Version: V8.2 2020-10-07

User Stories for IEC/IEEE 60802 system definition

Contributors:
Guenter Steindl
Marius-Petru Stanica
Martin Ostertag
Rene Hummen
Stephan Kehrer
Mark Hantel

Scope:

User stories used as a format to explain expectations of a user. User is in this case used in a very broad scope.

Examples:

The entity relying on synchronization is as user.

The person plugging a device into a network is a user.

…

CONTENTS

User Stories 3

Terms 3

Constraint 3

US1: Simple TSN Domain Startup 5

US2: Topology Updating / Topology change discovery 6

US2.1: Plugging an additional device 6

US2.2: Removing a Device from The Topology 8

US3: Combining two TSN Domains 9

US4: Splitting a TSN Domain 9

US5: Assigning TSN Domain Identifier 10

US5.1: Auto assignment of TSN Domain Identifier 10

US6: Media redundancy 11

US6.1: Switch over redundancy 12

US6.2: Seamless redundancy 12

US7: Application cycle 13

US8: Application (de-)coupling 13

US9: Timescales and Clocks 20

US9.1: GlobalTime (timescale TAI/PTP) 21

US9.1: WorkingClock (timescale ARB) 22

User Stories

Terms

ES	End Station
B	Bridge
ME	TSN Domain Management Engine
Device	Bridge or End Station
Plug&Produce	The application of an End Station which is plugged to a TSN domain can without additional user or engineering tool action request, establish and use (real-time) streams for communication.
Power on	"After power on" is a state in which any application would be able to run.

Constraint

The following is an unsorted list of constraints. The order is incidental and not intended to convey meaning.

Application

No application actions are shown, only end stations and bridge related behavior

Communication before ME applies policy[MS1]

The behavior depends on the history.[MS2]

A) Device is part of the TSN Domain, previous ME configuration is stored
All configured traffic classes are available and can be used

B) Device is part of a different TSN Domain
TSN Domain identifier check shows Domain Boundary,
Only limited traffic classes are available

C) Device is in "out of the box" state[MS3]
TSN Domain identifier check shows Domain Boundary,
Only limited traffic classes are available

Network configuration policy – in "ME applies policy"

Parameter set used by the ME to configure the Ethernet portion of bridges, bridged end stations and end stations. This should be done using Netconf or SNMP together with the needed YANGs and MIBs. The policy deployed by a ME may be configured or selected, using e.g. vendor specific means, by the responsible person for the TSN Domain.[MS4]

Security

Security, (e.g. authentication, encryption) is assumed to be part of the overall solution.

Synchronization

Synchronization is assumed to be part of the solution but ignored to simplify the user story discussions.

TSN Domain Identifier

The TSN Domain Identifier is expected to be unique in space and time to prevent accidental unintended combination of TSN domains (e.g. different feature sets and configurations may be active in these domains).

TSN Domain Management Engine

The following user stories assume at least three configuration models.

A) Complete offline engineered

- Topology
- Network and End Station[MS6]
- Streams

Offline is equal to Online!
Topology discovery and verification
Bridge and End Station configuration
Stream setup in Bridges and End Stations

B) Partial offline engineered

- Partial topology
- Network and End Station
- Partial streams[MS7]

Core portion of machine[MS8], offline is equal to Online. Machine variants define additional bridges and end stations.[MS9]
Topology discovery and (for the core) verification
Bridge and End Station configuration
Stream setup in Bridges and End Stations (for the core portion)

C) Network and End Station only[MS10]

- Network and End Station

Topology discovery
Bridge and End Station configuration

Device pre-configuration situations

If a device supports plug & produce it must support the minimal set of features required to be managed by an ME.

60802 devices must implement features to enable the ME configuration step that allows plug & produce[MS11].

Device Options

1. The device is shipped out of the box with all plug & produce features enabled.

2. The device comes out of the box unable to be used as plug & produce (e.g. LLDP is disabled) but is harmless to the network and needs to be configured in-place before being usable by the ME.

3. The out-of-the-box configuration is harmful (e.g. IP address reuse, loops with STP disabled) to the network and needs to be configured stand-alone before being introduced to the network.

Examples:

- A. End-station non-configured, out-of-the-box
    - The end station lacks whatever configuration (network-related or not)
    - It comes with a producer MAC address
    - Some examples of features that may be present which require configuration to allow an ME to configure:
        - The end-station has a default IP address
        - If there is a bridge embedded with the end station, then
            - No spanning tree activated
            - No DHCP activated
            - LLDP and SNMP/Netconf agents deactivated
            - No L3 routing active
        - Reachable over SSH over the default IP address, alternatively on another type of connection (e.g. USB, serial)

- B. Basic network configured end-station (OEM, system integrator, other)
    - The end station bears a requested MAC address/default MAC address
    - The end station bears a project-required fixed IP address / has its DHCP activated
    - The end station has its LLDP active, also SNMP/Netconf
    - If a bridge is present too
        - The bridge has the default STP activated

- C. Advanced network interface configured end station
    - The end station bears a requested MAC address/default MAC address
    - The end station bears a project-required fixed IP address / has its DHCP activated
    - The end station has its LLDP active, also SNMP/Netconf
    - The end station has a given TSN domain ID configured
    - The end station has its time synch protocol activated and pre-configured as required by a project
    - If a bridge is present too
        - The bridge has the default STP activated

US1: Simple TSN Domain Startup

Costumer creates a TSN Domain by configuring the ME with a TSN Domain identification and a network policy.

As a next step it creates a network out of four entities:

- 1 ME
- 1 Bridge
- 2 End Stations

as shown in Figure 1.

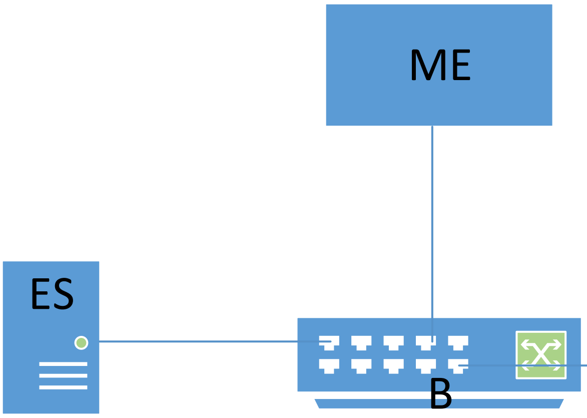


Figure 1: Simple TSN Domain

Expected behavior[MS12]

The ME discovers the connected network. For any discovered device assigned to its TSN Domain the Ethernet portion will be configured according to the ME stored policy.

Thus, all devices of the TSN Domain will be configured according to the ME network policy.

Any unused port will be configured as “TSN Domain boundary”.

US2: Topology Updating / Topology change discovery

A ME[RB13] needs an up-to-date topology including all bridges and end stations of a TSN Domain.

Expected Behavior

A ME can on a periodic basis walk its’ TSN domain to ensure the stored topology is still valid.[RB14]

US2.1: Plugging an additional device

Customer plugs another bridge and/or end station[MS15].[RB16]

When a user plugs a new device into a Time Sensitive Network domain, the user needs to be able to configure the device through the ME, and thus the ME needs to know it exists.

