10. Media-independent layer specification

10.1 Overview

10.1.1 General

Unless otherwise stated, information in this clause is per domain.

10.1.2 Model of operation

A PTP Instance contains a best master selection function and a synchronization function. These functions include PTP Port-specific aspects and aspects associated with the PTP Instance as a whole. The functions are distributed among a number of entities, which together describe the behavior of a compliant implementation. The functions are specified by a number of state machines.

The model for the media-independent layer of a PTP Instance is shown in Figure 10-1. It includes a single SiteSync entity, ClockMaster entity, and ClockSlave entity for the PTP Instance as a whole, plus one PortSync for each PTP Port. The PTP Instance also includes one MD entity for each PTP Port, which is part of the media-dependent layer. The media-dependent functions performed by the MD entity are described in the clauses for the respective media. In addition to the entities, Figure 10-1 shows the information that flows between the entities via the PortSyncSync, MDSyncSend, and MDSyncReceive structures (see 10.2.2.3, 10.2.2.1, and 10.2.2.2, respectively).

The SiteSync, ClockMaster, ClockSlave, and PortSync entities each contain a number of cooperating state machines, which are described later in this clause (the MD entity state machines are described in the respective media-dependent clauses). The ClockMaster entity receives information from an external time source, known as a *ClockSource entity* (see 9.2), via an application interface, and provides the information to the SiteSync entity. The ClockSlave entity receives Grandmaster Clock time-synchronization and current Grandmaster PTP Instance information from the SiteSync entity, and makes the information available to an external application, known as a *clockTarget entity* (see 9.3 through 9.6), via one or more application service interfaces. The SiteSync entity executes the portion of best master clock selection associated with the PTP Instance as a whole, i.e., it uses the best master information received on each PTP Port to determine which PTP Port has received the best information, and updates the states of all the ports (see 10.3.1.1 for a discussion of PTP Port states). It also distributes synchronization information received on the SlavePort to all the ports whose state is MasterPort (see 10.3.1.1). The PortSync entity for a SlavePort receives best master selection information from the PTP Instance at the other end of the associated link, compares this to the current best master information that it has, and forwards the result of the comparison to the Site Sync entity. The PortSync entity for a SlavePort also receives time-synchronization information from the MD entity associated with the PTP Port, and forwards it to the SiteSync entity. The PortSync entity for a MasterPort sends best master selection and time-synchronization information to the MD entity for the PTP Port, which in turn sends the respective messages.

NOTE—This clause does not require a one-to-one correspondence between the PortSync entities of PTP Instances attached to the same gPTP communication path (see 3.11), i.e., more than two PTP Instances can be attached to a gPTP communication path that uses a shared medium and meet the requirements of this clause. However, it is possible for a media-dependent clause to have additional requirements that limit the gPTP communication paths to point-to-point links for that medium; in this case, each link has exactly two PortSync entities, which can be considered to be in one-to-one correspondence. One example of this is the full-duplex point-to-point media-dependent layer specified in 11. In addition, one or more gPTP communication paths can be logically point-to-point but traverse the same shared medium.

The time-synchronization state machines are described in 10.2. The best master clock selection state machines are described in 10.3. The attributes and format of the Announce message are described in 10.5 and 10.6. The timing characterization of the protocol is described in 10.7.

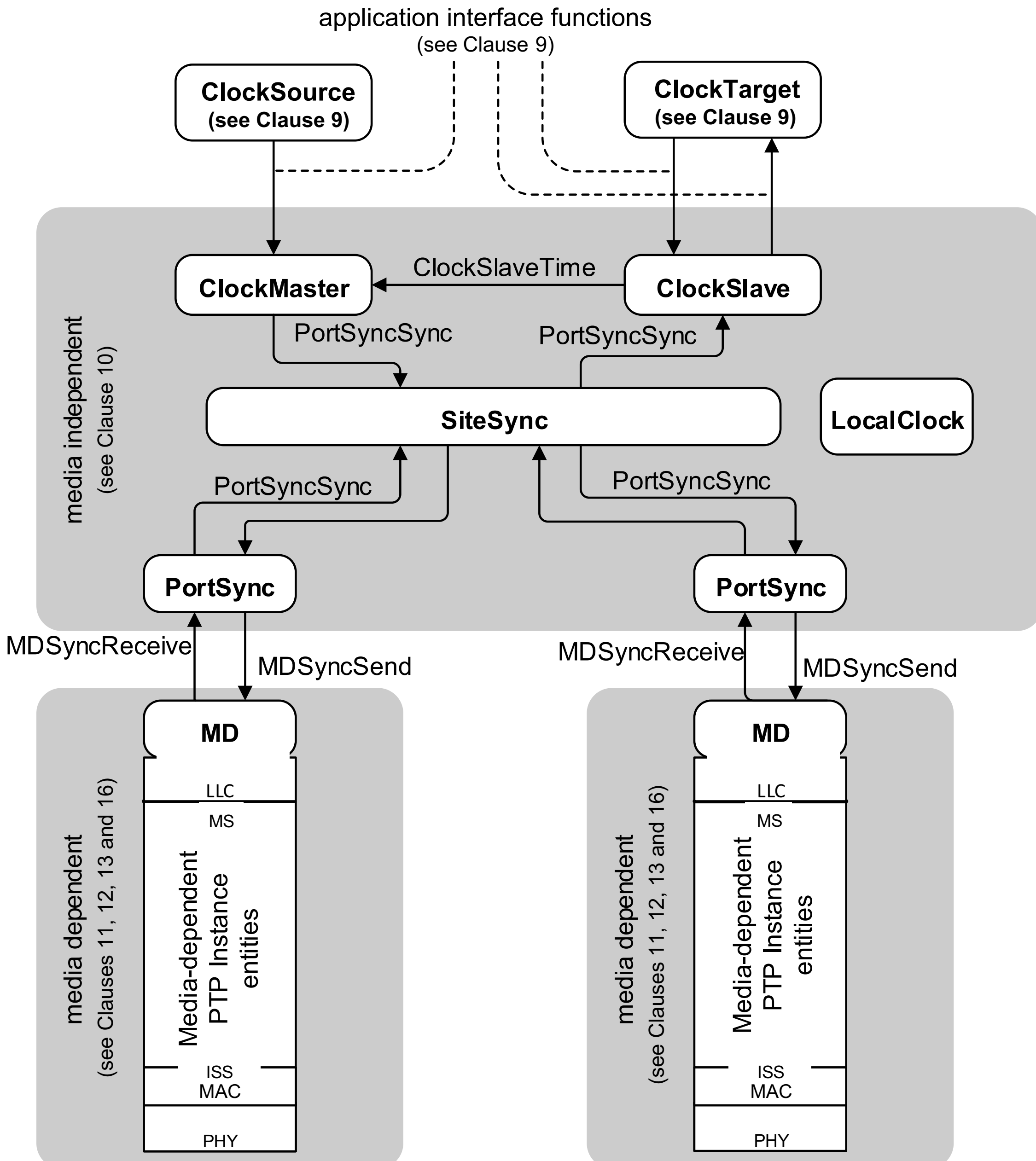


Figure 10-1—Model for media-independent layer of PTP Instance

10.1.2.1 LocalClock entity

The LocalClock entity is a free-running local clock (see 3.16) that provides a common time to the PTP Instance, relative to an arbitrary epoch. A PTP Instance contains a LocalClock entity. The requirements for the LocalClock entity are specified in B.1. All timestamps are taken relative to the LocalClock entity (see 8.4.3). The LocalClock entity also provides the value of currentTime (see 10.2.4.12), which is used in the state machines to specify the various timers.

