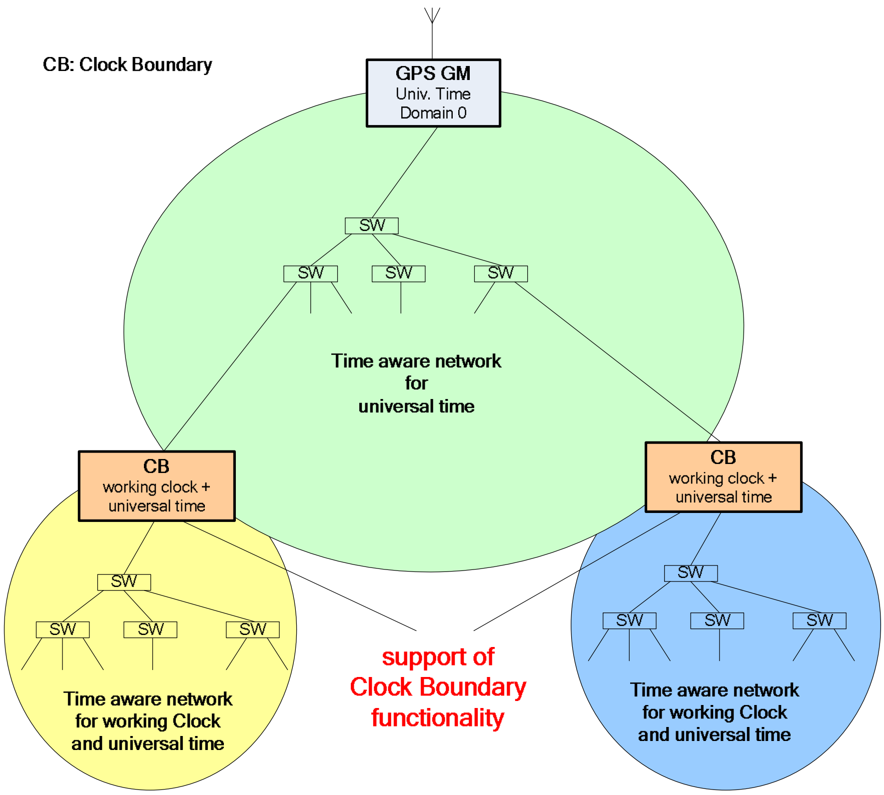
3.4.1 Energy Automation Use Case



Clock boundary function makes slewing to second of universal time possible.

3.4.2 Hierarchical Clock Use Cases

To guarantee accuracy within a working clock domain the working clock domain is separated by clock boundaries (CB's).



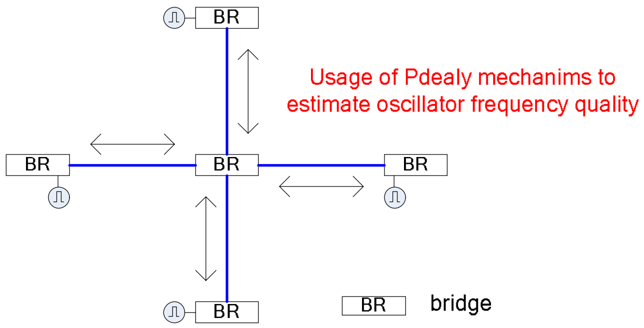
Time offset between universal time and working clock is announced by Alternate Timescale TLV.

4 Diagnostic for Clock Quality

For synchronization diagnostic a standardized algorithm to estimate frequency quality of time in bridges and end stations is required.

Without diagnostic information about oscillator frequency stability it is very difficult to locate frequency instable nodes.

To measure frequency stability, each node needs knowledge about its own frequency quality. The mechanism specified in IEEE802.1AS can be used to compare its own frequency quality with the neighbor’s frequency.



Furthermore, an algorithm is required to estimate the quality of synchronized time dependent on its own local oscillator quality and on the information of grandmaster quality which is provide by the synchronization protocol.

