Redundant Clock Synchronization

Astrit Ademaj

Günter Steindl

Timo Koskiahde

(in alphabetic order)

November 13, 2019

Issues with the Current Proposal

Fault Hypothesis

- Current solution considers permanent GM failures only
- GM failures require human intervention (60802 d.1.1 Clause 5.1.11.4)

Phase correction of the GM in the secondary domain

- 100ppm oscillator and a 125ms sync interval will lead to 12.5µs time deviation (not within the required accuracy boundaries)

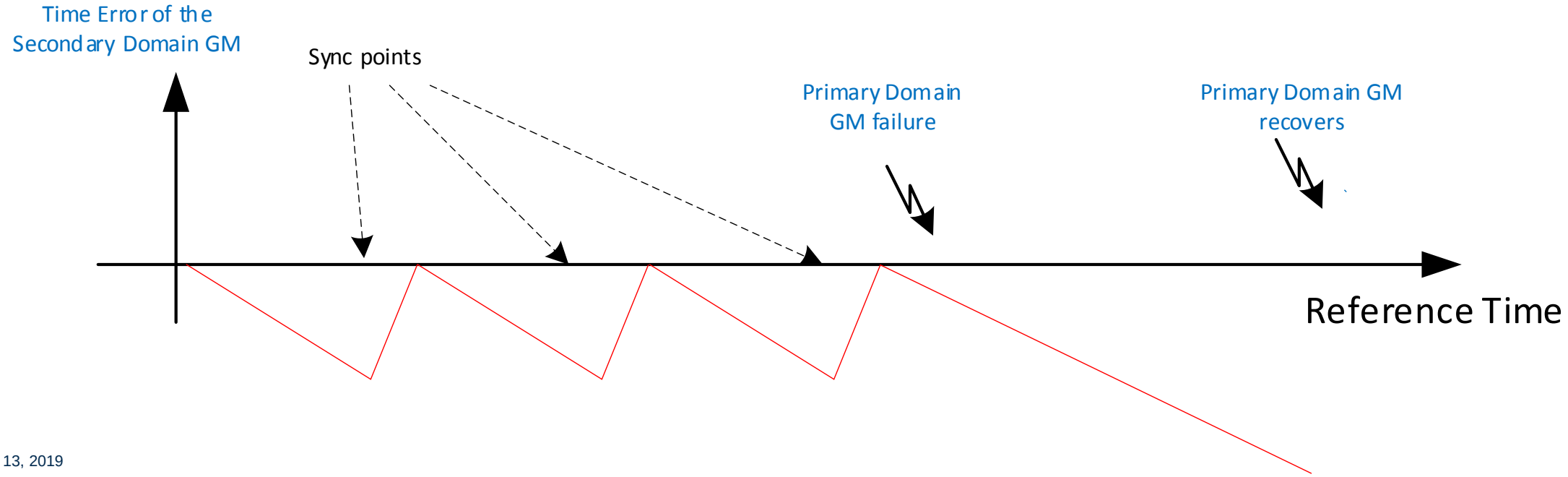
Issues with the Current Proposal

Fault Hypothesis

- Current solution considers permanent GM failures only
- GM failures require human intervention (60802 d.1.1 Clause 5.1.11.4)

Phase correction of the GM in the secondary domain

- 100ppm oscillator and a 125ms sync interval will lead to 12.5µs time deviation (not within the required accuracy boundaries)



Dual Role of Time [Kop97]

Time as Data

- Time is used to timestamp events
  - To establish order of events in a distributed computer system
- An error in the clock synchronization will invalidate the value of timestamps

Time as Control

- Time is used to control actions
  - Execution of scheduled traffic, traffic policing,…
  - Execution of distributed application tasks
- An error in the clock synchronization will invalidate the execution of the “control” mechanism

[Kop97] Hermann Kopetz , Real-Time Systems, Design Principles for Distributed Embedded Applications, Springer Verlag, 1997

