P802.1AS-2025-Rev: syncGrandMasterIdentity, syncStepsRemoved, and syncEgressTimestamp propagation

Alon Regev [Keysight]

Introduction

- 802.1AS-2025-Rev-D0.2 comment 50:

  with Drift_Tracking TLV usage, the global variables syncGrandMasterIdentity, syncStepsRemoved, and syncEgressTimestamp are copied to globals regardless of whether the receiving PTP Port is in TimeReceiver portState so a transmitting port (and internal usage) would randomly get the values from one of the ports that receive Sync / Follow_up messages.
- This slide set discusses the issues and options to address
- Underlying documents:
  - [AS]: 802.1AS-2025
  - [ASRev]: P802.1AS-2025 Rev/D0.3
- Previous relevant presentations:
  - https://www.ieee802.org/1/files/public/docs2026/as-regev-drift-tracking-global-assignments-0926-v00.pdf

PTP Port states observations (single PTP Instance)

- Can be assigned by BTCA or external port configuration
- At most one port is a TimeReceiver
- Ports that receive sync messages but are not the selected TimeReceiver are typically PassivePort
- A PassivePort typically connects to a TimeTransmitter

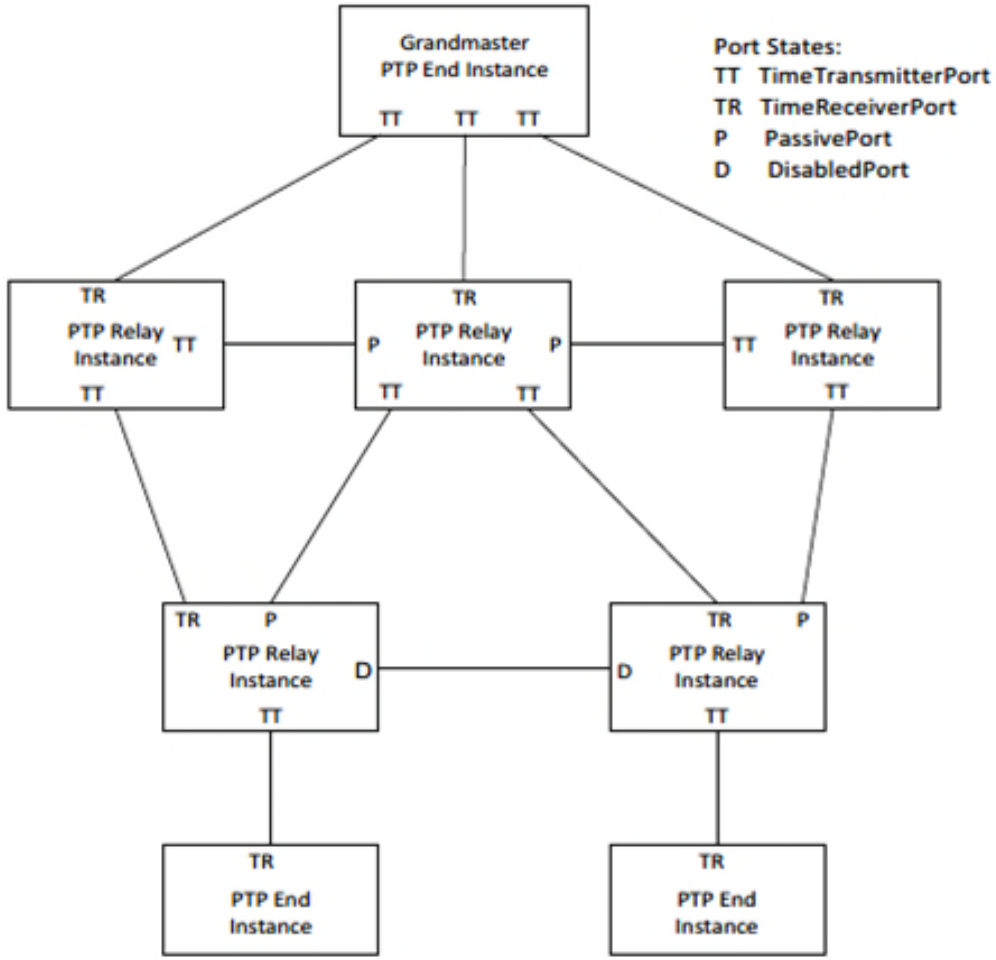


Figure 10-11—Example timeTransmitter/timeReceiver hierarchy of PTP Instances

Summary of nNew observations from Silvana

- In the previous slide, the key point was regarding Passive Ports would receive sync messages.
- However, 802.1AS utilizes the MDMI interface defined in 1588 (see figure 44 from 1588 on this slide)
- Clause 11.5.2.2.2 of IEEE 1588 states:
  - “If the PTP Port is in the PASSIVE state, no timing information is transferred across the MDMI interface.”
- So, there is no issues for other media, except for HDE