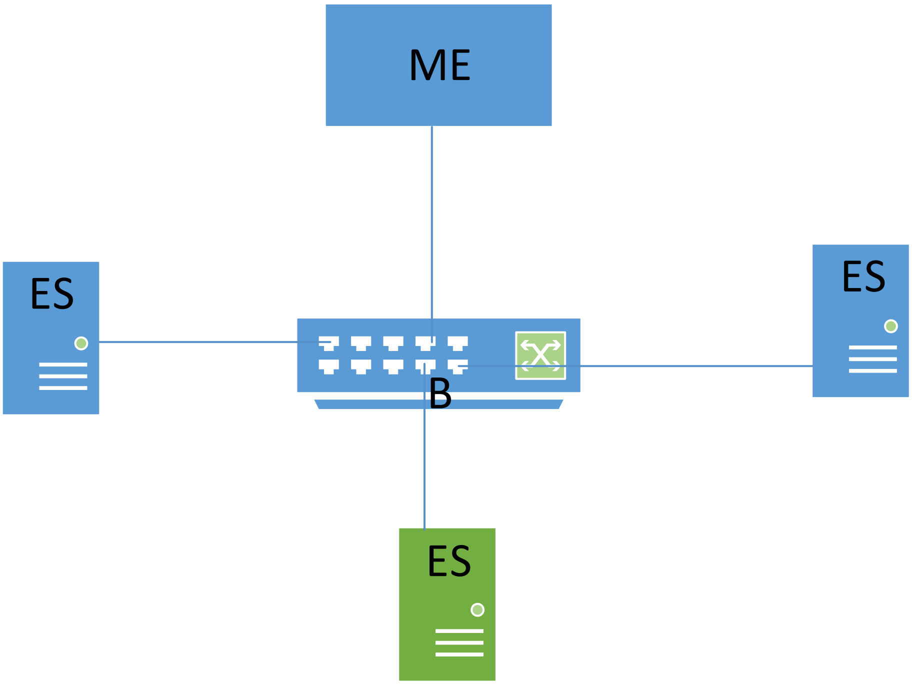


Figure 2: Simple TSN Domain – plug an ES

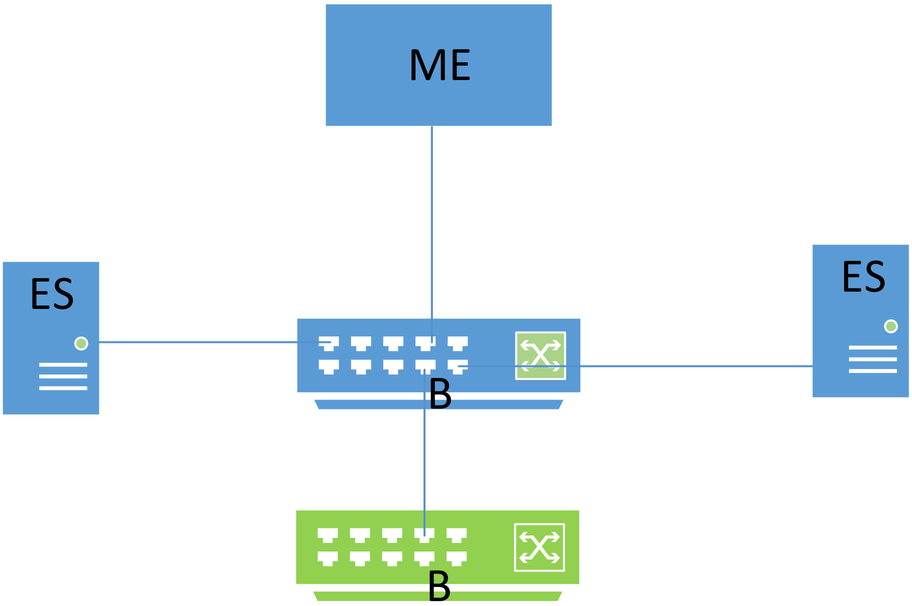


Figure 3: Simple TSN Domain – plug a Bridge

Expected behavior

ME discovers the additional connected device using Topology Discovery, checks (by using the TSN Domain identification) whether it belongs to this TSN Domain.

LLDP is implemented on every device that is 60802 conformant. The device plugged in will advertise its presence and identifiers to adjacent nodes. Identifiers must include MAC address, IP address, and TSN Domain Identifier.

If it belongs to the TSN Domain, then the ME configure the Ethernet portion according to the ME stored policy.

If not, no action.

Adjacent nodes can provide information about the new device to the ME. Adjacent nodes can store information about the new device in their memory to be read by a ME. Time constraints (60802 Use Case 20) may require the adjacent node to provide information to the ME.

US2.2: Removing[RB17] a Device from The Topology[MS18]

When a user removes a device from a TSN domain, the user needs to see this device removed from the ME.

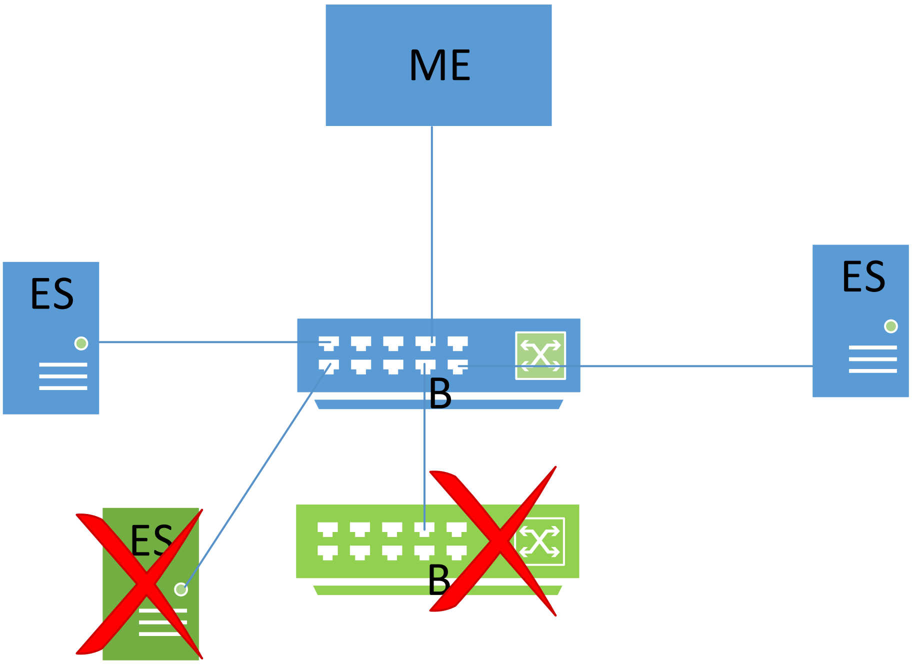


Figure 4: Simple TSN Domain – remove a device

Expected Behavior

The removed device will no longer be a part of the discovered topology.

Adjacent nodes can provide information about the removed device to the ME. Adjacent nodes can store information about the removed device in their memory to be read by a ME. Time constraints (60802 Use Case 20) may require the adjacent node to provide information to the ME.

