Fault-Tolerant gPTP

Byzantine Fault-Tolerant GM Time Agreement Generation and Preservation

Dan Finnegan – TTTech North America

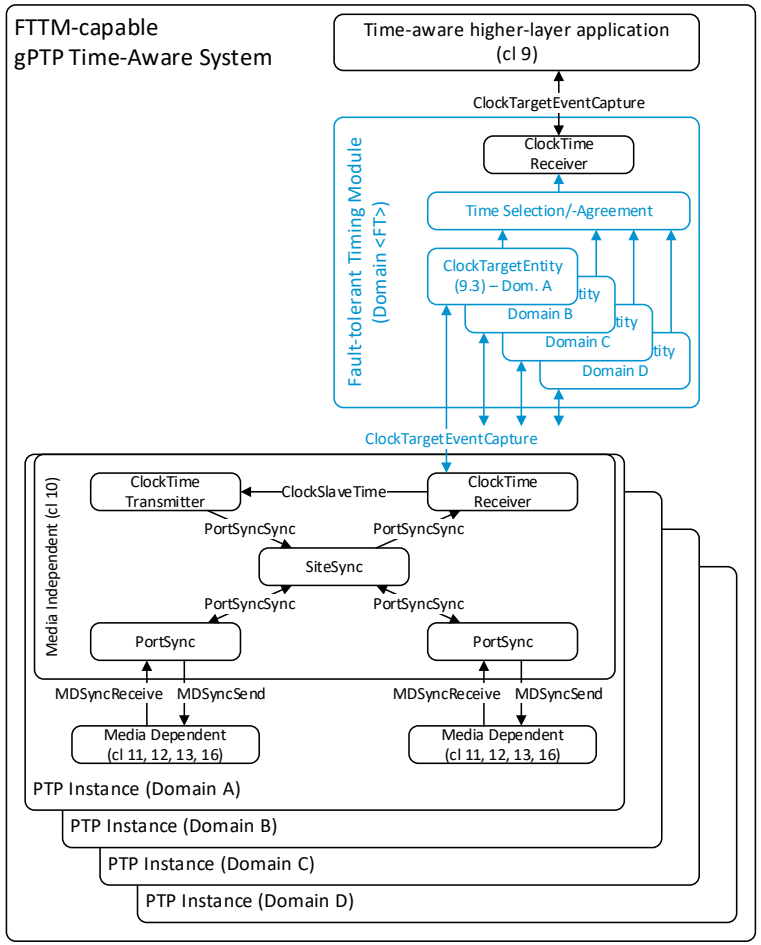
TTTech

FTTM

Structural Outline

- FTTM on top of *N* domains
- modelled as an instance, with
  - `ClockTimeReceiver`, and
  - dedicated/reserved domain ID