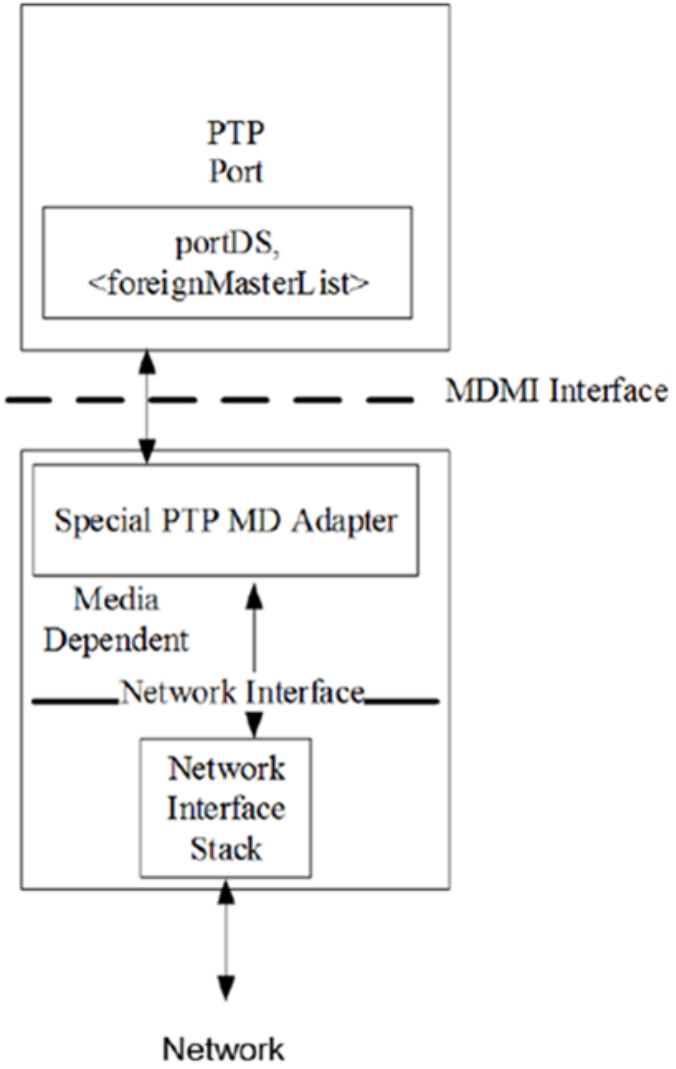


Figure 44—Special Port model

High level notes from email discussions

- If 802.1AS uses the MDMI Interface from 1588 and that it is using the “Special PTP MD Adapter” this should be explicitly stated in 802.1AS
- The MDMI interface used in 802.1AS is different than the one defined in 1588 (neither the MDSyncReceive nor the MDSyncSend structures match)
  - This needs to be addressed on one side or the other, but effectively 802.1AS is using a different interface than that defined in 1588
- 1588 refers to clauses in 802.1AS-2011 specifically (and not 802.1AS-2020 or 2025 or Rev)
  - From 11.5.2.2.3 of 1588:

> The subclauses of IEEE Std 802.1AS-2011 referenced below are the only subclauses of IEEE Std 802.1AS-2011 that are normative for the Special PTP MD Adapter block.
>
> The Special PTP MD Adapter block for IEEE 802.11 transport shall meet the requirements in the following subclauses of Clause 12 of IEEE Std 802.1AS-2011: 12.1.1, which describes the Timing Measurement; 12.1.2, which describes the layering; 12.4.1, which describes the master state machine; 12.4.2, which describes the slave state machine; and 12.5, which describes the carrying of information needed by the PTP protocol.

  - 802.1AS-2011 is being replaced by 802.1AS-2025 in the 1588 rollup, but the issues in the structures are not being addressed as they are not in approved amendments
- We may still have an issue with HDE, but is there a use-case for Drift-Tracking with HDE?