5 Overview of Concepts

5.1 Introduction

Sections 2 and 3 introduced the notion of a universal time for the entire network and, in addition, a local time for a small portion, or subnetwork[1] of the network. The universal time is typically the time distributed by GPS (or by another global navigation satellite system (GNSS)), i.e., it is traceable to TAI. The local time for the subnetwork is traceable to the working clock for that subnetwork. The reasons for distributing both the universal time and the subnetwork time are given in sections 2 and 3, respectively, along with requirements for the two time distributions.

In this section (i.e., in the following subsections), three approaches for distributing and maintaining both universal time and working-clock time are described. These approaches are: (a) reference clock model, (b) use of multiple domains, and (c) use of the alternate timescale TLV.

5.2 Reference Clock Model

This subsection describes the reference clock model approach for distributing working clock and universal time. This approach is very similar to the approach described in [1].

Figure 5-1 illustrates the entire network (union of grey and yellow areas) and a sub-network (yellow area). The orange-shaded PLC at the left is the grandmaster (GM) for the entire network and the sub-network. All the PLCs, DEVs, and bridges are time-aware systems. The time labeled *RefTime* is distributed via gPTP. This time may or may not be traceable to TAI; in either case, all the time-aware systems in the full network will know RefTime. Figure 5-1 also shows the working-clock time, which is labeled *PLCTime*. In general, there is a certain time and rate (i.e., frequency) offset between PLCTime and RefTime (and, in general, both the time and rate offsets change with time). If a time-aware system knows both RefTime and the current time and rate offset between RefTime and PLCTime (i.e., the values of these offsets at RefTime), it can compute PLCTime at the current time and at subsequent times (until new offset information is received, at which time it uses the new offset information and corresponding RefTime).

The time and rate offset information can be distributed in a new message. The information consists of RefTime and the corresponding time offset. The rate offset does not need to be distributed explicitly, because it can be calculated at each slave from successive messages. This calculation is analogous to the calculation of neighbor rate ratio in IEEE Std 802.1AS™ – 2011 (the calculation can be a simple ratio of RefTime and PLCTime differences, or various filtering or averaging schemes may be used). In Figure 5-1, this distribution of this information is shown generically via a new message termed the *TimeTP message*.

As indicated above, RefTime may or may not be traceable to TAI. In the discussion in Section 2, the universal time for the entire network is traceable to TAI and is usually distributed by GPS. However, it is possible that a TAI-traceable source of time may not always be available. In this case, RefTime could have an arbitrary epoch; in addition, it might have a rate offset relative to TAI. In the event that a TAI-traceable source of time is initially not available and later becomes available, it is desirable that the network time traceable to RefTime at each time-aware system not experience a large jump (and, also, not experience a backward jump). While it is possible to use filtering to limit the rate at which time adjustments are made, the durations of resulting transients could be very long if large adjustments are made when TAI becomes available. Therefore, we instead use RefTime as the time source for the GM, with the understanding that RefTime may be maintained locally at the GM and may not be traceable to TAI. However, since TAI is still needed for some applications, it is distributed in the same manner that the working clock is distributed, i.e., using the TimeTP message. The particular TimeTP message that distributes TAI is referred to as the TimeTP_1 message. The TimePT message that distributes the n^{th} working clocked is referred to as the TimeTP_n message.

[1] In this introduction, we use the generic terms *portion* and *subnetwork* rather than the terms *domain* or *subdomain* to avoid confusion with PTP domains. One method for maintaining the separate times for the entire network and the subnetworks is to use PTP domains; however this is not the only method. Both the use of PTP domains and other methods are described in the subsections that follow.