  to realize `ClockTargetEventCapture`



TTTech

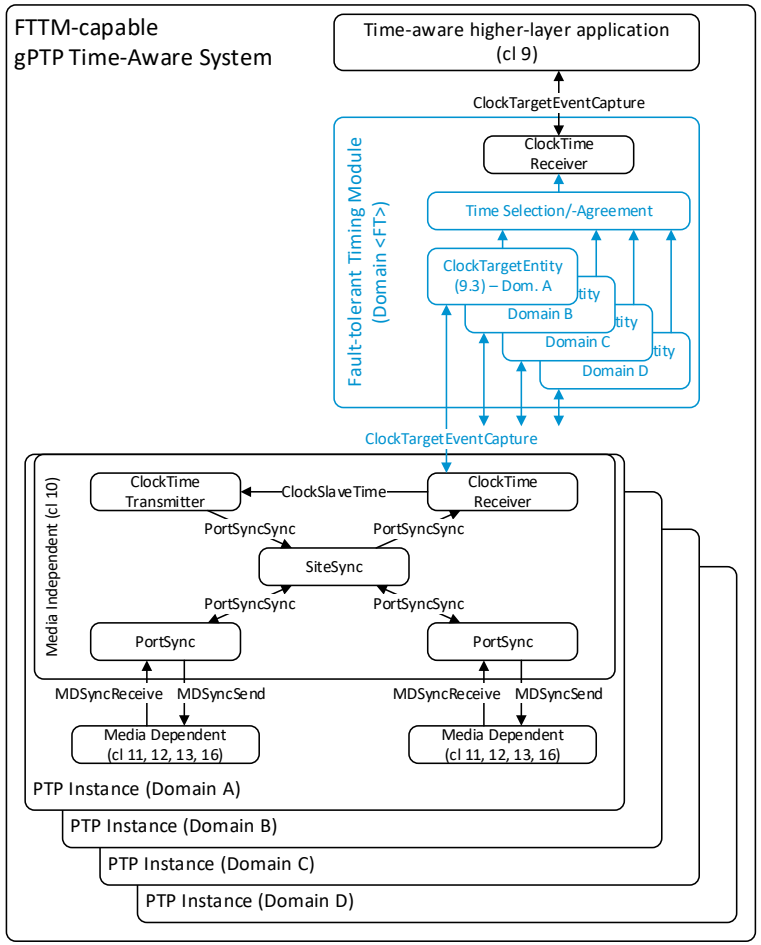
FTTM

Mid-Value Time-Index Selection Algorithm (MVTISA) and GM-agreement

- selects one time over all assigned domains (discards all others)
- provides this upon `ClockTargetEventCapture` of domain *FT*
- Propagation of Agreement to `ClockTimeTransmitter`
- Setting of domains via `ClockSourceTime`
- (only necessary to domains which host at least one `TimeTransmitterPort`)

→ Feasible for gPTP relay instances (non-GM end-points and bridges)
→ Requires at least two “paired” GMs for normal function
→ high computational complexity

---How to achieve GM agreement?---



Grandmaster Time Agreement Generation and Preservation Function

- FTTM using MVTISA expects as an input multiple independent but synchronized time domains
  - Requires multiple independent Grandmaster clocks

  AND
  - These Grandmaster clocks must be synchronized

- FTTM requires at least two “paired” GMs for normal function
  - A breakdown in the synchronization of the GMs will result in a failure of the FTTM to identify valid Redundant Time Domains

- **The actual degree of fault tolerance provided by the FTTM is limited by the reliability of the synchronization of the input GMs**

- To use Synchronous 802.1DP TSN for critical functions Grandmaster Time Agreement Generation and Preservation Function must be Fault-Tolerant!

TTTech

Time Agreement using FTA

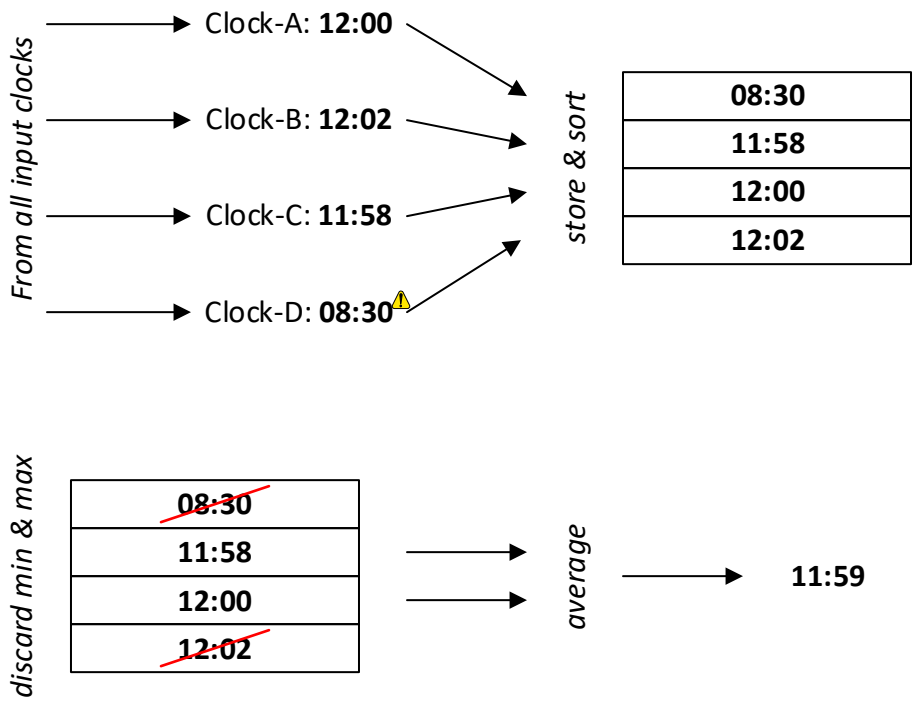
in the presence of Byzantine faults.