US3: Combining two TSN Domains[MS19]

A user introduces a new physical link that joins two TSN domains.

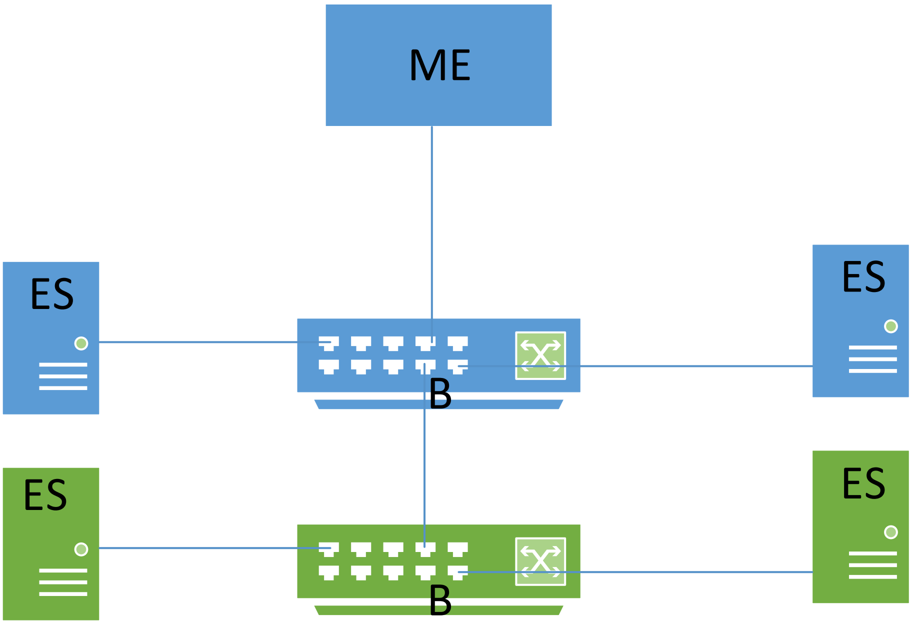


Figure 5: Simple TSN Domain – combine previously splitted TSN Domain

Expected Behavior

If each TSN Domain has the same identifier because they were previously combined or engineered to be combined by sharing compatible TSN mechanisms and identifier, the two domains will be joined into one. If each TSN domain has an independent ME, one ME will be selected. If the identifiers are different TSN Interdomain communications may be established by the Management Entities.[MS20]

US4: Splitting a TSN Domain

A user removes a physical link that creates a single TSN domain.

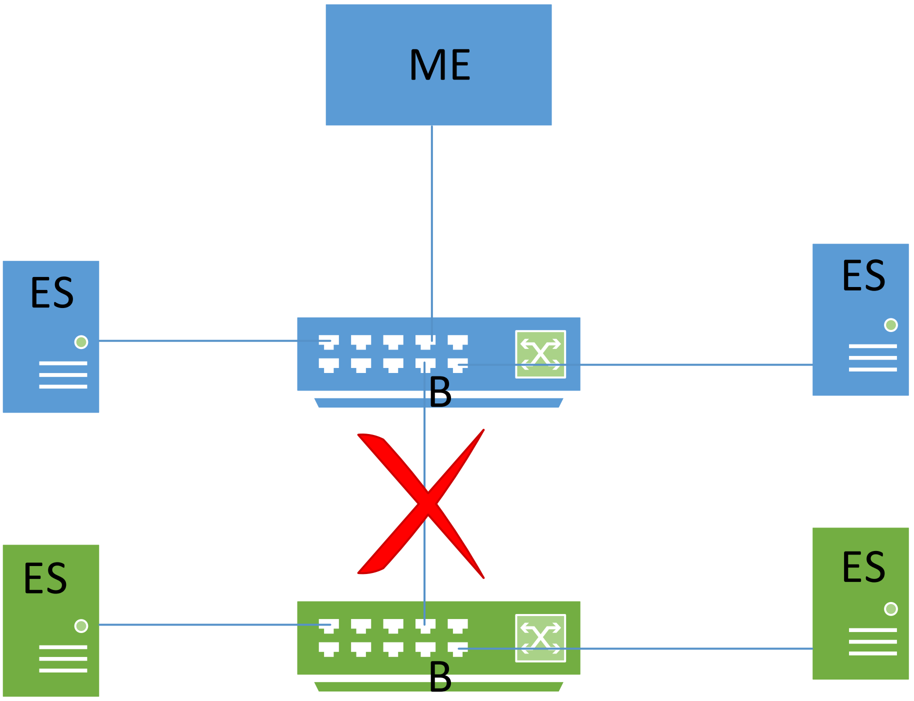


Figure 6: Simple TSN Domain –split TSN Domain

Expected Behavior[MS21]

Each network will continue to operate as two separate TSN domains and are managed by separate Management Entities. If one of the resulting domains does not have a ME, no functions that require an ME will be available. Each will automatically maintain the same TSN Domain Identifier[MS22].

US5: Assigning TSN Domain Identifier

Whether a device belongs to a TSN Domain is identified by checking its TSN Domain Identifier. This identifier is expected to be available as LLDP TLV / MIB object defined in IEEE802.1Q scope.[MS23]

Expected Behavior

When the user assigns the application driven identification, additionally the to be connected TSN Domain needs to be specified[RB24].[MS25]

Thus, the ME can identify whether this device is in its responsibility or not.

US5.1: Auto assignment of TSN Domain Identifier

Additional to the TSN Domain Identifier, different means may be used to identify whether a connected device belongs to the TSN Domain or not.[MS26]

Expected Behavior

If the A) configuration model of the ME is used, the device and its position in topology can be used to assign all addressing information to the device. This includes the TSN Domain Identifier.[MS27]

US6: Media redundancy

Media redundancy is used to ensure availability of communication.

Simple topologies e.g. rings are often used to fulfill the requirements for media redundancy. More complex requirements lead to more complex topologies e.g. coupled rings allowing multiple concurrent faults.