NOTE—The epoch for the LocalClock entity can be the time that the PTP Instance is powered on.

10.1.3 Grandmaster-capable PTP Instance

A PTP Instance may be grandmaster-capable. An implementation may provide the ability to configure a PTP Instance as grandmaster-capable via a management interface.

NOTE 1—The managed object gmCapable is read only (see Table 14-1). gmCapable is configured by setting the value of the managed object priority1, which is read/write (see Table 14-1). If the value of priority1 is 255, then

a) gmCapable is set to FALSE (see 8.6.2.1), and
b) The value of the managed object clockClass, which is read only, is set to 255 (see 8.6.2.2).

NOTE 2—While a PTP Instance that is not grandmaster-capable can never be the Grandmaster PTP Instance of the gPTP domain, such a PTP Instance contains a best master selection function, invokes the best master selection algorithm, and conveys synchronization information received from the current Grandmaster PTP Instance.

10.2 Time-synchronization state machines

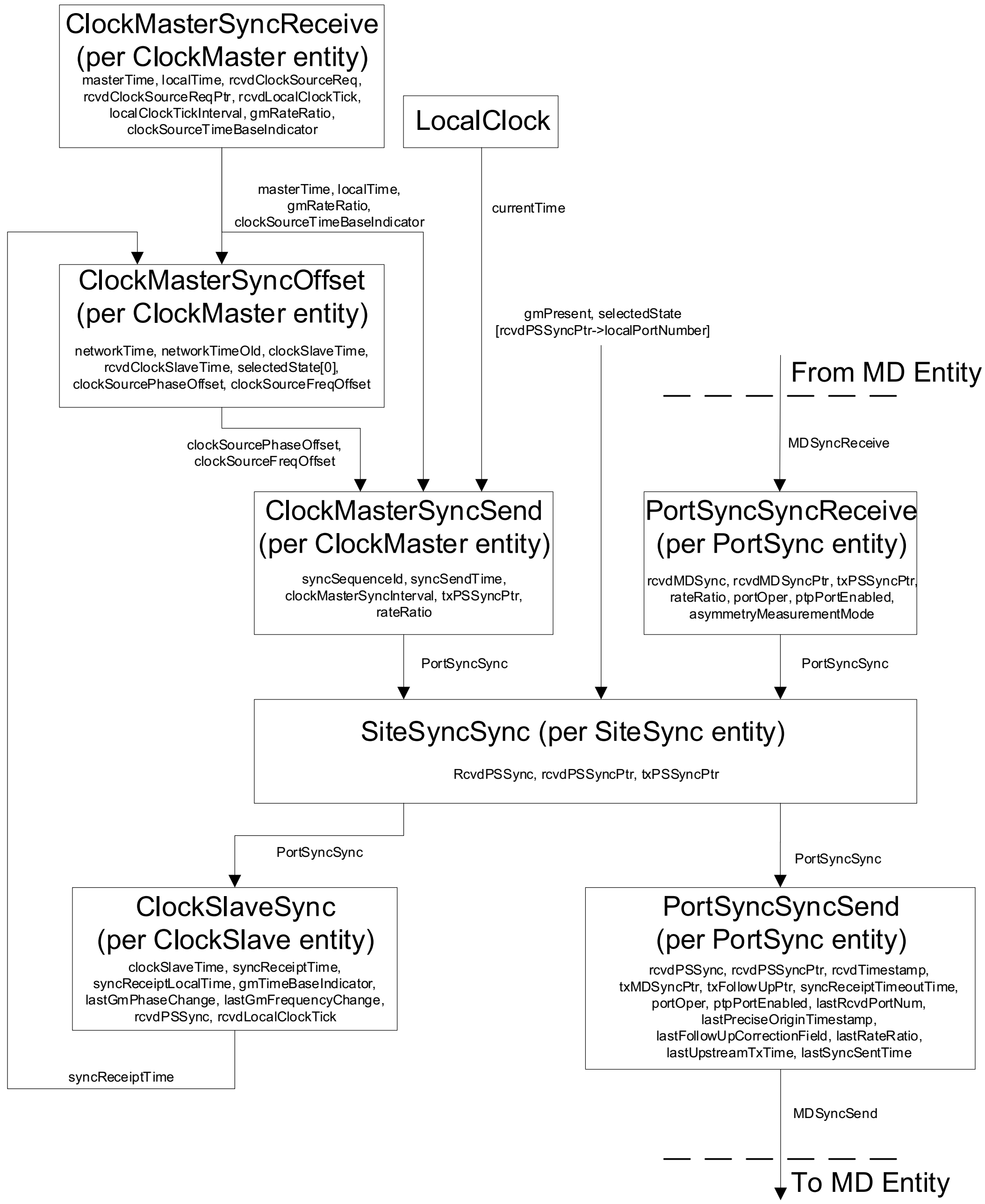
10.2.1 Overview

The time-synchronization function in a PTP Instance is specified by a number of cooperating state machines. Figure 10-2 illustrates these state machines, their local variables, their interrelationships, and the global variables and structures used to communicate between them. The figure indicates the interaction between the state machines and the media-dependent layer and LocalClock entity.

The ClockMasterSyncReceive, ClockMasterSyncOffset, and ClockMasterSyncSend state machines are optional for PTP Instances that are not grandmaster-capable (see 8.6.2.1 and 10.1.3). These state machines may be present in a PTP Instance that is not grandmaster-capable; however, any information supplied by them, via the ClockMasterSyncSend state machine, to the SiteSyncSync state machine is not used by the SiteSyncSync state machine if the PTP Instance is not grandmaster-capable.