MDMI history as it relates to 802.1AS

- 802.1AS was released in 2011, referencing 1588-2008
- MDMI (Media-Dependent-Media-Independent) was introduced in IEEE 1588-2019
  - To apply MDMI to 802.1AS-2011, text was added in 1588-2019 with the references to (and requirements for) 802.1AS-2011
- Newer versions of 802.1AS (2020, 2025, etc.) were released but nether 1588 reference any newer version of 802.1AS nor does 802.1AS require the use of MDMI
- 1588 rollup currently in development updates references to AS-2025

MDSyncReceive

- 1588-2019:

```
MDSyncReceive {
domainNumber,
sdoId,
sourcePortIdentity,
originOrPreciseOriginTimestamp,
correctionField,
logMessageInterval,
upstreamTxTime,
cumulativeRateRatio
}
```

- 802.1AS-2025:

```
MDSyncReceive {
        domainNumber,
        followUpCorrectionField,
        sourcePortIdentity,
        logMessageInterval,
        preciseOriginTimestamp,
        upstreamTxTime,
        rateRatio,
        gmTimeBaseIndicator,
        lastGmPhaseChange,
        lastGmFreqChange
}
```

MDSyncSend

- 1588-2019:

```
MDSyncSend {
domainNumber,
sdoId,
sourcePortIdentity,
originOrPreciseOriginTimestamp,
correctionField,
logMessageInterval,
upstreamTxTime,
cumulativeRateRatio
}
```

- 802.1AS-2025:

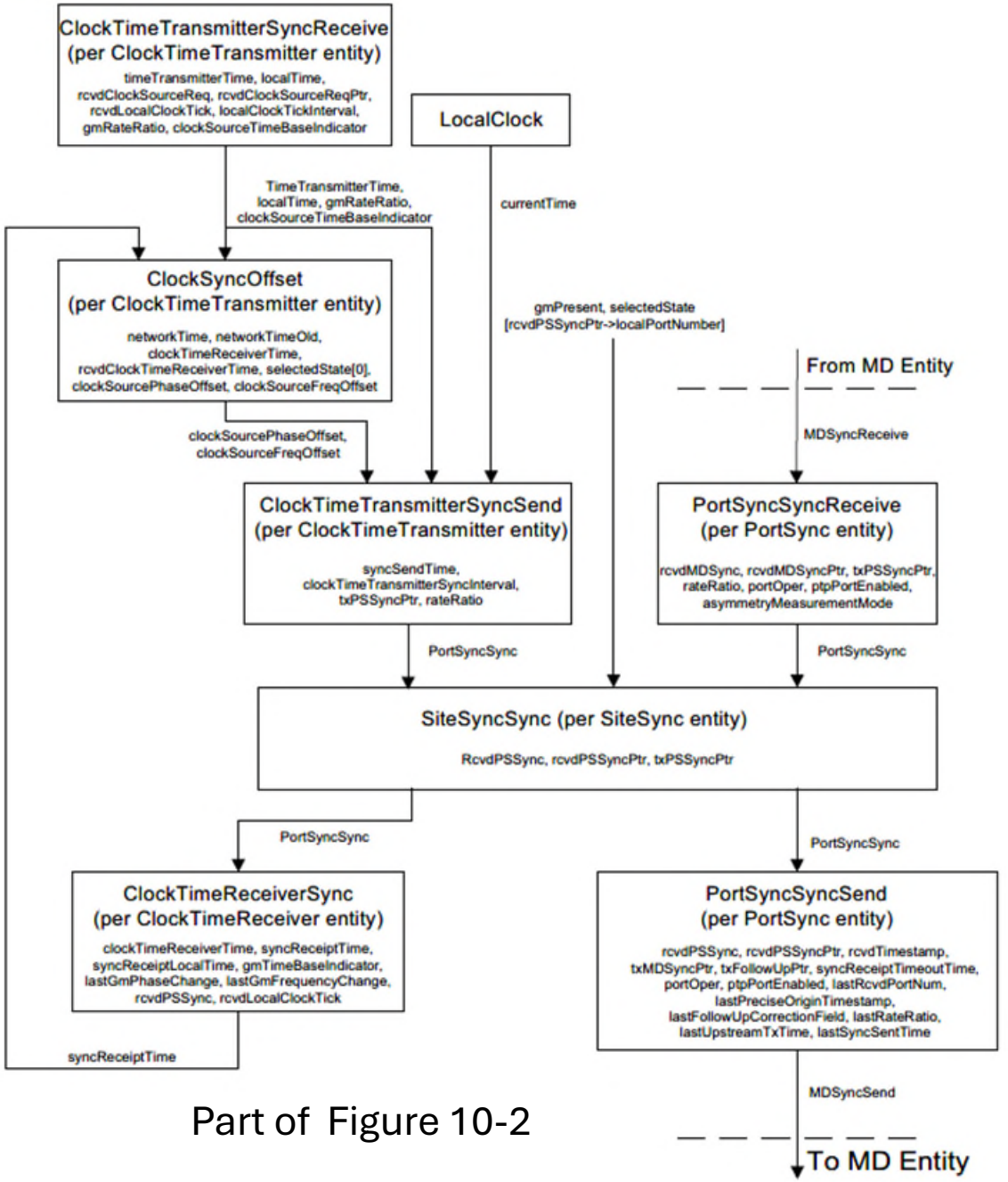
```
MDSyncSend    {
        domainNumber,
        followUpCorrectionField,
        sourcePortIdentity,
        logMessageInterval,
        preciseOriginTimestamp,
        upstreamTxTime,
        rateRatio,
        gmTimeBaseIndicator,
        lastGmPhaseChange,
        lastGmFreqChange
}
```