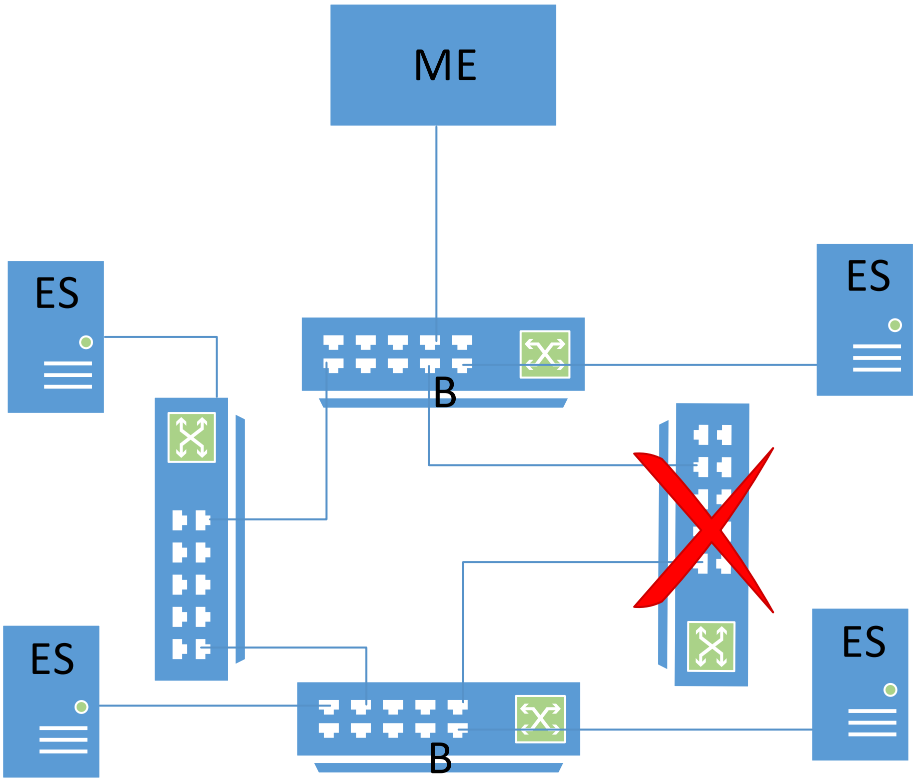


Figure 7: Simple ring topology – one fault

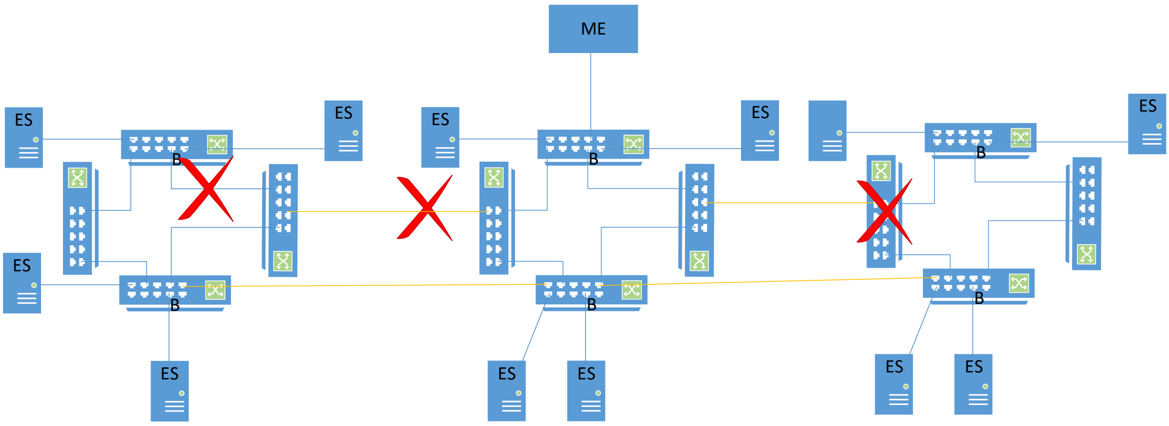


Figure 8: Coupled rings topology – multiple faults

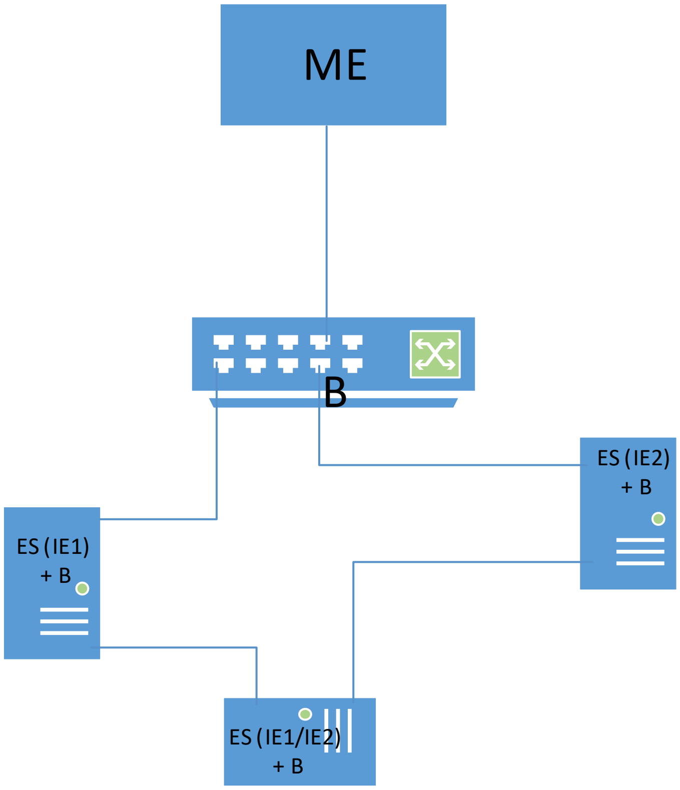


Figure 9: Ring composed of end stations and embedded bridges from various vendors and supporting various Industrial Ethernet (IEx) frameworks

Expected Behavior

One or more faults, e.g. wire break or disconnected bridge, do not stop the production until a repair happens.[MS28]

US6.1: Switch over redundancy

Simple topologies (e.g. rings) are often used to fulfill the requirements for resiliency.

These requirements apply to simple rings and to coupled rings.

Expected Behavior

As soon as a fault happened[RB29], the system redirects the traffic to follow the substitution path. Depending on the application requirements the time which is needed to redirect the traffic may or may lead[RB30] to a communication disturbance.

Spanning-tree protocols (e.g. RSTP, MSTP) can be used as a resiliency protocol.

If traffic engineered streams are used, the ME may need to “reroute” them according the new topology.

US6.2: Seamless redundancy

Traffic engineering two most disjunct paths from Talker to Listener allow[RB31] seamless redundancy with zero switch over time.

These requirements apply to simple rings and to coupled rings.

Seamless redundancy based on FRER may be solved by end-station FRER or bridge FRER at the end-station ports; it needs to be solved by bridge FRER at the ring coupling ports.

Expected Behavior

Seamless redundancy shall be available for simple and coupled rings and support[RB32] one fault per ring.

US7: Application cycle

Synchronized applications need the same understanding of their application cycle. This understanding shall be independent from the time when the device is powered up or connected to the TSN Domain.

A common understanding of an application cycle is built out of the working clock in the following manner:

If you divide the integer working clock value by the integer “length of application cycle”, and the remainder is zero, the application cycle starts.

If a defined application cycle is addressed, the resulting quotient may be used to create a common understanding of future or past application cycles.

Expected Behavior

Application cycles are synchronized, if needed, independent from the time a device is connected to the network.

Base is the Working Clock.

US8: Application (de-)coupling

Expected Behavior

Moving from parallel to sequential control raised the problem of the data basis for the program execution together with its execution speed.

That lead to the implementation of a so-called Process Image, which was in the beginning only a few bytes in size.

Basic timing model – fetching inputs, writing outputs[RB33], compute, repeat

Increasing the available memory, the available compute, the amount of connected IOs, integrating motion and of remote IOs just extended this basic model. Always ensuring that the customer applications are running unchanged!

Nowadays, the process image has a size from less than 1k Bytes up to 1M Byte.