The media-independent layer state machines in Figure 10-2 are as follows:

a) ClockMasterSyncReceive (one instance per PTP Instance): receives ClockSourceTime.invoke functions from the ClockSource entity and notifications of LocalClock entity ticks (see 10.2.4.18), updates masterTime, and provides masterTime to ClockMasterSyncOffset and ClockMasterSyncSend state machines.
b) ClockMasterSyncOffset (one instance per PTP Instance): receives syncReceiptTime from the ClockSlave entity and masterTime from the ClockMasterSyncReceive state machine, computes phase offset and frequency offset between masterTime and syncReceiptTime if the PTP Instance is not the Grandmaster PTP Instance, and provides the frequency and phase offsets to the ClockMasterSyncSend state machine.
c) ClockMasterSyncSend (one instance per PTP Instance): receives masterTime from the ClockMasterSyncReceive state machine, receives phase and frequency offset between masterTime and syncReceiptTime from the ClockMasterSyncOffset state machine, and provides masterTime (i.e., synchronized time) and the phase and frequency offset to the SiteSync entity using a PortSyncSync structure.
d) PortSyncSyncReceive (one instance per PTP Instance, per PTP Port): receives time-synchronization information from the MD entity of the corresponding PTP Port, computes accumulated rateRatio, computes syncReceiptTimeoutTime, and sends the information to the SiteSync entity.
e) SiteSyncSync (one instance per PTP Instance): receives time-synchronization information, accumulated rateRatio, and syncReceiptTimeoutTime from the PortSync entity of the current slave port or from the ClockMaster entity; and sends the information to the PortSync entities of all the ports and to the ClockSlave entity.



Notes:
a) selectedState for each port and gmPresent are set by Port State Selection state machine (see 10.3.12)
b) currentTime is a global variable that is always equal to the current time relative to the local oscillator
c) application interfaces to higher layers are not shown
d) the ClockMasterSyncReceive, ClockMasterSyncSend, and ClockMasterSyncOffset state machines are optional for PTP Instances that are not grandmaster-capable.

Figure 10-2—Time-synchronization state machines—overview and interrelationships

f) PortSyncSyncSend (one instance per PTP Instance, per PTP Port): receives time-synchronization information from the SiteSync entity, requests that the MD entity of the corresponding PTP Port send a time-synchronization event message, receives the syncEventEgressTimestamp for this event message from the MD entity, uses the most recent time-synchronization information received from the SiteSync entity and the timestamp to compute time-synchronization information that will be sent by the MD entity in a general message (e.g., for full-duplex IEEE 802.3 media) or a subsequent event message (e.g., for IEEE 802.11 media), and sends this latter information to the MD entity.

g) ClockSlaveSync (one instance per PTP Instance): receives time-synchronization information from the SiteSync entity; computes clockSlaveTime and syncReceiptTime; sets syncReceiptLocalTime, GmTimeBaseIndicator, lastGmPhaseChange, and lastGmFreqChange; sends clockSlaveTime to the ClockMaster entity; and provides information to the ClockTarget entity (via the ClockTargetPhaseDiscontinuity interface; see 9.6) to enable that entity to determine if a phase or frequency discontinuity has occurred.

10.2.2 Data structures communicated between state machines

The following subclauses describe the data structures communicated between the time-synchronization state machines.

10.2.2.1 MDSyncSend

10.2.2.1.1 General

This structure contains information that is sent by the PortSync entity of a PTP Port to the MD entity of that PTP Port when requesting that the MD entity cause time-synchronization information to be sent. The structure contains information that reflects the most recent time-synchronization information received by this PTP Instance and is used to determine the contents of the time-synchronization event message and possibly separate general message that will be sent by this PTP Port.

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

The members of the structure are defined in the following subclauses.

10.2.2.1.2 domainNumber (UInteger8)

This parameter is the domain number of the gPTP domain in which this structure is sent.

NOTE—The domain number member is not essential because the state machines that send and receive this structure are per domain, and each state machine implicitly knows the number of the domain in which it operates.

10.2.2.1.3 followUpCorrectionField (ScaledNs)

The followUpCorrectionField contains the accumulated time since the preciseOriginTimestamp was captured by the Grandmaster PTP Instance. This is equal to the elapsed time, relative to the Grandmaster Clock, between the time the Grandmaster PTP Instance sent the received time-synchronization event message, truncated to the nearest nanosecond, and the time at which that event message was sent by the upstream PTP Instance. The followUpCorrectionField is equal to the value of the followUpCorrectionField member of the most recently received PortSyncSync structure from the PortSync entity of this PTP Port (see 10.2.2.3.5).

10.2.2.1.4 sourcePortIdentity (PortIdentity)

The sourcePortIdentity is the portIdentity of this PTP Port (see 8.5.2).

10.2.2.1.5 logMessageInterval (Integer8)

The logMessageInterval is the value of currentLogSyncInterval for this PTP Port (see 10.7.2.3).

10.2.2.1.6 preciseOriginTimestamp (Timestamp)

The preciseOriginTimestamp is the sourceTime of the ClockMaster entity of the Grandmaster PTP Instance, with any fractional nanoseconds truncated, when the received time-synchronization information was sent by the Grandmaster PTP Instance. The preciseOriginTimestamp is the value of the preciseOriginTimestamp member of the most recently received PortSyncSync structure from the PortSync entity of this PTP Port (see 10.2.2.3.8).

10.2.2.1.7 upstreamTxTime (UScaledNs)

The upstreamTxTime is given by the following equation:

$$\text{upstreamTxTime} = \text{syncEventIngressTimestamp} - \frac{\text{meanLinkDelay}}{\text{neighborRateRatio}}$$

where

syncEventIngressTimestamp	corresponds to the receipt of the time-synchronization information at the slave port of this PTP Instance
meanLinkDelay	is defined in 10.2.5.8
neighborRateRatio	is defined in 10.2.5.7
upstreamTxTime	is the value of the upstreamTxTime member of the most recently received PortSyncSync structure from the PortSync entity of this PTP Port (see 10.2.2.3.9)

10.2.2.1.8 rateRatio (Float64)

The rateRatio is the value of the rateRatio member of the most recently received PortSyncSync structure from the PortSync entity of this PTP Port (see 10.2.2.3.10). It is equal to the ratio of the frequency of the Grandmaster Clock to the frequency of the LocalClock entity of this PTP Instance (see 10.2.8.1.4).

10.2.2.1.9 gmTimeBaseIndicator (UInteger16)

The gmTimeBaseIndicator is the timeBaseIndicator of the ClockSource entity of the current Grandmaster PTP Instance. It is set equal to the gmTimeBaseIndicator of the received time-synchronization information.

The gmTimeBaseIndicator is the value of the gmTimeBaseIndicator member of the most recently received PortSyncSync structure from the PortSync entity of this PTP Port (see 10.2.2.3.11).

10.2.2.1.10 lastGmPhaseChange (ScaledNs)

The lastGmPhaseChange is the time of the current Grandmaster Clock minus the time of the previous Grandmaster Clock, at the time that the current Grandmaster PTP Instance became the Grandmaster PTP Instance, or the step change in the time of the current Grandmaster Clock at the time of the most recent gmTimeBaseIndicator change. It is set equal to the lastGmPhaseChange of the received time-synchronization information. The lastGmPhaseChange is the value of the lastGmPhaseChange member of the most recently received PortSyncSync structure from the PortSync entity of this PTP Port (see 10.2.2.3.12).

10.2.2.1.11 lastGmFreqChange (Float64)

The lastGmFreqChange is the fractional frequency offset of the current Grandmaster Clock relative to the previous Grandmaster Clock, at the time that the current Grandmaster PTP Instance became the Grandmaster PTP Instance, or relative to itself prior to the last change in gmTimeBaseIndicator. It is set equal to the lastGmFreqChange of the received time-synchronization information. The lastGmFreqChange is the value of the lastGmFreqChange member of the most recently received PortSyncSync structure from the PortSync entity of this PTP Port (see 10.2.2.3.13).