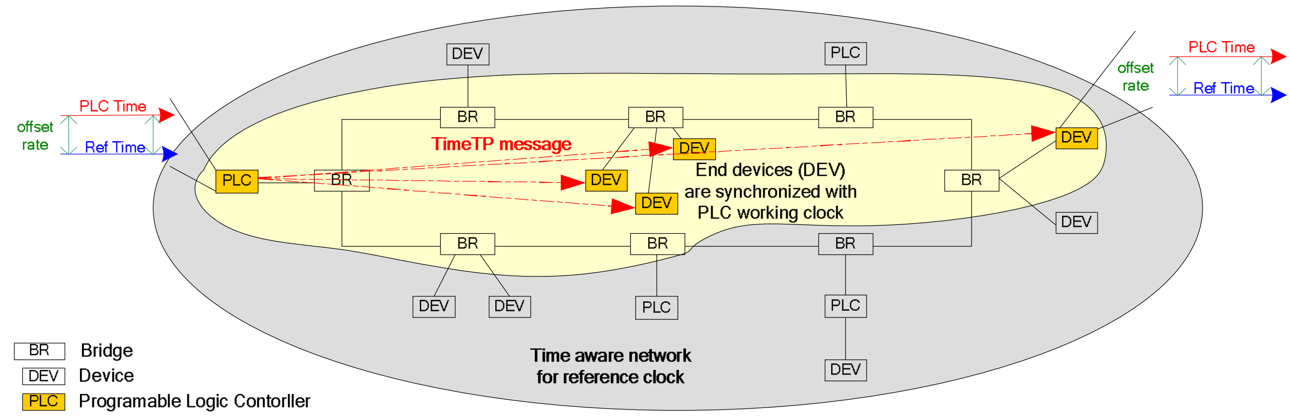


Figure 5-1. Illustration of distribution of universal and working-clock time in a sub-network.

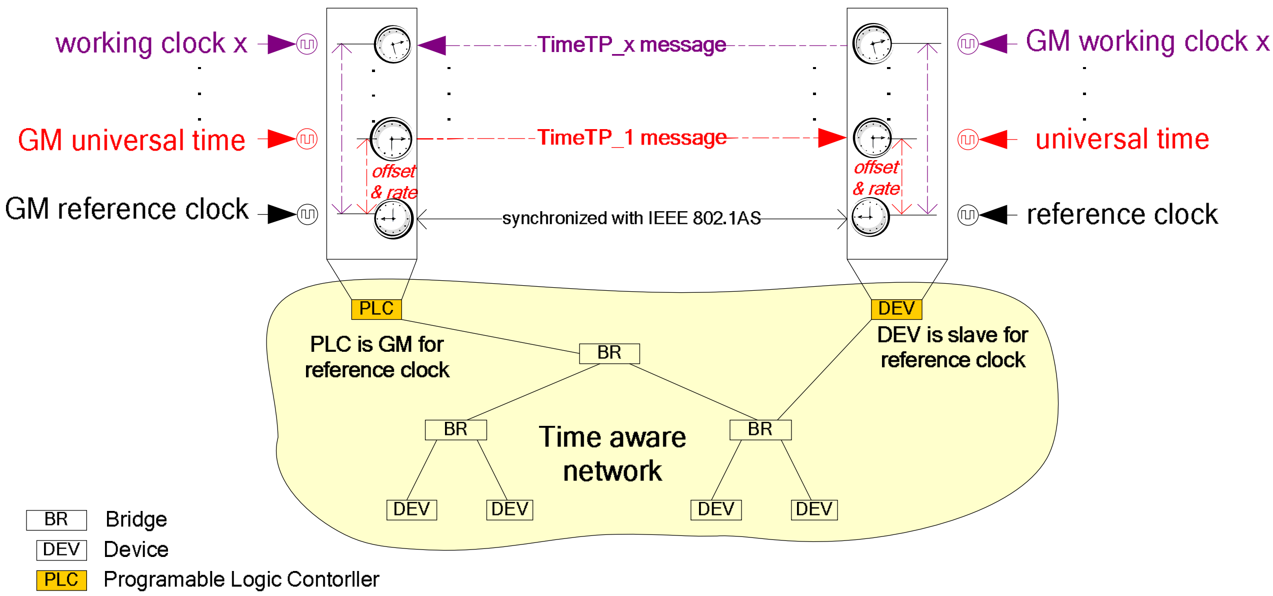


Figure 5-2. Illustration of relations among reference clock, universal time, working clock, and local device time.