This amount of data, provided by up to 1024 communication partners, takes some time to be collected. Applications require update times of their portion of the process image from 25µs/31,25µs up to 1s.

Thus, the update of the Process Image and the Network Access interact to support the requirements for timeliness from the applications. Compute contains the background task, which creates the trigger for a snapshot of the process image from the intermediate process image.

Additionally, many time triggered tasks, either by working clock or global time are executed concurrently.

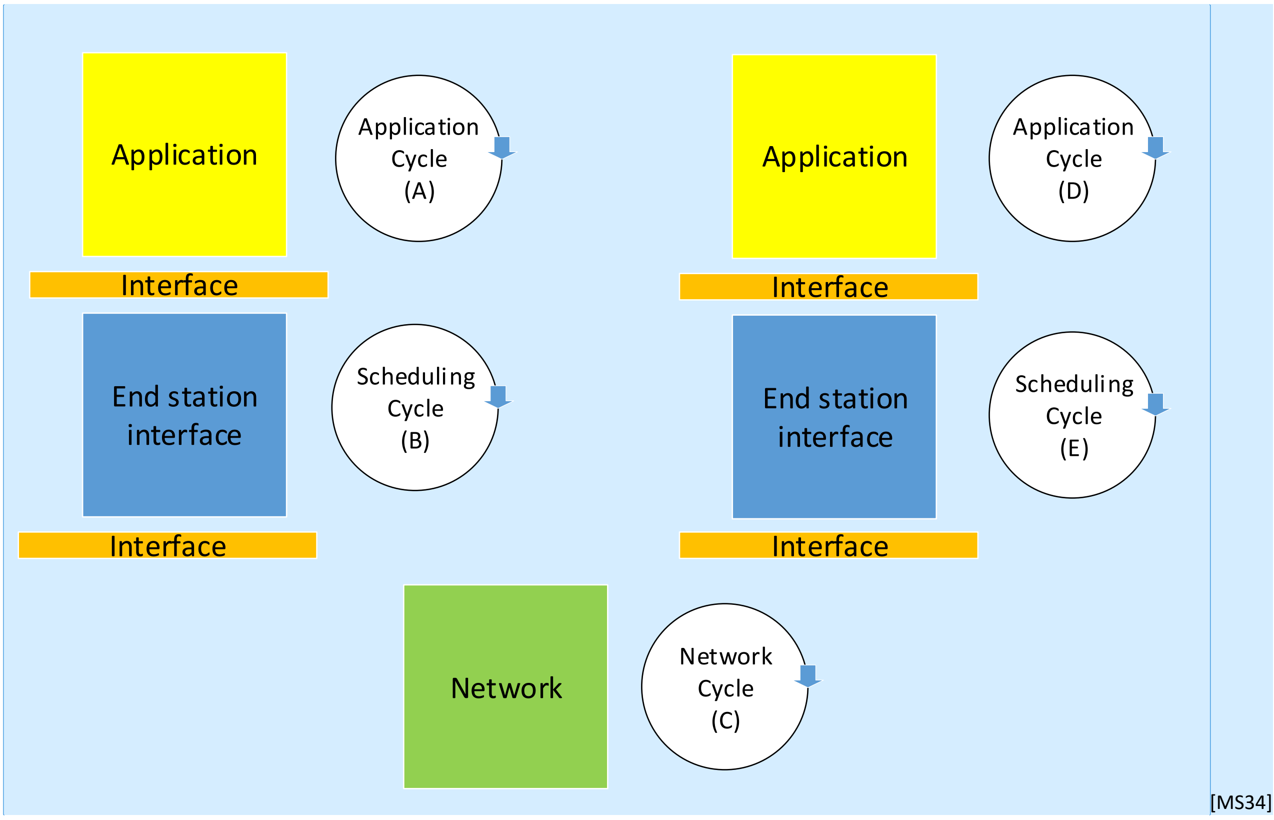


Figure 10: Principles

Figure 10 shows the connection between two end stations.

Principle use cases

Automation systems do support, depending on their needs, many different models. The following five are very often seen in the automation arena.

Case "unsynchronized"
All entities do have their own cycle.

Latency variation are the result of this "unsynchronized" cycles.

Case "synchronized application cycles"
All entities do have their own cycle, but the application cycles (A) and (D) are synchronized to each other.

Latency variation are the result of the "unsynchronized" cycles of this model.

Case "synchronized scheduling cycles"
All entities do have their own cycle, but the scheduling cycles (B) and (E) are synchronized to each other.

Latency variation are the result of the “unsynchronized” cycles of this model.

Case “synchronized application and scheduling cycles”

All entities do have their own cycle, but the scheduling cycles (B) and (E), and the application cycles (A) and (D) are synchronized to each other.

Latency variation are the result of the “unsynchronized” cycles of this model.

Case “synchronized application, scheduling cycles and network cycle”

All entities do have their own cycle, but they are synchronized to each other.

Latency variation are the result of the “synchronized” cycles of this model.

Application

General

The following application models are based on a decoupling between communication and application based on a shared memory concept used as interface for the exchange of data.

The application space thus looks like Figure 11.

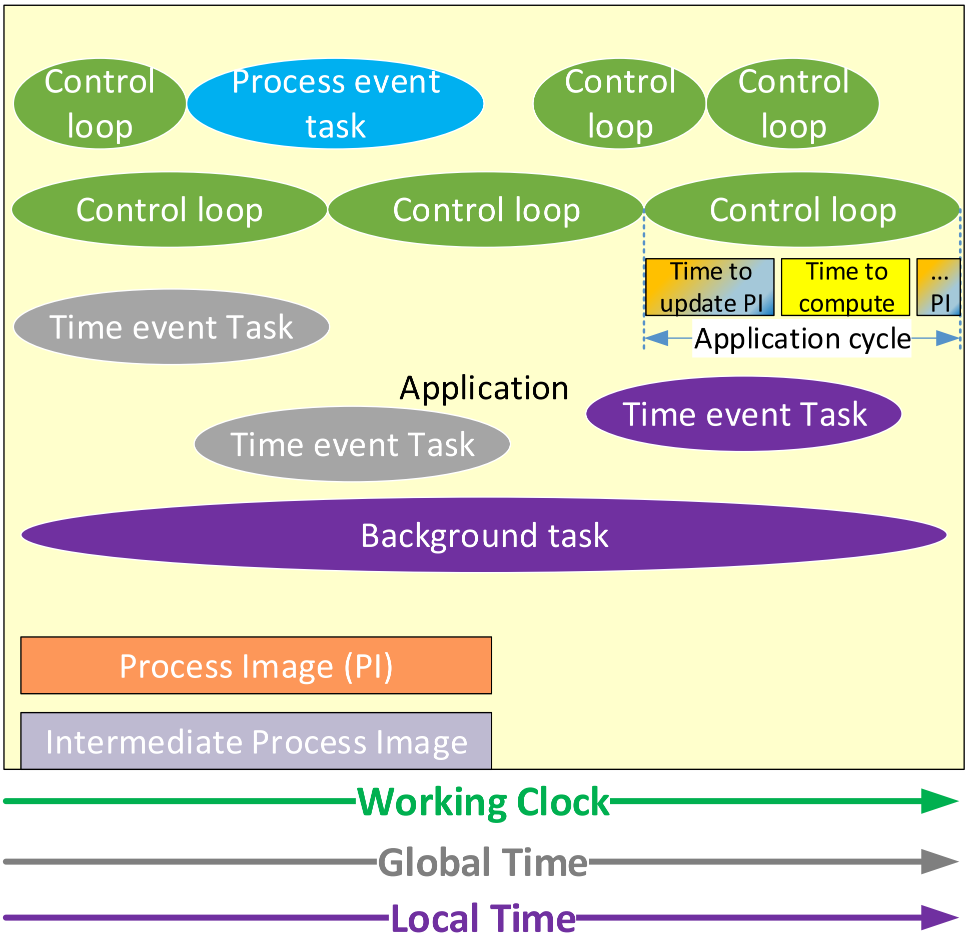


Figure 11: Application area

Behavior of Process Image

Application cycle

Most devices do have an application cycle which updates the Process Image bound to local understanding of time (Local time).