10.2.2.2 MDSyncReceive

10.2.2.2.1 General

This structure contains information that is sent by the MD entity of a PTP Port to the PortSync entity of that PTP Port. It provides the PortSync entity with master clock timing information and timestamp of receipt of a time-synchronization event message compensated for propagation time on the upstream link. The information is sent to the PortSync entity upon receipt of time-synchronization information by the MD entity of the PTP Port. The information is in turn provided by the PortSync entity to the SiteSync entity. The information is used by the PortSyncSyncReceive state machine of the PortSync entity to compute the rate ratio of the Grandmaster Clock relative to the local clock and is communicated to the SiteSync entity, and then the SiteSync entity communicates it to the other PortSync entities for use in computing master clock timing information.

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

The members of the structure are defined in the following subclauses.

10.2.2.2.2 domainNumber (UInteger8)

This parameter is the domain number of the gPTP domain in which this structure is sent.

NOTE—The domain number member is not essential because the state machines that send and receive this structure are per domain, and each state machine implicitly knows the number of the domain in which it operates.

10.2.2.2.3 followUpCorrectionField (ScaledNs)

The followUpCorrectionField contains the elapsed time, relative to the Grandmaster Clock, between the time the Grandmaster PTP Instance sent the received time-synchronization information, truncated to the nearest nanosecond, and the time at which this information was sent by the upstream PTP Instance.

NOTE 1—The sum of followUpCorrectionField and preciseOriginTimestamp is the synchronized time that corresponds to the time the most recently received time-synchronization event message was sent by the upstream PTP Instance.

NOTE 2—For a medium that uses separate event and general messages (for example, full-duplex point-to-point media described in 11), the event message corresponding to the most recently received network synchronization information is the event message that corresponds to the most recently received general message. For a medium that places synchronization information based on the event message timestamp in the next event message (for example, IEEE 802.11 media described in Clause 12), the event message corresponding to the most recently received network synchronization information is the previous event message; in this case, the time-synchronization information in the current event message refers to the previous event message.

10.2.2.2.4 sourcePortIdentity (PortIdentity)

The sourcePortIdentity is the value of the sourcePortIdentity of the time-synchronization event message received by this PTP Port. It is the portIdentity of the upstream MasterPort that sent the event message.

10.2.2.2.5 logMessageInterval (Integer8)

The logMessageInterval is the value of the logMessageInterval of the time-synchronization event message received by this PTP Port. It is the currentLogSyncInterval (see 10.7.2.3) of the upstream MasterPort that sent the event message.

10.2.2.2.6 preciseOriginTimestamp (Timestamp)

The preciseOriginTimestamp is the sourceTime of the ClockMaster entity of the Grandmaster PTP Instance, with any fractional nanoseconds truncated, when the time-synchronization event message was sent by the Grandmaster PTP Instance.

10.2.2.2.7 upstreamTxTime (UScaledNs)

The upstreamTxTime is given by the following equation:

$$\text{upstreamTxTime} = \text{syncEventIngressTimestamp} - \frac{\text{meanLinkDelay}}{\text{neighborRateRatio}}$$

where

syncEventIngressTimestamp	corresponds to the receipt of the time-synchronization information at the slave port of this PTP Instance (i.e., at this PTP Port)
meanLinkDelay	is defined in 10.2.5.8
neighborRateRatio	is defined in 10.2.5.7

10.2.2.2.8 rateRatio (Float64)

The rateRatio is the value of rateRatio of the received time-synchronization information. It is equal to the ratio of the frequency of the Grandmaster Clock to the frequency of the LocalClock entity of the PTP

Instance at the other end of the link attached to this PTP Port, i.e., the PTP Instance that sent the most recently received time-synchronization event message (see 10.2.8.1.4).

10.2.2.2.9 gmTimeBaseIndicator (UInteger16)

The gmTimeBaseIndicator is the timeBaseIndicator of the ClockSource entity of the current Grandmaster PTP Instance. It is set equal to the gmTimeBaseIndicator of the received time-synchronization information.

10.2.2.2.10 lastGmPhaseChange (ScaledNs)

The lastGmPhaseChange is the time of the current Grandmaster Clock minus the time of the previous Grandmaster Clock, at the time that the current Grandmaster PTP Instance became the Grandmaster PTP Instance, or the step change in the time of the current Grandmaster Clock at the time of the most recent gmTimeBaseIndicator change. It is set equal to the lastGmPhaseChange of the received time-synchronization information.

10.2.2.2.11 lastGmFreqChange (Float64)

The lastGmFreqChange is the fractional frequency offset of the current Grandmaster Clock relative to the previous Grandmaster Clock, at the time that the current Grandmaster PTP Instance became the Grandmaster PTP Instance, or relative to itself prior to the last change in gmTimeBaseIndicator. It is set equal to the lastGmFreqChange of the received time-synchronization information.

10.2.2.3 PortSyncSync

10.2.2.3.1 General

This structure is sent by the PortSync and ClockMaster entities to the SiteSync entity and also from the SiteSync entity to the PortSync and ClockSlave entities.

When sent from the PortSync or ClockMaster entity, it provides the SiteSync entity with master clock timing information, timestamp of receipt of a time-synchronization event message compensated for propagation time on the upstream link, and the time at which sync receipt timeout occurs if a subsequent Sync message is not received by then. The information is used by the SiteSync entity to compute the rate ratio of the Grandmaster Clock relative to the local clock and is communicated to the other PortSync entities for use in computing master clock timing information.

When sent from the SiteSync entity to the PortSync or ClockSlave entity, the structure contains information needed to compute the synchronization information that will be included in respective fields of the time-synchronization event and general messages that will be sent and also to compute the synchronized time that the ClockSlave entity will supply to the ClockTarget entity.

```
PortSyncSync    {
        domainNumber,
        localPortNumber,
        syncReceiptTimeoutTime,
        followUpCorrectionField,
        sourcePortIdentity,
        logMessageInterval,
        preciseOriginTimestamp,
        upstreamTxTime,
        rateRatio,
        gmTimeBaseIndicator,
```

```
        lastGmPhaseChange,
        lastGmFreqChange
}
```

The parameters of the PortSyncSync structure are defined in the following subclauses for when the structure is sent from the PortSync or ClockMaster entity to the SiteSync entity. If the structure is sent from the SiteSync entity to the PortSync or ClockSlave entity, the member values are copied from the most recently received PortSyncSync structure where the PTP Port that received this structure has PTP Port state of SlavePort.

10.2.2.3.2 domainNumber (UInteger8)

This parameter is the domain number of the gPTP domain in which this structure is sent.

NOTE—The domain number member is not essential because the state machines that send and receive this structure are per domain, and each state machine implicitly knows the number of the domain in which it operates.