Figure 5-2 illustrates the relations among the reference clock, universal time, and working clock. At each slave device (and at each time-aware bridge, though it is not shown) the time and rate offset between universal time and reference clock time is known via the information received in the Time_TP_1 message. In addition, the time and rate offset between working clock *n* time and reference clock time is known via the information received in the Time_TP_n message. The time and rate offset information is obviously different for different working clocks and for TAI. In addition, not every slave device or bridge will necessarily receive time and rate offset information for every working clock; a slave device or bridge will receive information only for the working clock whose time it needs (and possibly other working clocks whose information it must transport). However, all time-aware systems receive the time and rate offset information for TAI (i.e., the Time_TP_1 message) when it is available.

The TimeTP message contains at least the following information:

TimeTP_mesage(*n*, RefTime, CorrespondingTime)

where

n	= working clock number (i.e., subnetwork number); *n* = 1 for TAI
RefTime	= value of reference clock time
CorrespondingTime	= working clock *n* (or TAI) time that corresponds to RefTime (i.e., when the reference clock time is RefTime, the working clock *n* time (or TAI time) is CorrespondingTime.

Since the timeTP message is timestamped at the PLC and used at the end device, the path it takes from the PLC to the end device is not critical; it could, for example, follow the data spanning tree.

As indicated at the beginning of this subsection, the method described here is very similar (it is functionally equivalent) to the method based on IEEE Std 1722™ – 2011, described in [1]. The TimeTP message is analogous to the 1722 message; note that the 1722 message carries additional information not needed here, and the format of the time information might possibly be different.

One possible issue that could arise with this approach is that there is no guarantee that any two end devices of a subnetwork will be separated by a sufficiently small number of hops of the full network spanning tree that the more stringent working clock subnetwork accuracy requirements can be met, i.e.,< 1 μs over 64 hops and < 100 ns over 8 hops for the working clock (section 3.3, item 1) versus < 100 μs over 128 hops for industrial automation and < 1 μs over 16 hops for energy automation (i.e., power systems)(section 2.3, item 1). Even if we assume that the working clock requirements are always met for any two nodes separated by 8 hops, there is no guarantee that any two nodes of a working clock subnetwork will always be separated via the spanning tree by 8 or fewer hops. [Editor's note: this could be another area where multiple spanning trees and IS-IS might be useful; see the previous comment above.]