3k + 1 clock readings required to tolerate *k* Byzantine faults, e.g.,

- 4 clock readings for *k = 1*
- 7 clock readings for *k = 2*

Possible solution:

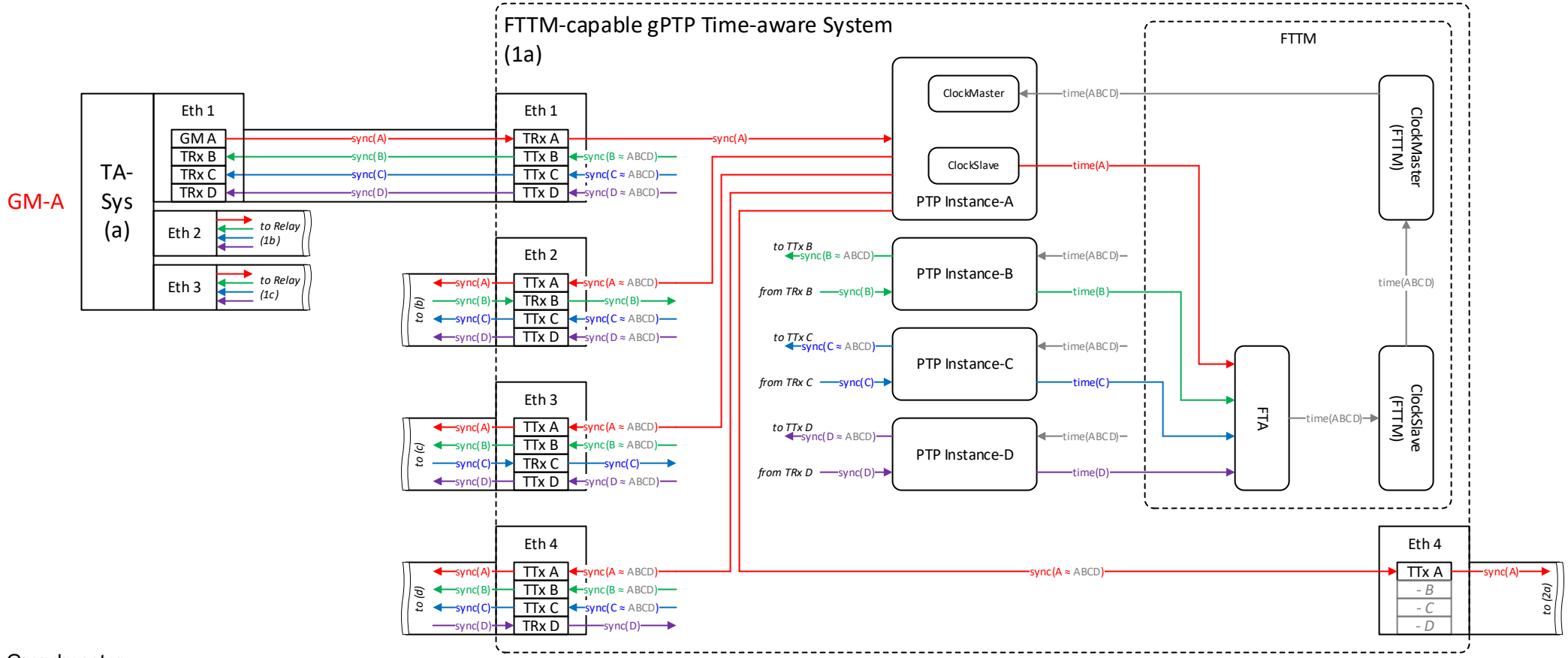
Fault **T**olerant **A**verage (FTA) Algorithm