10.2.2.3.3 localPortNumber (UInteger16)

If the structure is sent by a PortSync entity, the localPortNumber is the port number of the PTP Port whose PortSync entity sent this structure. If the structure is sent by a ClockMaster entity, the localPortNumber is zero.

10.2.2.3.4 syncReceiptTimeoutTime (UScaledNs)

If the structure is sent by a PortSync entity, the syncReceiptTimeoutTime is the value of the local time (i.e., the free-running, local clock time) at which sync receipt timeout occurs if a subsequent time-synchronization event message is not received by that time. If the structure is sent by a ClockMaster entity, the syncReceiptTimeoutTime is $\mathrm{FFFFFFFFFFFFFFFF}_{16}$ [see item h) in 10.2.9.2.1].

10.2.2.3.5 followUpCorrectionField (ScaledNs)

If the structure is sent by a PortSync entity, the followUpCorrectionField is the value of the followUpcorrectionField member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.2). If the structure is sent by a ClockMaster entity, the followUpCorrectionField is the sub-nanosecond portion of the ClockMaster time.

10.2.2.3.6 sourcePortIdentity (PortIdentity)

If the structure is sent by a PortSync entity, the sourcePortIdentity is the value of the sourcePortIdentity member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.4). If the structure is sent by a ClockMaster entity, the clockIdentity member of the sourcePortIdentity is the clockIdentity of this PTP Instance, and the portNumber member of the sourcePortIdentity is 0.

10.2.2.3.7 logMessageInterval (Integer8)

If the structure is sent by a PortSync entity, the logMessageInterval is the value of the logMessageInterval member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.5). If the structure is sent by a ClockMaster entity, the logMessageInterval is the value of clockMasterLogSyncInterval (see 10.7.2.4).

10.2.2.3.8 preciseOriginTimestamp (Timestamp)

If the structure is sent by a PortSync entity, the preciseOriginTimestamp is the value of the preciseOriginTimestamp member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.6). If the structure is sent by a ClockMaster entity, the preciseOriginTimestamp is the ClockMaster time truncated to the next lower nanosecond.

10.2.2.3.9 upstreamTxTime (UScaledNs)

If the structure is sent by a PortSync entity, the upstreamTxTime is the value of the upstreamTxTime member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.7). If the structure is sent by a ClockMaster entity, the upstreamTxTime is the local clock time corresponding to the ClockMaster time.

10.2.2.3.10 rateRatio (Float64)

If the structure is sent by a PortSync entity, the rateRatio is the value of the rateRatio member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.8). It is equal to the ratio of the frequency of the Grandmaster Clock to the frequency of the LocalClock entity of the PTP Instance at the other end of the link attached to this PTP Port, i.e., the PTP Instance that sent the most recently-received time-synchronization event message (see 10.2.8.1.4). If the structure is sent by a ClockMaster entity, the rateRatio is equal to gmRateRatio (see 10.2.4.14).

10.2.2.3.11 gmTimeBaseIndicator (UInteger16)

If the structure is sent by a PortSync entity, the gmTimeBaseIndicator is the value of the gmTimeBaseIndicator member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.9). If the structure is sent by a ClockMaster entity, the gmTimeBaseIndicator is equal to clockSourceTimeBaseIndicator (see 10.2.4.8).

10.2.2.3.12 lastGmPhaseChange (ScaledNs)

If the structure is sent by a PortSync entity, the lastGmPhaseChange is the value of the lastGmPhaseChange member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.9). If the structure is sent by a ClockMaster entity, the lastGmPhaseChange is equal to clockSourcePhaseOffset (see 10.2.4.7).

10.2.2.3.13 lastGmFreqChange (Float64)

If the structure is sent by a PortSync entity, the lastGmFreqChange is the value of the lastGmFreqChange member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.9). If the structure is sent by a ClockMaster entity, the lastGmFreqChange is equal to clockSourceFreqOffset (see 10.2.4.6).

10.2.3 Overview of global variables used by time synchronization state machines

Subclauses 10.2.4 and 10.2.5 define global variables used by time synchronization state machines whose scopes are as follows:

- Per PTP Instance (i.e., per domain)
- Per PTP Instance, per PTP Port
- Instance used by the Common Mean Link Delay Service (CMLDS) (see 11.2.17) (i.e., variable is common across all LinkPorts)
- Instance used by CMLDS, per LinkPort

Table 10-1 summarizes the scope of each global variable of 10.2.4 and 10.2.5.

Table 10-1—Summary of scope of global variables used by time synchronization state machines (see 10.2.4 and 10.2.5)

Variable name	Subclause of definition	Per PTP Instance (i.e., per domain)	Per PTP Instance, per PTP Port	Instance used by CMLDS (i.e., variable is common across all LinkPorts)	Instance used by CMLDS, per LinkPort
BEGIN	10.2.4.1	Yes	No	Yes	No
clockMasterSyncInterval	10.2.4.2	Yes	No	No	No
clockSlaveTime	10.2.4.3	Yes	No	No	No
syncReceiptTime	10.2.4.4	Yes	No	No	No
syncReceiptLocalTime	10.2.4.5	Yes	No	No	No
clockSourceFreqOffset	10.2.4.6	Yes	No	No	No
clockSourcePhaseOffset	10.2.4.7	Yes	No	No	No
clockSourceTimeBaseIndicator	10.2.4.8	Yes	No	No	No
clockSourceTimeBaseIndicatorOld	10.2.4.9	Yes	No	No	No
clockSourceLastGmPhaseChange	10.2.4.10	Yes	No	No	No
clockSourceLastGmFreqChange	10.2.4.11	Yes	No	No	No
currentTime	10.2.4.12	Yes	No	No	No
gmPresent	10.2.4.13	Yes	No	No	No
gmRateRatio	10.2.4.14	Yes	No	No	No
gmTimeBaseIndicator	10.2.4.15	Yes	No	No	No
lastGmPhaseChange	10.2.4.16	Yes	No	No	No
lastGmFreqChange	10.2.4.17	Yes	No	No	No
localClockTickInterval	10.2.4.18	Yes	No	No	No
localTime	10.2.4.19	Yes	No	No	No
selectedState	10.2.4.20	Yes	No	No	No
masterTime	10.2.4.21	Yes	No	No	No
thisClock	10.2.4.22	Yes	No	Yes	No
parentLogSyncInterval	10.2.4.23	Yes	No	No	No
instanceEnable	10.2.4.24	Yes	No	No	No
syncReceiptTimeoutTime	10.2.4.25	Yes	No	No	No
asCapable	10.2.5.1	No	Yes	No	No
asymmetryMeasurementMode	10.2.5.2	No	Yes[a]	No	Yes
syncReceiptTimeoutTimeInterval	10.2.5.3	No	Yes	No	No
currentLogSyncInterval	10.2.5.4	No	Yes	No	No
initialLogSyncInterval	10.2.5.5	No	Yes	No	No
syncInterval	10.2.5.6	No	Yes	No	No

Table 10-1—Summary of scope of global variables used by time synchronization state machines (see 10.2.4 and 10.2.5) *(continued)*