5.3 Multiple Synchronization Domains

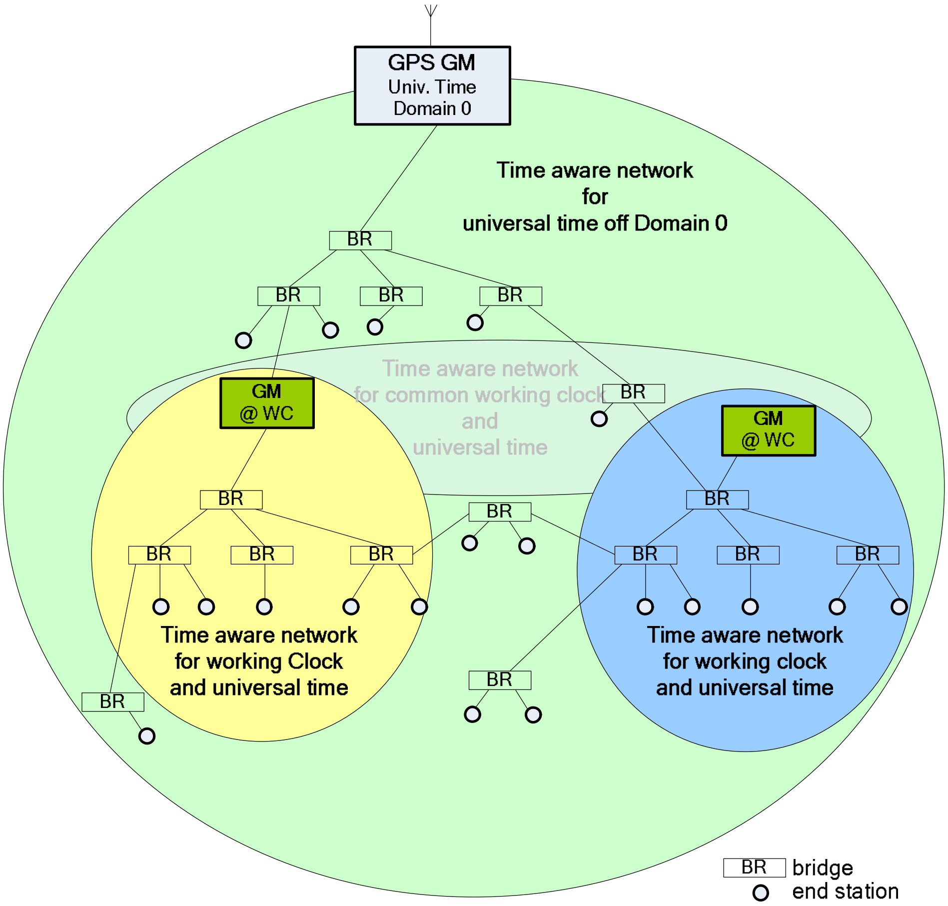


Figure 5-3. Illustration of network with multiple synchronization domains.

A second approach for distributing and maintaining universal time and working-clock time is to use multiple synchronization domains. While IEEE Std 802.1AS – 2011 specifies a single gPTP domain with domain number 0, IEEE Std 1588™ – 2008 allows multiple domains that can be distinguished by different domain numbers.

Domains are described in 7.1 of IEEE Std 1588 – 2008, where it is indicated that “a domain consists of one or more PTP devices communicating with each other as defined by the protocol” and that “a domain shall define the scope of PTP message communication, state, operations, data sets, and timescale.” It also indicates that “PTP devices may participate in multiple domains; however, unless otherwise specified in this standard, the operation of the protocol and the timescale in different domains is independent.” Based on this text, a boundary or ordinary clock (we omit transparent clocks from this discussion because they are not part of gPTP) can implement multiple domains; however, there is one instance of the PTP protocol stack for each domain. This does not mean that there must be a physically separate oscillator for each domain, nor that all the PTP code be replicated for each domain. However, it does mean that the time is maintained separately for each domain, as well as the portRoles, state machines, and execution of the best master clock algorithm (BMCA). However, in practice one exception to this is the operation of the peer delay mechanism at each port of a time-aware system. In theory, the peer delay mechanism would operate separately in each domain, and would measure average link delay in the time base of that domain. However, for the accuracies of the local node oscillators (no worse than ± 100 ppm) and the average link delays in question, the difference in the delay of a link measured in the time bases of different domain GMs is negligible (see note 2 at the top of p.117 of IEEE Std 802.1AS – 2011). Therefore, while a time-aware system will send and receive Sync, Follow_Up, Announce, and Signaling messages on each port for each domain, a single exchange of successive Pdelay messages will be performed by each port; each exchange will apply to all domains.

Domains are used to synchronize islands within a network with an alternate time scale like working clock. In practice, a time-aware system would implement at most two domains, one for the working clock and one for universal time. However, a time-aware system that was not part of a working clock domain would only need to implement one domain, for universal time. There are different reasons to synchronize nodes within a network with alternate time scale. One of them is high availability and high accuracy which can only be guaranteed within a small area.

Within a working clock domain bridges must handle at least one sync message per domain. If higher requirements on high availability for working clock must be covered, bridges must be able to handle also multiple sync messages (i.e., from redundant working clocks) within a working clock domain.

The domains that a time-aware system supports would be pre-configured. For example, if the universal time domain were Domain 0, then every time-aware system would support Domain 0. Each working clock domain would be given a separate domain number, and every time-aware system in working clock domain *n* would support Domain *n*. Then, domain boundaries are automatically established by this pre-configuration because each time aware system ignores messages whose domain number is not a domain that it supports. Note that Pdelay messages would all carry domain number of 0.

5.4 Alternate Time Scale TLV for Hierarchical Networks

[Editor's note: In IEEE 1588, the word "timescale" is always one word. However, in 802.1AS it is written as both "timescale" and "time-scale". This should be fixed in 802.1AS-Corr-1, and the one-word form should be used here as well.]