Redundant Clock Synchronization

Working Clock (time as control)

- Seamless redundant (hot standby) clock synchronization is required to be able to meet the accuracy requirements
- The “working clock” shall provide linear increasing monotonic time (no time jumps, no frequency steps) in case of device failures
- Two domains - shall provide the same time and frequency

Global Time (time as data)

- Single domain is enough
- Clock synchronization (by using BMCA or externally manages sync trees) is able to fulfill the requirements

Concept for Redundant Clock Synchronization – Working Clock Only

- Use of two domains, whereas one domain is a primary domain
  - Primary Domain (PD – e.g., Domain A)
  - Secondary Domain (SD – e.g., Domain B)
- Each domain has two devices with GM capability, whereas one of them is an active GM
  - GM-A1 (GM in Domain A), GM-A2 (GM capability)
  - GM-B1 (GM in Domain B), GM-B2 (GM capability)
- GM within the respective domain is externally managed (static config). Upon a GM failover a GM switchover mechanisms shall elect the new GM
- No device may act as a GM in multiple domains (i.e. only one PTP instance in a device can act as a GM capable)
- GM in Secondary Domain (e.g., GM-B1) synchronizes in phase and frequency to the GM in the Primary Domain (PD-GM)
- Slaves in both domains use both domains to synchronize its working clock (if the GM in the respective domain is active and running)

Concept for Redundant Clock Synchronization – Failure Scenario

	Primary Domain (Domain A)		Secondary Domain (Domain B)	
Node:	**GM-A1**	GM-A2	**GM-B1**	GM-B2
Role:	PD-GM, GM in Domain A	Slave with GM capability	SD-GM / GM in Domain B	Slave with GM capability

1. GM-A1 → PD-GM
2. GM-B1 synchronizes to PD-GM (GM-A1)
   - slaves in both domain will synchronize to the working clock in Domain A and B.
3. GM-A1 fails, GM-B1 is running with the existing settings–
   - slaves (incl GM-A2) will synchronize to working clock from Domain B
4. GM-A2 becomes GM in Domain A (external managed sync tree)
   - redundant clock sync will declare the GM-A2 to PD-GM
   - GM-B1 synchronizes to PD-GM (GM-A2)
   - slaves in both domain will synchronize to the working clock in Domain A and B.
5. GM-A1 – restarts
   - GMA-A1 will synchronize to GM-A2 (PD-GM)
   - GM-A2 stays PD-GM

Concept for Redundant Clock Synchronization - Slave Topics

- Slaves use two time-domains to synchronize its working clock
  - In case of GM failure the time information from one domain is used
- Slave synchronization algorithm need not be defined
  - even a simple algorithm using two (multiple) domains will improve resilience

Concept for Redundant Clock Synchronization - GM Switchover

- Standard BMCA can be used in both domains
  - Provides failsafe (and still redundant) GM operations in case when a GM fails as another GM may step in using BMCA
  - Simple standard operation, no need for configuration

- Externally managed synchronization trees is a second option but it should be noted that network “static tree” reconfiguration is needed upon a GM failure

Concept for Redundant Clock Synchronization – Roles and States

Roles

- PD-GM: Primary Domain GM
- SD-GM: Secondary Domain GM
- Slave

States: PD-GM

INIT:	Executing Start-Up
FREE_RUNNING:	Operational

States: SD-GM

INIT:	Executing Start-Up
***SINGLE_SYNC*:**	Synchronized to one domain
FREE_RUNNING:	Not synchronized to PD-GM

States: Slave

INIT:	Executing Start-Up
***SINGLE_SYNC*:**	Synchronized to one domain
***REDUNDANT_SYNC*:**	Synchronized to both domains
FREE_RUNNING:	Not synchronized to any domain

Advantages of this proposal

- No clock jumps in case of GM failures and reintegration
- Redundant and robust clock sync mechanism
- Availability of GM (and the redundant sync) is “increased”
- No manual intervention is needed in case of (transient) GM failures