Variable name	Subclause of definition	Per PTP Instance (i.e., per domain)	Per PTP Instance, per PTP Port	Instance used by CMLDS (i.e., variable is common across all LinkPorts)	Instance used by CMLDS, per LinkPort
neighborRateRatio	10.2.5.7	No	Yes[a]	No	Yes
meanLinkDelay	10.2.5.8	No	Yes[a]	No	Yes
delayAsymmetry	10.2.5.9	No	Yes[a]	No	Yes
computeNeighborRateRatio	10.2.5.10	No	Yes[a]	No	Yes
computeMeanLinkDelay	10.2.5.11	No	Yes[a]	No	Yes
portOper[b]	10.2.5.12	No	Yes	No	Yes
ptpPortEnabled	10.2.5.13	No	Yes	No	No
thisPort	10.2.5.14	No	Yes	No	Yes
syncLocked	10.2.5.15	No	Yes	No	No
neighborGptpCapable	10.2.5.16	No	Yes	No	No
syncSlowdown	10.2.5.17	No	Yes	No	No
oldSyncInterval	10.2.5.18	No	Yes	No	No
gPtpCapableMessageSlowdown	10.2.5.19	No	Yes	No	No
gPtpCapableMessageInterval	10.2.5.20	No	Yes	No	No
oldGptpCapableMessageInterval	10.2.5.21	No	Yes	No	No
currentLogGptpCapableMessageInterval	10.2.5.22	No	Yes	No	No
initialLogGptpCapableMessageInterval	10.2.5.23	No	Yes	No	No

[a] The instance of this variable that is per PTP Instance, per PTP Port exists only for domain 0.
[b] There is one instance of this variable per physical port, which is accessible by all PTP Ports and LinkPorts associated with the physical port.

10.2.4 Per PTP Instance global variables

10.2.4.1 BEGIN: A Boolean controlled by the system initialization. If BEGIN is true, all state machines, including per-PTP Port state machines, continuously execute their initial state. See Annex E of IEEE Std 802.1Q-2018.

10.2.4.2 clockMasterSyncInterval: A variable containing the mean time interval between successive messages providing time-synchronization information by the ClockMaster entity to the SiteSync entity. This value is given by $1000000000 \times 2^{\mathrm{clockMasterLogSyncInterval}}$ ns, where clockMasterLogSyncInterval is the logarithm to base 2 of the mean time between the successive providing of time-synchronization information by the ClockMaster entity (see 10.7.2.4). The data type for clockMasterSyncInterval is UScaledNs.

10.2.4.3 clockSlaveTime: The synchronized time maintained, at the slave, at the granularity of the LocalClock entity [i.e., a new value is computed every localClockTickInterval (see 10.2.4.18) by the ClockSlave entity]. The data type for clockSlaveTime is ExtendedTimestamp.

10.2.4.4 syncReceiptTime: The synchronized time computed by the ClockSlave entity at the instant time-synchronization information, contained in a PortSyncSync structure, is received. The data type for syncReceiptTime is ExtendedTimestamp.

10.2.4.5 syncReceiptLocalTime: The value of currentTime (i.e., the time relative to the LocalClock entity) corresponding to syncReceiptTime. The data type for syncReceiptLocalTime is UScaledNs.

10.2.4.6 clockSourceFreqOffset: The fractional frequency offset of the ClockSource entity frequency relative to the current Grandmaster Clock frequency. The data type for clockSourceFreqOffset is Float64.

10.2.4.7 clockSourcePhaseOffset: The time provided by the ClockSource entity, minus the synchronized time. The data type for clockSourcePhaseOffset is ScaledNs.

10.2.4.8 clockSourceTimeBaseIndicator: A global variable that is set equal to the timeBaseIndicator parameter of the ClockSourceTime.invoke application interface function (see 9.2.2.3), by the ClockMaster entity. The parameter timeBaseIndicator of ClockSourceTime.invoke is set by the ClockSource entity and is changed by that entity whenever the time base changes. The data type for clockSourceTimeBaseIndicator is UInteger16.

10.2.4.9 clockSourceTimeBaseIndicatorOld: A global variable that is set equal to the previous value of clockSourceTimeBaseIndicator. The data type for clockSourceTimeBaseIndicatorOld is UInteger16.

10.2.4.10 clockSourceLastGmPhaseChange: A global variable that is set equal to the lastGmPhaseChange parameter of the ClockSourceTime.invoke application interface function (see 9.2.2.4). That parameter is set by the ClockSource entity and is changed by that entity whenever the time base changes. The data type for clockSourceLastGmPhaseChange is ScaledNs.

10.2.4.11 clockSourceLastGmFreqChange: A global variable that is set equal to the lastGmFreqChange parameter of the ClockSourceTime.invoke application interface function (see 9.2.2.5). That parameter is set by the ClockSource entity and is changed by that entity whenever the time base changes. The data type for clockSourceLastGmFreqChange is Float64.

10.2.4.12 currentTime: The current value of time relative to the LocalClock entity clock. The data type for currentTime is UScaledNs.

10.2.4.13 gmPresent: A Boolean that indicates whether a grandmaster-capable PTP Instance is present in the domain. If TRUE, a grandmaster-capable PTP Instance is present; if FALSE, a grandmaster-capable PTP Instance is not present.

10.2.4.14 gmRateRatio: The measured ratio of the frequency of the ClockSource entity to the frequency of the LocalClock entity. The data type for gmRateRatio is Float64.

10.2.4.15 gmTimeBaseIndicator: The most recent value of gmTimeBaseIndicator provided to the ClockSlaveSync state machine via a PortSyncSync structure. The data type for gmTimeBaseIndicator is UInteger16.

10.2.4.16 lastGmPhaseChange: The most recent value of lastGmPhaseChange provided to the ClockSlaveSync state machine via a PortSyncSync structure. The data type for lastGmPhaseChange is ScaledNs.

10.2.4.17 lastGmFreqChange: The most recent value of lastGmFreqChange provided to the ClockSlaveSync state machine via a PortSyncSync structure. The data type for lastGmFreqChange is Float64.

10.2.4.18 localClockTickInterval: The time interval between two successive significant instants (i.e., "ticks") of the LocalClock entity. The data type for localClockTickInterval is TimeInterval.

10.2.4.19 localTime: The value of currentTime when the most recent ClockSourceTime.invoke function (see 9.2) was received from the ClockSource entity, or when the LocalClock entity most recently updated its time. The data type for localTime is UScaledNs.

10.2.4.20 selectedState: An Enumeration2 array of length numberPorts+1 (see 8.6.2.8). selectedState[j] is set equal to the PTP Port State (see Table 10-2) of the PTP Port whose portNumber is j.

10.2.4.21 masterTime: The time maintained by the ClockMaster entity, based on information received from the ClockSource and LocalClock entities. The data type for masterTime is ExtendedTimestamp.

10.2.4.22 thisClock: The clockIdentity of the current PTP Instance. The data type for thisClock is ClockIdentity.