Example: FTA for *k = 1, with one faulty clock (Clock-D)*

FTTM Agreement using FTA

in gPTP Time-aware Systems



GM : Grandmaster
TTx: Time Transmitter
TRx: Time Receiver

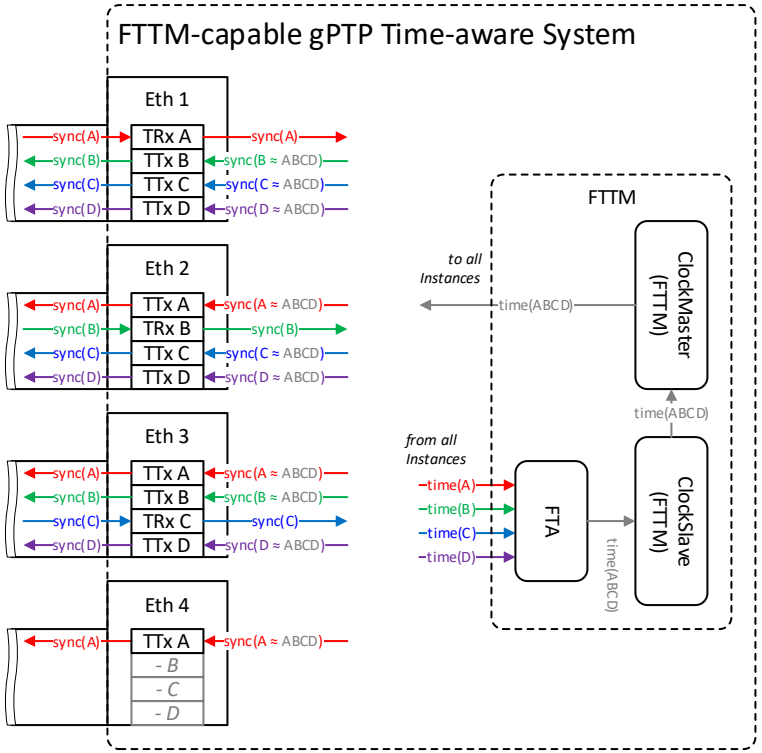
TTTech

Time Agreement using FTA

in gPTP Time-aware Systems

- Every GM requires
  - 3 remote clock readings from other GMs
  - (it already knows its own time)
- gPTP operates link-locally

FTTM with FTA
- must be executed by every GM
- may be executed by every gPTP relay instance
  - eliminate faulty times to prevent propagation through the network
    - → Fault may even have originated at the respective GM



TTx: Time Transmitter
TRx: Time Receiver

TTTech

FTTM

Suggestions for P802.1ASed

- Explicitly model FTTM as “domain” to provide interface `ClockTargetEventCapture`
  - May also need `ClockSourceTime` for GPS uplink?
- Require MVTISA ***or*** FTA as Time Selection/Agreement Algorithm
  - Each requires a dedicated management/YANG subtree
  - Require at least one of MVTISA or FTA in PICS proforma (O.x notation)
- Specify feedback from selected/agreed time back to locally supported domains
  - Required for GMs
  - Optional for gPTP relay instances
  - (Not needed for End instances)

Note: “Other” Time Selection/Agreement algorithms or external GM agreement may be allowed, but this would make common management impossible

Open Questions

Establishment of a reasonable wall clock

- Establishment of reasonable wall clock
  - MVTISA cannot guarantee
    - all devices select the same domain, or
    - It can select a time at all (e.g., due to GM-agreement not achieved yet)
  - FTA can always calculate an average, but the average may be not useful
    → 01 Jan 1970 averaged with 17 Mar 2025 is 05 Aug 1997

- Possible options
  - Plan A - External time source, e.g., GPS?
    - What about GPS-denied environments?
      - Issue potentially avoidable if only relied upon during startup
      - May resync via FTTM-instance's `ClockSourceTime` interface, when available again

    → Restarting with GPS unavailable remains an issue
  - Plan B - Startup preferring a certain domain?
    - Issue: this would require a relaxed fault hypothesis during startup
  - Plan C - Establish a default start time as "time of last resort" when GPS or preferred domains fail
    - At start of epoch? (01 Jan 1970)
    - Include an indication "degraded" status?


TTTech