5.4.1 General

A third approach for distributing and maintaining universal time and working-clock time is to use the alternate time scale TLV. This approach is developed only for hierarchical networks. In theory, the approach could be extended to non-hierarchical networks; however, the mechanism for distributing the TLV could be very complex.

Figure 5-4 shows a hierarchical network, i.e., a network in which the time scales are hierarchically arranged. The GPS GM is the GM for the green sub-network, and provides universal time. In this example, the green sub-network does not form a working clock sub-network; however, if it did, the only difference is that the GPS GM would also serve as the working clock.
Two additional working clock sub-networks are shown, i.e., the yellow and blue areas. By *hierarchical network*, we mean that arrangement of the non-working-clock subnetwork (i.e., green area) and working-clock subnetworks forms a tree structure, with the non-working-clock subnetwork at the root of the tree. Additional working-clock subnetworks can be attached to the non-working-clock subnetwork or to the working-clock subnetworks, but with the constraint that the structure remains a tree, i.e., there are no loops and the path from the GM to any working-clock subnetwork is through a unique sequence of subnetworks.

A key aspect of this approach is that each sub-network forms a gPTP domain. However, except for the nodes at the boundary between two domains (each of these nodes is a *Clock Boundary*; the Clock Boundary concept is described shortly), each gPTP node implements a single domain and can use domain number 0. The domains are distinguished because they are physically separate, i.e., the Clock Boundary does not permit a gPTP message of one domain to enter another domain. The Clock Boundary is a new functionality, and is described below.

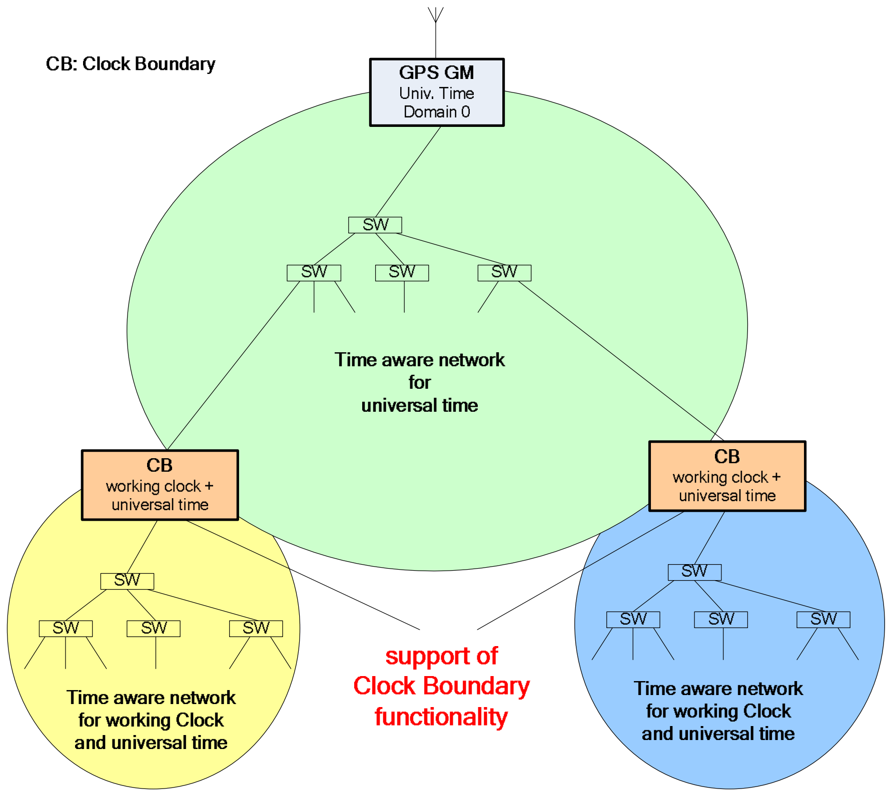


Figure 5-4. Illustration of Hierarchical Network.