10.2.4.23 parentLogSyncInterval: The most recent logMessageInterval value received on the slave port. If this PTP Instance is the Grandmaster PTP Instance, then this is the clockMasterLogSyncInterval (see 10.7.2.4). The data type for parentLogSyncInterval is Integer8.

10.2.4.24 instanceEnable: A per-domain Boolean used to enable gPTP on all ports that are enabled for that domain (i.e., ports for which portOper and ptpPortEnabled are both TRUE). Setting instanceEnable to FALSE causes all per-domain state machines to go to the initial state.

NOTE—instanceEnable has no effect on the operation of the MDPdelayReq (see 11.2.19) and MDPdelayResp (see 11.2.20) state machines because those state machines are not per domain (i.e., there is a single instance of each of those state machines, per link, for all domains).

10.2.4.25 syncReceiptTimeoutTime: The value of the syncReceiptTimeoutTime member of the most recently received PortSyncSync structure. The data type for syncReceiptTimeoutTime is UScaledNs.

10.2.5 Per-port global variables

10.2.5.1 asCapable: A Boolean that is TRUE if and only if it is determined that this PTP Instance and the PTP Instance at the other end of the link attached to this PTP Port can interoperate with each other via the IEEE 802.1AS protocol. As a result,

a) This PTP Instance is capable of executing the IEEE 802.1AS protocol,
b) The PTP Instance at the other end of the link is capable of executing the IEEE 802.1AS protocol, and
c) There are no non-IEEE-802.1AS systems in between this PTP Instance and the PTP Instance at the other end of the link that introduce sufficient impairments that the end-to-end time-synchronization performance of B.3 cannot be met.

The determination of asCapable is different for each medium and is described in the respective media-dependent clauses.

There is one instance of this variable per PTP Instance (i.e., per domain), per PTP Port.

NOTE—The per-port global variable asCapableAcrossDomains (see 11.2.13.12) is common across, and accessible by, all the domains. It is computed by the MDPdelayReq state machine (see 11.2.19). For full-duplex point-to-point links (see 11), asCapableAcrossDomains is used when setting the instance of asCapable for each domain (for the link in question).

10.2.5.2 asymmetryMeasurementMode: A Boolean that contains the value of the managed object asymmetryMeasurementMode (see 14.8.45). For full-duplex IEEE 802.3 media, the value is TRUE if an asymmetry measurement is being performed for the link attached to this port and FALSE otherwise. For all other media, the value is FALSE. There is one instance of this variable for all the domains, i.e., all the PTP Instances (per port). The variable is accessible by all the domains.

10.2.5.3 syncReceiptTimeoutTimeInterval: The time interval after which sync receipt timeout occurs if time-synchronization information has not been received during the interval. The value of syncReceiptTimeoutTimeInterval is equal to syncReceiptTimeout (see 10.7.3.1) multiplied by the syncInterval (see 10.2.5.6) for the PTP Port at the other end of the link to which this PTP Port is attached. The value of syncInterval for the PTP Port at the other end of the link is computed from logMessageInterval of the received Sync message (see 10.6.2.2.14). The data type for syncReceiptTimeoutTimeInterval is UScaledNs.

10.2.5.4 currentLogSyncInterval: The current value of the logarithm to base 2 of the mean time interval, in seconds, between the sending of successive time-synchronization event messages (see 10.7.2.3). This value is set in the SyncIntervalSetting state machine (see 10.3.18). The data type for currentLogSyncInterval is Integer8.

10.2.5.5 initialLogSyncInterval: The initial value of the logarithm to base 2 of the mean time interval, in seconds, between the sending of successive time-synchronization event messages (see 10.7.2.3). The data type for initialLogSyncInterval is Integer8.

10.2.5.6 syncInterval: A variable containing the mean time-synchronization event message transmission interval for the PTP Port. This value is set in the SyncIntervalSetting state machine (see 10.3.18). The data type for syncInterval is UScaledNs.

10.2.5.7 neighborRateRatio: The measured ratio of the frequency of the LocalClock entity of the time-aware system at the other end of the link attached to this port, to the frequency of the LocalClock entity of this time-aware system. The data type for neighborRateRatio is Float64. There is one instance of this variable for all the domains, i.e., all the PTP Instances (per port). The variable is accessible by all the domains.

10.2.5.8 meanLinkDelay: The measured mean propagation delay (see 8.3) on the link attached to this port, relative to the LocalClock entity of the time-aware system at the other end of the link (i.e., expressed in the time base of the time-aware system at the other end of the link). The data type for meanLinkDelay is UScaledNs. There is one instance of this variable for all the domains, i.e., all the PTP Instances (per port). The variable is accessible by all the domains.

NOTE—The variable meanLinkDelay was named neighborPropDelay in the 2011 edition of this standard.

10.2.5.9 delayAsymmetry: The asymmetry in the propagation delay on the link attached to this port. If propagation delay asymmetry is not modeled, then delayAsymmetry is zero. The data type for delayAsymmetry is ScaledNs. There is one instance of this variable for CMLDS (see 11.2.17), and there is also one instance of this variable for each domain that uses the instance-specific peer-to-peer delay mechanism. The instance of this variable for CMLDS is relative to the local clock. The instance of this variable for a domain that uses the instance-specific peer-to-peer delay mechanism is relative to the grandmaser time base.

10.2.5.10 computeNeighborRateRatio: A Boolean, set by the LinkDelayIntervalSetting state machine (see 11.2.21), that indicates whether neighborRateRatio is to be computed by this port. There is one instance of this variable for all the domains, i.e., all the PTP Instances (per port). The variable is accessible by all the domains.

10.2.5.11 computeMeanLinkDelay: A Boolean, set by the LinkDelayIntervalSetting state machine (see 11.2.21), that indicates whether meanLinkDelay is to be computed by this port. There is one instance of this variable for all the domains, i.e., all the PTP Instances (per port). The variable is accessible by all the domains.

10.2.5.12 portOper: A Boolean that is TRUE if and only if the port is up and able to send and receive messages. There is one instance of this variable for all the domains, i.e., all the PTP Instances (per port). The variable is accessible by all the domains. The term *port* in this definition is a physical port.

NOTE 1—portOper is an indicator, and not a control, and reflects the operational status of the underlying medium. It is not administratively set by gPTP.

NOTE 2—The variable portOper corresponds to the variable portEnabled in the 2011 edition of this standard. The change is reflected in many state machines.

NOTE 3—portOper is the same as MAC_Operational (see IEEE Std 802.1AC-2016).

NOTE 4—When portOper is referenced for a logical port, the reference is to portOper for the physical port that corresponds to the logical port.

10.2.5.13 ptpPortEnabled: A Boolean that is administratively set to TRUE if time-synchronization is to be enabled on this PTP Port.

NOTE 1—It is expected that the value of ptpPortEnabled will be set via the management interface (see 14.8.4). A PTP Port can be enabled for data transport but not for synchronization transport.