Updating the Process Image means, that the content of the intermediate Process Image is copied (actions for coherence and consistence are done) into Process Image (Inputs) and vice versa (Outputs).

Background task

Just working on the actual content of the Process Image.

Time event task

Time event tasks may either be triggered by Local Time (e.g. start every 10ms) or Global Time (e.g. at the top of the hour). They may create a local copy of the Process Image portion, if they need to ensure that they work on data from one Process Image update.

Process event task

Process event tasks are triggered by process events (e.g. diagnosis alarm). These are normally working on the actual content of the Process Image.

Control loop task

Control loop tasks are triggered by Working Clock (otherwise they are implemented as Time event tasks). They may create a local copy of the Process Image portion, if they need to ensure that they work on data from one Process Image update.

Communication

General

The following communication models are based on a decoupling between communication and application based on a shared memory concept used as interface for the exchange of data.

The communication space thus looks like Figure 12.

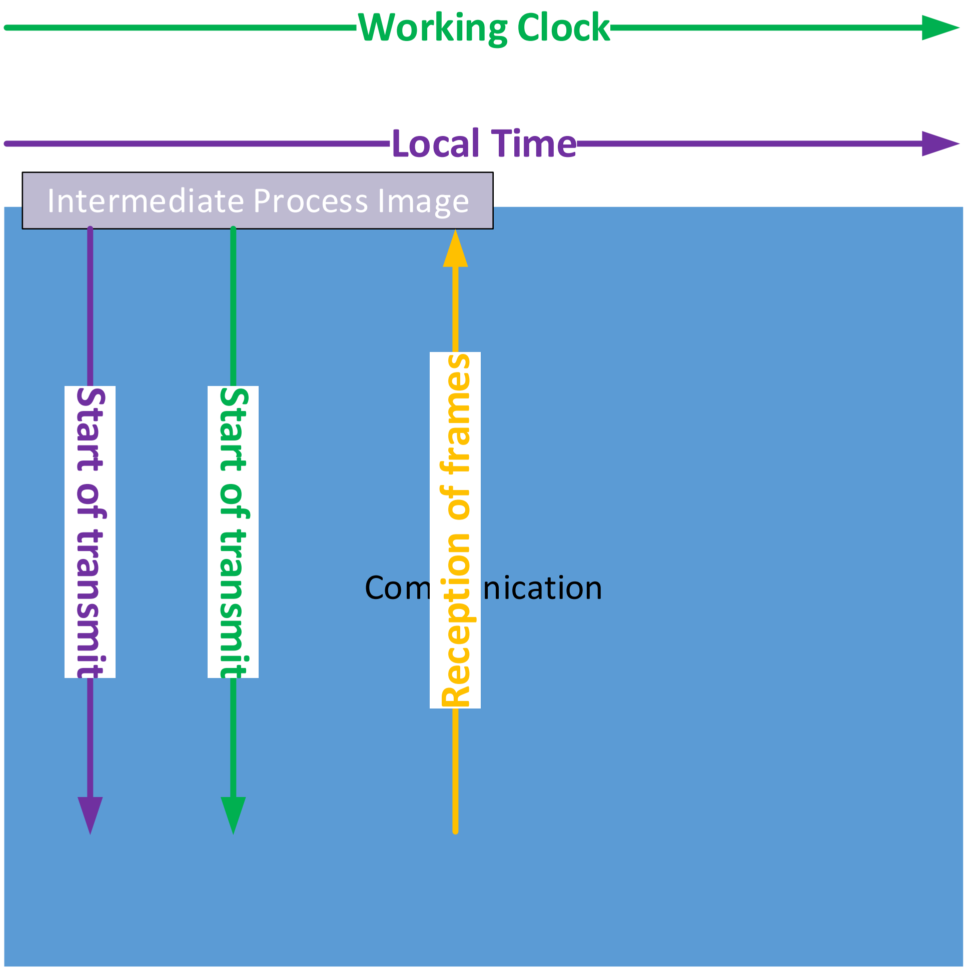


Figure 12: Communication area

Periodic transmission of frames is either driven by Local Time or Working Clock. Source and sink of data are the intermediate Process Image.

Transmit (Local Time triggered)
Periodic transmission is driven by Local Time. An alignment with the Application is not intended.

Transmit (Working Clock triggered)
Periodic transmission is driven by Working Clock. An alignment with the Application is possible based on events driven by Working Clock timescale.

Reception
Reception of frames is independent from the timescale used for transmitting.

Alignment between Application and Communication

Alignment between Application and Communication is done by relying on Working Clock as shown in Figure 13.

Deadline is used as “latest point in time” when the data for the control loop need to be available in intermediate Process Image.