5.4.2 Clock Boundary (CB) functionality

The Clock Boundary (CB) function is described in more detail in 3.4. It is a node that has ports in two different sub-networks. Since the arrangement of sub-networks is hierarchical, at least one of the ports in the sub-network closer to (i.e., within a fewer number of hops of) the universal time GM is a slave port. In addition, as least one of the ports in the other sub-network is a master port. The remaining ports may be master or passive. The CB is on the border between two sub-networks. The main differences between the CB and a regular time-aware system is that (a) the transport of time within the CB from the slave port to the master ports of the other sub-network (i.e., other sub-network relative to the sub-network of the master port) occurs outside of gPTP (b) the CB is the GM of the domain that is further (i.e., in terms of number of hops) from the universal time GM), i.e., it is the working clock GM, and (c) the time of the universal time GM is transported to the domain further from the universal time GM via the alternate timescale TLV.
[Editor’s note: In 3.4, three different possibilities are shown for the working clock in the CB, namely: (a) free-running working clock, (b) syntonized working clock, and (c) synchronized working clock. In the present section, only (a) is used. Should (b) and (c) also be possibilities?]

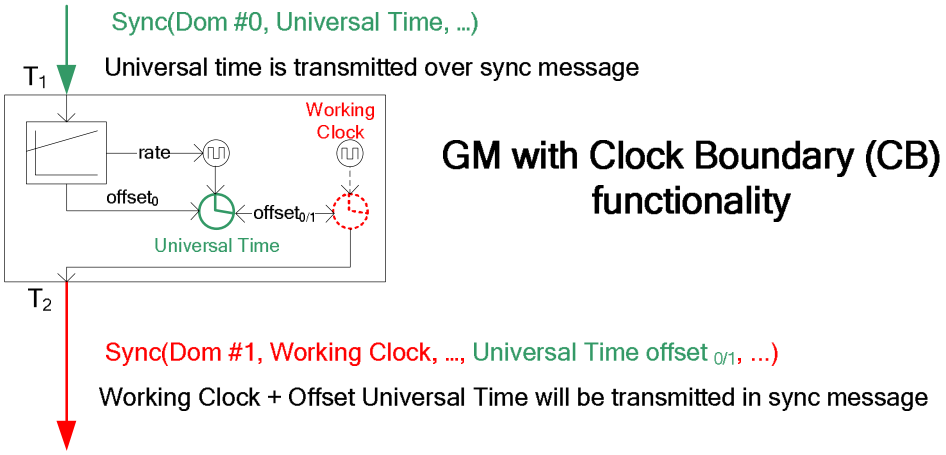


Figure 5-5. Illustration of Clock Boundary.

The CB is illustrated in Figure 5-5. It has at least two physical ports. One port is a slave port, and receives Sync and Follow_Up messages from the domain closer to the universal time GM. Another port is in the master state, and is the GM of the domain further from the universal time GM. In Figure 5-5, the working clock is free-running, and provides the GM function for the domain further from the universal time GM. In addition, a local clock function is slaved to the universal time GM via the Sync and Follow_Up messages received on the slave port. While Figure 5-5 shows the working clock and local clock as two separate functions, they need not be two separate oscillators. Since the model of IEEE Std 802.1AS – 2011 allows the local clock function to be a free-running oscillator, it is possible for the free-running working clock function to also serve as the local clock function. If there are additional master ports, or passive ports, in the domain further from the universal time GM, these ports lbehave as would be expected of ports on a GM. In addition, if there are additional ports in the domain closer to the universal time GM, these ports are either master or passive and behave as normal gPTP ports in this domain.

The CB measures and calculates time and rate offset between the universal time received from the domain closer to the universal time GM and the working clock function.This time and rate offset are transmitted via ALTERNATE TIME SCALE TLV[2] in the sync or follow up message over the ports in the domain further from the universal time GM that are in master state.

If the domain closer to the universal time GM is itself downstream from the domain of the universal time GM (e.g., if the clock boundary in question is between the blue working-clock domain of Figure 5-5 and a new domain (not shown) downstream of the blue domain), then the universal time at the clock boundary must be computed using the received working-clock time of the upstream domain (received via the Sync and Follow_Up messages) and the received ALTERNATE TIMESCALE TLV. The transmitted ALTERNATE TIMESCALE TLV always contains the time and rate offset between the working clock and received universal time, at the CB.