Discussion

Backup slides from past discussions

Port State awareness in time synchronization state machines

- Only the SiteSyncSync state machine is acting on the basis of Port State
  - It checks if (selectedState[rcvdPSSyncPtrSSS->localPortNumber] == TimeReceiverPort) as a qualifier to entering the RECEIVING_SYNC state and forwarding data from the incoming PortSyncSyncSync structure to all the PortSyncSyncSend and ClockTImeRecieveSync SMs.
- The PortSyncSyncReceive and related MDSyncReceiveSM do not take Port State into account
  - Data from all incoming Sync messages is processed and forwarded towards the SiteSyncSync SM



Part of Figure 10-2

Drift tracking implementation

- In 11.2.13.2.1 setMDSyncReceiveMDSR() f):

  1) syncGrandmasterIdentity (see 10.2.4.26) is set equal to the syncGrandmasterIdentity of the Drift_Tracking TLV.

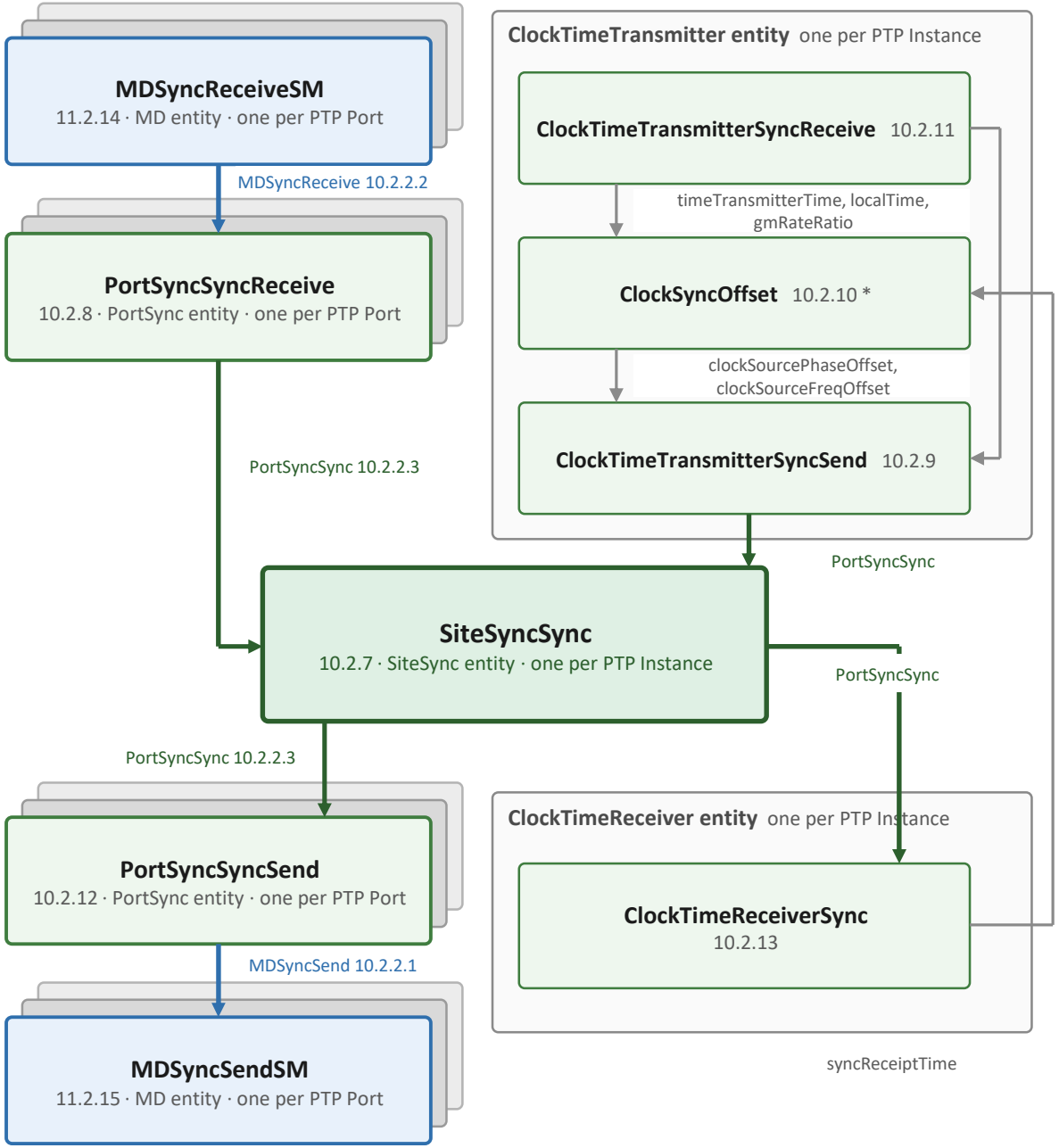
  2) If the value of syncStepsRemoved of the Drift_Tracking TLV is not equal to 0xFFFF, syncStepsRemoved (see 10.2.4.27) is set equal to the value of syncStepsRemoved of the Drift_Tracking TLV plus one. If the value of syncStepsRemoved of the Drift_Tracking TLV is equal to 0xFFFF, syncStepsRemoved (see 10.2.4.27) is set equal to 0xFFFF.

- In 11.2.14.2.3 setFollowUp() g):

  1) syncGrandmasterIdentity is set equal to the value of the global variable syncGrandmasterIdentity (see 10.2.4.26).