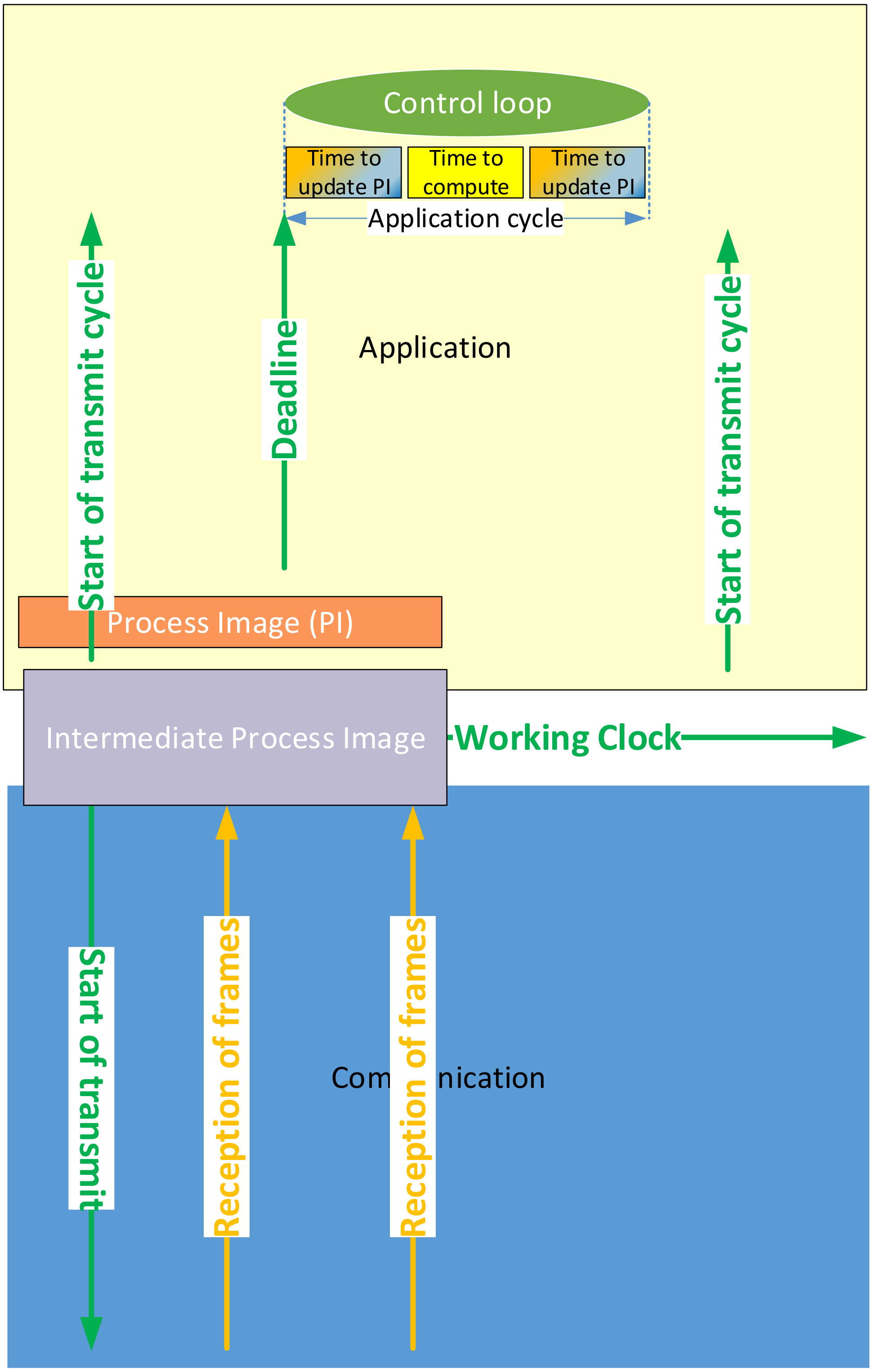


Figure 13: Alignment model

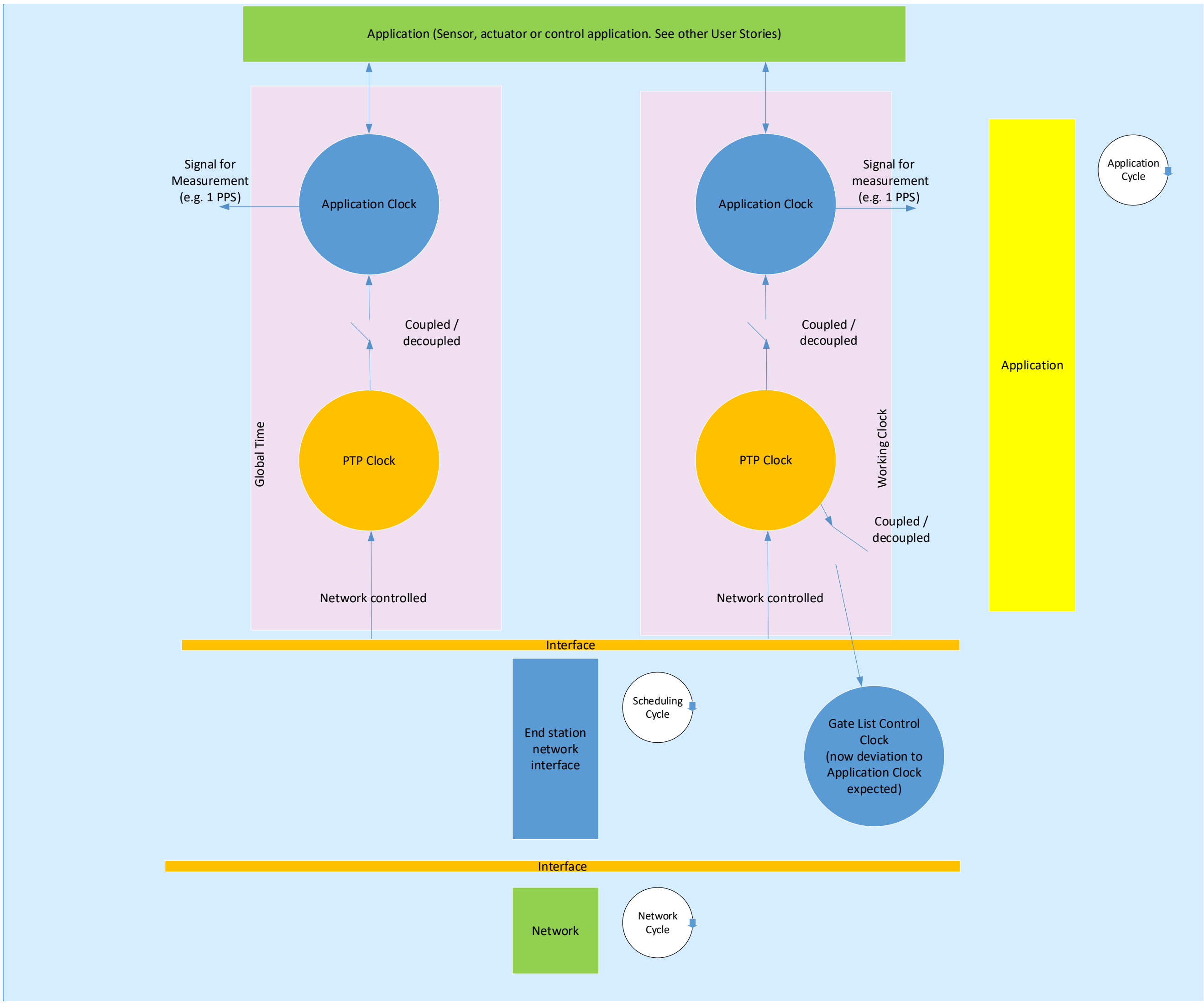
US9: Timescales and [SG(FTAE35]Clocks

Expected Behavior

GlobalTime and WorkingClock are needed in many automation devices.

Automation applications need to work independent from connected communication interfaces. Thus, use cases in which the synchronization is lost or reestablished shall be covered. This is often done by implementing two instances of a Clock for one PTP end instance. shows this clock usage model.

Deviation of the Application Clocks, at the GM and the PTP end instance, are important and thus, checked during certification, for the industrial use cases. The deviation of intermediate Clocks e.g. the PTP Clock in Figure 14, are only of interest as influence to the quality of the Application Clock and the Gate Control List Clock.