The TimeTP ALTERNATE TIMESCALE TLV contains at least the following information:

ALTERNATE TIMESCALE TLV (time offset of universal time relative to working clock time, rate offset of universal time frequency relative to working clock frequency.)

Since the ALTERNATE TIMESCALE TLV is attached to Sync or Follow_Up, it applies at working-time indicated by the originTimestamp (or preciseOriginTimestamp) and correctionField of the Sync or Follow_Up message.

[2] The ALTERNATE TIMESCALE TLV described here is a different TLV from the TLVs described in the alternate timescale feature of 16.3 of IEEE Std 1588 – 2008. In that feature, the alternate timescale is a defined timescale external to the GM timescale transported by PTP (e.g., timescale relative to a local timezone). In contrast, the alternate timescale here is a GM timescale that is transported by gPTP (namely, the working clock timescale).

As with the rate offset information computed for the reference clock approach, the rate offset can be computed in a manner analogous to the calculation of neighbor rate ratio in IEEE Std 802.1AS™ – 2011 (the calculation can be a simple ratio of universal time and working-clock time differences, or various filtering or averaging schemes may be used).

Certain applications do accept the topology constraint that the network must be hierarchical. The Alternate Time Scale TLV concept with clock boundaries is for them an acceptable solution.

6 Concept Benchmark based on new Requirements for synchronization

New requirements	**Reference clock model**	**Multiple sync domains**	**ATS TLV hierarchical**
Support multiple time scales (e.g. universal time + working clock)	- once common reference clock - time scale synchronization is not independent (Editor's note: more information is needed here; not clear why time scale synchronization is not independent) - data messages (IEEE 1722) or time transport messages are used	- multiple sync messages to support multiple time scales - only network components within regions with multiple time scales must support multiple sync messages	- Alternate Time Scale TLV
High availability region (e.g. - multiples sync messages over preconfigured path - hot standby grand master (i.e., multiple GMs, each sending Sync messages; all Sync messages are processed, with corresponding new Sync messages sent out, but only Sync messages from current active GM are used to produce the synchronized time supplied to the end application. - …)	- all components within a network have to support multiple sync messages	- only components within a high availability region have to support multiple sync messages	- only components within a high availability region have to support multiple sync messages (dependent on topology)
Support for high accuracy regions (e.g for industrial paramterter set - higher sync rate, - short residence time for sync messages - high time stamping accuracy within bridges - …)	- all components within a network have to support high accuracy	- only components within a high accuracy regions have to support multiple sync messages and have to cover high accuracy - a sync domain concept is required	- Clock Boundaries (CB role must be introduced) - can only used in a network with hierarchical time (i.e., the working clock and non-working clock subnetworks must form a tree, with the non-working clock subnetwork at the root. - a sync domain concept is required

7 Conclusions

The recent different solutions and derivates of IEEE 1588 all have their validity for their dedicated applications; they are optimized and fulfill their dedicated application needs.

In the IEEE 1588 version 3, parallel to the trend to convergent networks, time sync function has to become convergent too. A common stack of solutions should be defined, which covers all the necessary functions of industrial and other high precision time sync applications. This can prevent parallel solutions of IEEE 1588 on a convergent Ethernet network which is designed for a common use of multiple services like real-time application parallel to standard IP-traffic.

The combination of a system distributing TAI time with a working clock system described in the offers a comprehensive solution for various use cases of modern industrial networking. It provides a solution for the issue of combination of time stamp aware applications and cycle driven applications which eventually have to work in a combined manner.
The working clock can solve the problems combining parts of an application pre-commissioned to another part or extending an existing application with new parts.

Therefore the description of such a working clock solution can become an important function of the future Version 3 of IEEE 1588.

8 References

[1] Don Pannell, *What Makes Talkers & Listeners AVB Compliant*, Marvell presentation to 802.1 AVB TG, May, 2012.