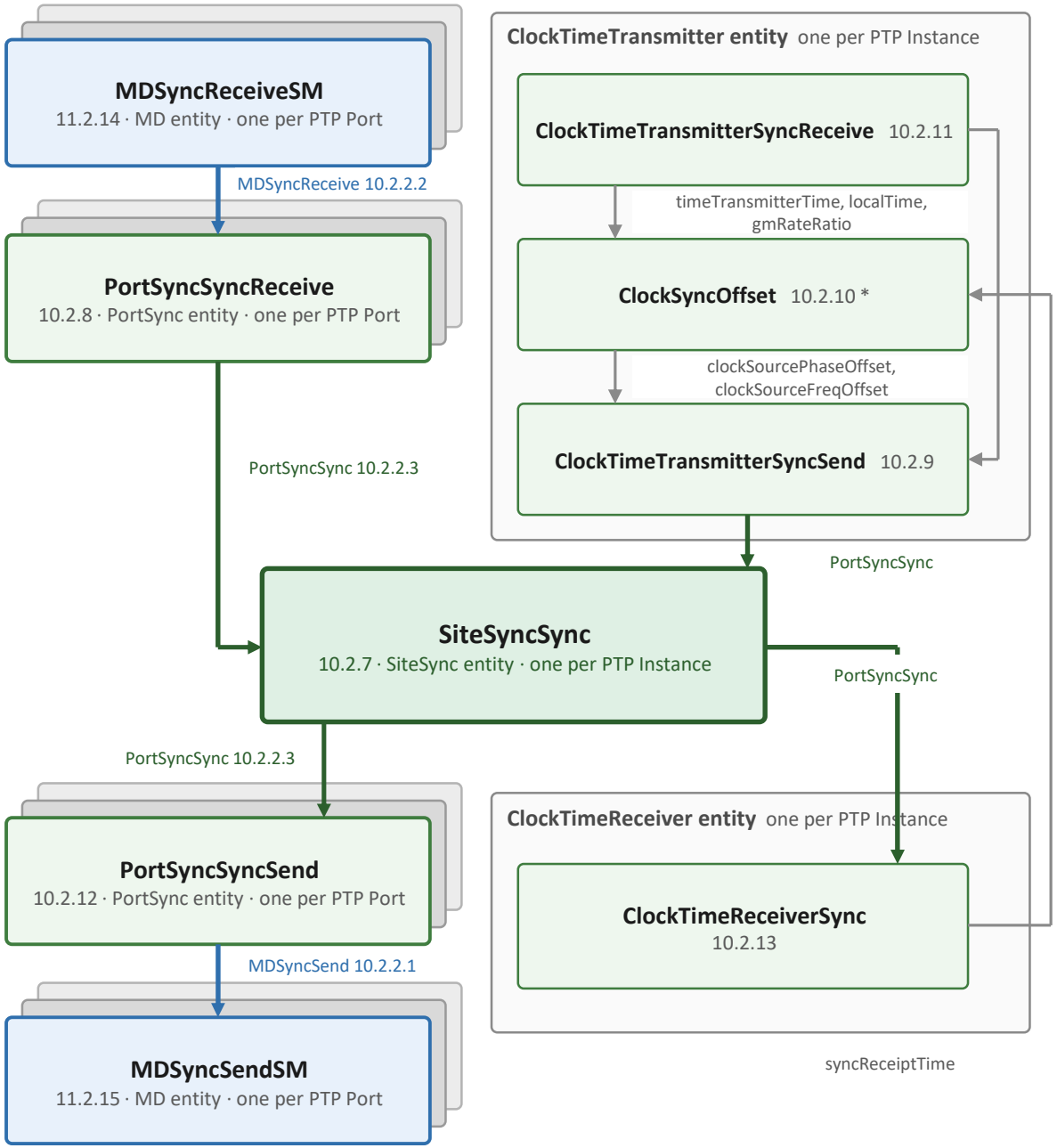
  2) syncStepsRemoved is set equal to the value of the global variable syncStepsRemoved (see 10.2.4.27).

- The MDSyncReceiveSM does not take Port state into effect, so a Sync message received on any port (not only the TimeReceiver port) will update the globals.
- The value sent by the MDSyncSendSM could contain a combination of Sync data received on one port (the TimeReceiver port) and globals set from Drift_Tracking TLV received on a different port (such as a Passive or Disabled port)