[MS36][MS37]

Figure 14: Clock usage model

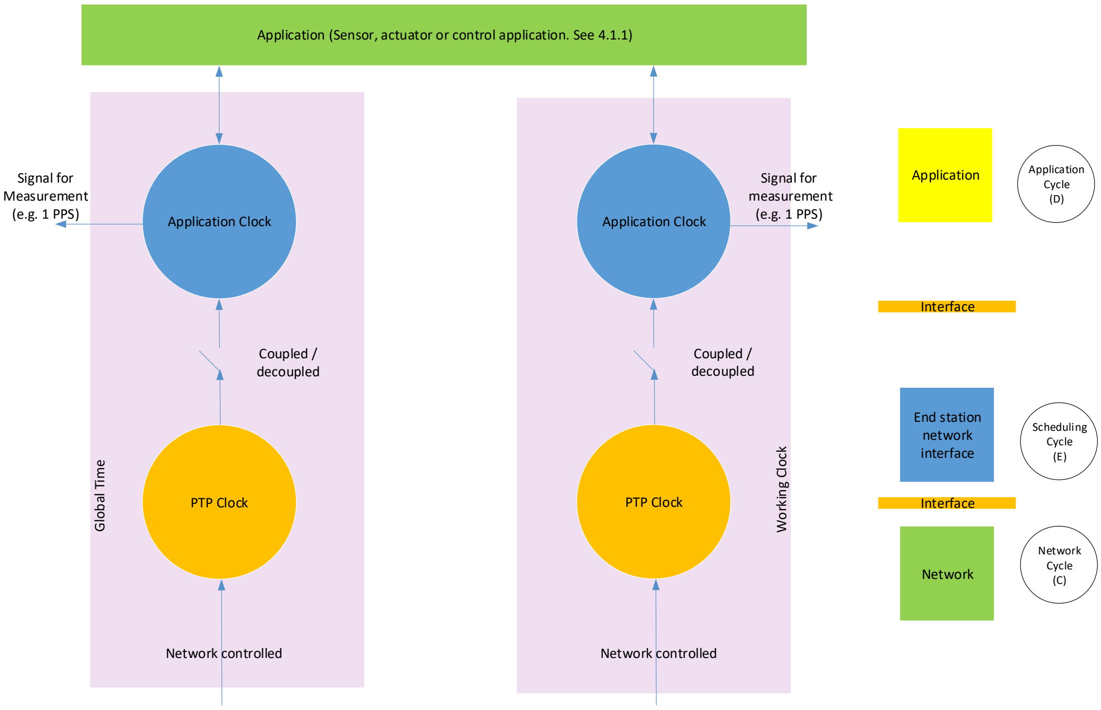


Figure 15a: Alternative clock usage model

State information is provided together with the time information to the application.

US9.1: GlobalTime (timescale TAI/PTP)

Expected Behavior

GlobalTime is if supported by the device always available independent from an existing / connected GM.

- Monotonically increasing time, where monotonic means that a tick can mean +0, +1 or + >1. Both, how often your able to do “+0” or “+ >1” is limited by the allowed acceleration/deceleration per 1 ms.
- The maximum acceleration/deceleration is also regulated and is in the range of e.g. +/-1000 ppm (that’s equal to 1 µs for an interval of 1 ms) for the PTP end instance offset correction
- (“value jump in case of set value” is allowed, but must be documented at every PTP end instance to comply with application needs or legal rules (e.g. Power Generation))
- The threshold/algorithm for the PTP end instance’s clock to declare inSync is within +/-100 µs deviation to the GM
- 64/100 Hops must comply with the +/-100's deviation window to the GM
- GM and PTP relay and PTP end station timer have +/-50 ppm oscillators
- The frequency of these oscillators may change due to external influences e.g. df/dt of 3 ppm
- Connection to GPS or similar possible/supported

Any violation of these requirements makes GlobalTime either useless OR disrupts machines and equipment in the worst case.

US9.1: WorkingClock (timescale ARB)

Expected Behavior

WorkingClock is if supported by the device always available independent from an existing / connected GM.

- Monotonically increasing time, where monotonic means that a tick can mean +0, +1 or + >1. Both, how often your able to do “+0” or “+ >1” is limited by the allowed acceleration/deceleration per 1 ms.
- The maximum acceleration/deceleration is also regulated and is in the range of e.g. +/- 10 ppm (that’s equal to 10 ns for an interval of 1 ms) for the PTP end instance offset correction
- (“value jump in case of set value” is always associated with SyncLoss to avoid destruction of equipment)
- The threshold/algorithm for the PTP end instance’s clock to declare inSync is within +/- 1 µs deviation to the GM
- 64/100 Hops must comply with the +/-1 µs deviation window to the GM
- GM and PTP relay and PTP end station timer have +/-50 ppm oscillators
- The frequency of these oscillators may change due to external influences e.g. df/dt of 3 ppm

Any violation of these requirements makes WorkingClock either useless OR destroys machines and equipment in the worst case.

Automation applications need to work independent from connected communication interfaces. Thus, use cases in which the synchronization is lost or reestablished shall be covered. This is often done by implementing two instances of a Clock for one PTP end instance. Figure 13 shows this clock usage model.

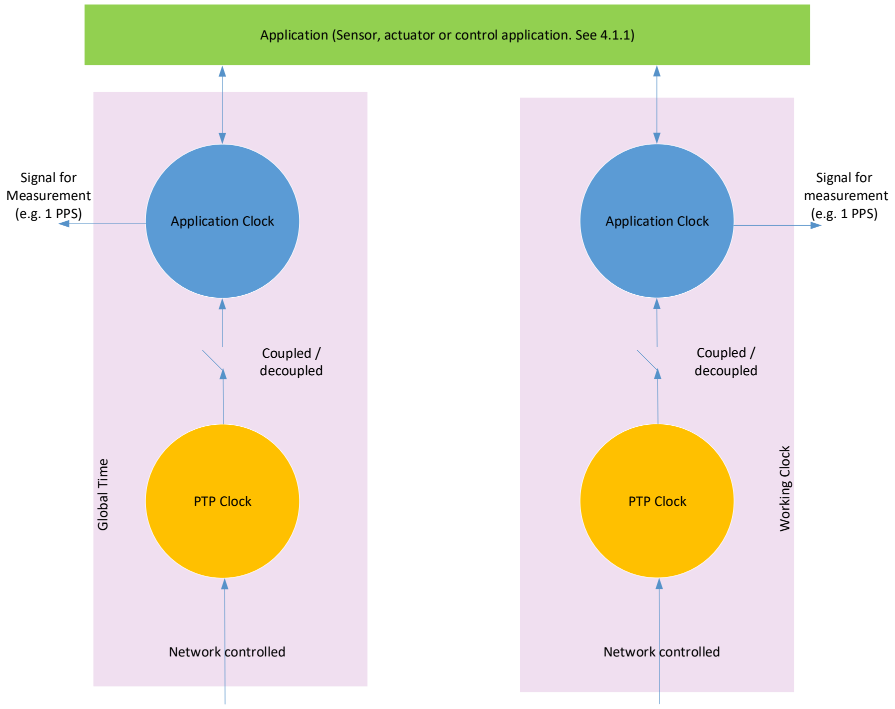


Figure 13: Clock usage model