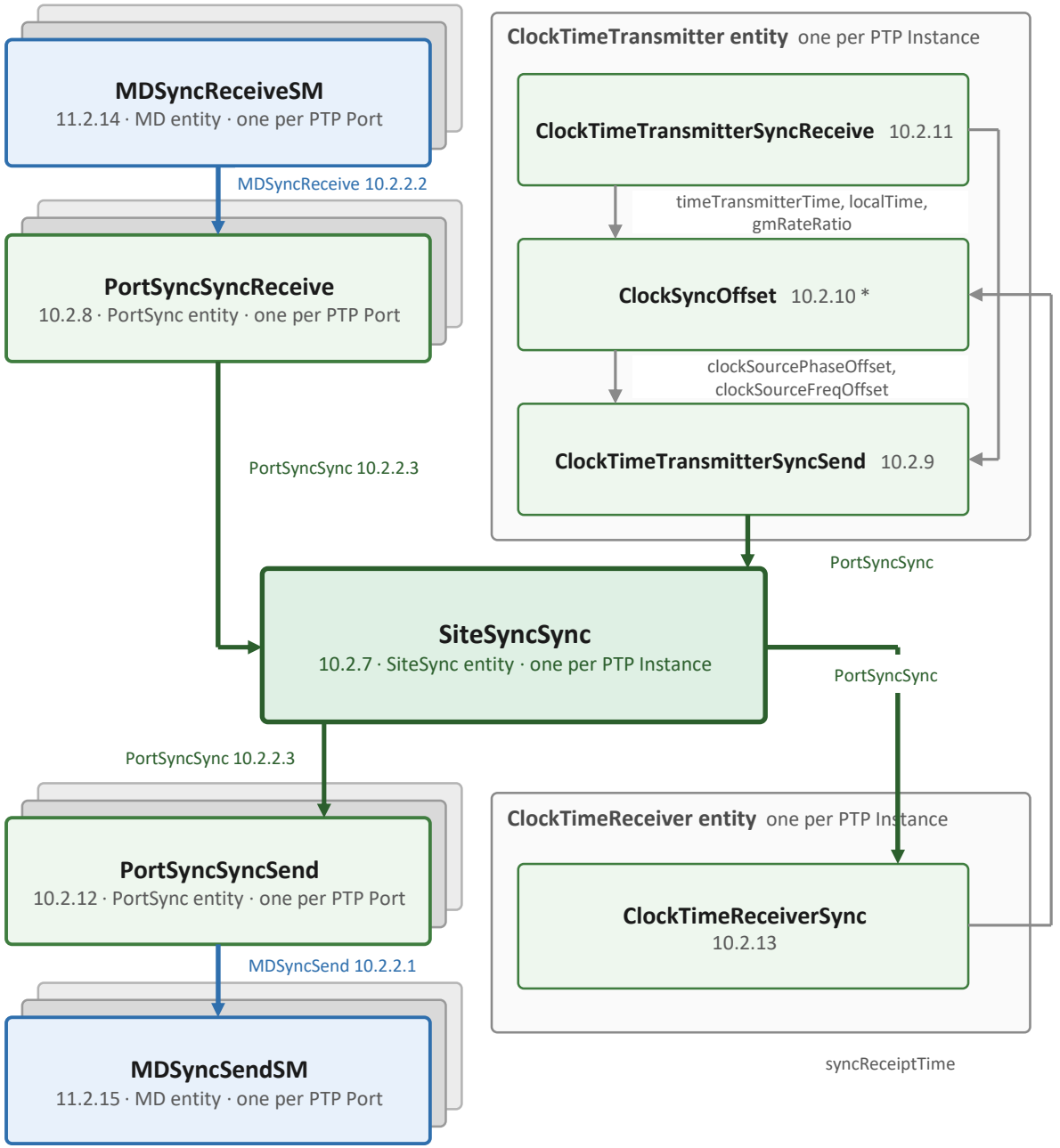
Possible solution 1: MDSyncReceiveSM to check Port State prior to assigning data to globals

- Modify 11.2.13.2.1 f) and g), add “If the Port State of the PTP Port is TimeReceiver”
- Pros
  - Likely simplest solution
- Cons
  - Potential race condition when clock source changes
  - Inconsistent with structure of the rest of the state machines
  - Does not account for drift between the GM clock source and GM local clock
  - Does not allow Drift Tracking support by other MD types



Possible solution 2: Transfer the syncGrandmasterIdentity and syncStepsRemoved to the SiteSyncSync SM and have the SiteSyncSync choose the source

- Modify MDSyncReceive, PortSyncSync, and MDSyncSend structures to add the Drift_Tracking data
- Transfer the data along the path from the MDSyncReceiveSM to the SiteSyncSync
- The SiteSyncSync will transfer the data from the current TimeReceiver
- Pros
  - Consistent with structure of the rest of the state machines
  - Allows adding Drift Tracking support to other MD types and to drift between the GM clock source and GM local clock
- Cons
  - May require touching every MD clause to set Drift Tracking fields in the MDSyncReceive structure to default values.
  - More complicated change
- Pro/Con:
  - Moves Drift Tracking from Clause 11 to Clause 10 (which depends on your opinion may be good or bad)
  - How much of Drift Tracking to move can be discussed



Possible solution 3: Add optional “MD-specific” or “feature-specific” additions to the MDSyncReceive, PortSyncSync, and MDSyncSend structures

- Modify MDSyncReceive, PortSyncSync, and MDSyncSend structures to allow adding either “MD_specific” or “feature-specific” additions
- Transfer the data along the path from the MDSyncReceiveSM to the SiteSyncSync
- If the otional “MD-specific” or “feature-specific” fields are present, the SiteSyncSync will transfer the data from the current TimeReceiver
- Pros
  - Consistent with structure of the rest of the state machines
  - Allows adding Drift Tracking support to other MD types and to drift between the GM clock source and GM local clock
  - Potentially allows supporting other future data
- Cons
  - More complicated change
- Pro / Con
  - Need to decide where Drift Tracking lives (as this approach allows making it mostly independent of a specific MD clause if desired