NOTE 2—The variable ptpPortEnabled was named pttPortEnabled in the 2011 edition of this standard. Only the name of this variable has changed; the definition and function of this variable are the same as in the 2011 edition. The name change is reflected in many state machines.

10.2.5.14 thisPort: The portNumber of the current port. The data type for thisPort is UInteger16.

10.2.5.15 syncLocked: A Boolean, set by the PortSyncSyncSend state machine (see 10.2.12.3), that indicates that this PTP Port, when operating as a master port, shall transmit a Sync as soon as possible after the slave port received a Sync (ignoring syncInterval). If FALSE, the PTP Port shall use the timing set by syncInterval.

10.2.5.16 neighborGptpCapable: A Boolean, set by the GptpCapableReceive state machine (see 10.4.2), that indicates that the neighbor of this PTP Port (i.e, the PTP Port at the other end of the link attached to this PTP Port) is capable of invoking gPTP.

10.2.5.17 syncSlowdown: A Boolean that is set to TRUE if the SyncIntervalSetting state machine (see Figure 10-20 in 10.3.18.3) receives a TLV that requests a larger sync interval (see 10.7.2.3) and FALSE otherwise. When syncSlowdown is set to TRUE, the PortSyncSyncSend state machine (see Figure 10-8) continues to send time synchronization event messages (see 11.4.3, 12.1, 12.2, and 13.3.1) at the old (i.e., faster) rate until the number of time synchronization event messages equal to syncReceiptTimeout (see 10.7.3.1) have been sent, but with the respective time synchronization event message transmission interval field (see 11.4.2.9, 12.7, Figure 12-8, and 13.3.1.2.10) of the time synchronization event message set equal to the new sync interval (i.e., corresponding to the slower rate). After syncReceiptTimeout Sync messages have been sent, subsequent time synchronization event messages are sent at the new (i.e., slower) rate and with the respective time synchronization event message transmission interval field of the time synchronization event message set to the new sync interval. When syncSlowdown is set to FALSE, the PortSyncSyncSend state machine immediately sends time synchronization event messages at the new (i.e., slower) rate.

NOTE—If a receiver of time synchronization event messages (see 11.4.3, 12.1, 12.2, and 13.3.1) requests a slower rate, the receiver will continue to use the upstream sync interval value, which it obtains from the respective time synchronization event message transmission interval field (see 11.4.2.9, 12.7, Figure 12-8, and 13.3.1.2.10) of the received time synchronization event message, until it receives a time synchronzation event message where that value has changed. If, immediately after requesting a slower time synchronization event message rate, up to syncReceiptTimeout consecutive time synchronization event messages sent to the receiver are lost, sync receipt timeout could occur if the sender had changed to the slower rate immediately. Delaying the slowing down of the sending rate of time synchronization event messages for syncReceiptTimeout messages prevents this timeout from happening.

10.2.5.18 oldSyncInterval: The saved value of the previous sync interval, when a new time-synchronization event message transmission interval is requested via a Signaling message that contains a message interval request TLV. The data type for oldSyncInterval is UScaledNs.

10.2.5.19 gPtpCapableMessageSlowdown: A Boolean that is set to TRUE if the GptpCapableIntervalSetting state machine (see Figure 10-19 in 10.3.17.3) receives a TLV that requests a larger gPTP-capable message interval (see 10.7.2.5) and FALSE otherwise. When gPtpCapableMessageSlowdown is set to TRUE, the GptpCapableTransmit state machine (see Figure 10-21 in 10.4.1.3) continues to send Signaling messages containing the gPTP-capable TLV at the old (i.e., faster) rate until a number of Signaling messages containing the gPTP-capable TLV, equal to gPtpCapableReceiptTimeout (see 10.7.3.3), have been sent, but with the logGptpCapableMessageInterval field of the gPTP-capable TLV (see 10.6.4.5.6) set equal to the new gPTP-capable message interval (i.e., corresponding to the slower rate). After gPtpCapableReceiptTimeout Signaling messages containing the gPTP-capable TLV have been sent, subsequent such Signaling messages are sent at the new (i.e., slower) rate and with the logGptpCapableMessageInterval field of the gPTP-capable TLV set to the new gPTP-capable message interval. When gPtpCapableSlowdown is set to FALSE, the GptpCapableTransmit state machine immediately sends Signaling messages containing the gPTP-capable TLV at the new (i.e., slower) rate.

NOTE—If a receiver of Signaling messages containing the gPTP-capable TLV requests a slower rate, the receiver will continue to use the old gPTP-capable message interval value in determining, via the GptpCapableReceive state machine (see 10.4.2), if its neighbor is no longer capable of invoking gPTP, until it has received gPtpCapableReceiptTimeout such Signaling messages. If, immediately after requesting a slower rate, up to gPtpCapableReceiptTimeout consecutive Signaling messages, containing the gPTP-capable TLV, sent to the receiver are lost, a declaration that the sender is no longer capable of invoking gPTP could occur if the sender had changed to the slower rate immediately. Delaying the slowing down of the sending rate of Signaling messages containing the gPTP-capable TLV for gPtpCapableReceiptTimeout messages prevents this timeout from happening.

10.2.5.20 gPtpCapableMessageInterval: A variable whose value is the mean time, in seconds, between the sending of successive Signaling messages that carry the gPTP-capable TLV (see 10.7.2.5 and 10.6.4.4). The data type for gPtpCapableMessageInterval is UScaledNs.

10.2.5.21 oldGptpCapableMessageInterval: The saved value of the previous interval between the sending of successive Signaling messages that carry the gPTP-capable TLV (see 10.2.5.20), when a new such interval is requested via a Signaling message that contains a gPTP-capable message interval request TLV. The data type for oldGptpCapableMessageInterval is UScaledNs.

10.2.5.22 currentLogGptpCapableMessageInterval: The current value of the logarithm to base 2 of the mean time interval, in seconds, between the sending of successive Signaling messages that carry the gPTP-capable TLV (see 10.6.4.4). This value is set in the GptpCapableIntervalSetting state machine (see 10.4.3). The data type for currentGptpCapableMessageInterval is Integer8.

10.2.5.23 initialLogGptpCapableMessageInterval: The initial value of the logarithm to base 2 of the mean time interval, in seconds, between the sending of successive Signaling messages that carry the gPTP-capable TLV (see 10.6.4.4). The data type for initialLogGptpCapableMessageInterval is Integer8.

10.2.6 Function used by multiple state machines

10.2.6.1 random(): Returns a uniformly-distributed pseudo-random number whose data type is UInteger16 (i.e., the function returns a uniformly distributed, pseudo-random integer in the range $[0, 2^{16} – 1]$).

10.2.7 SiteSyncSync state machine

10.2.7.1 State machine variables

The following variables are used in the state diagram in Figure 10-3 (in 10.2.7.3):

10.2.7.1.1 rcvdPSSyncSSS: A Boolean variable that notifies the current state machine when a PortSyncSync structure (see 10.2.2.3) is received from the PortSyncSyncReceive state machine of a PortSync entity or from the ClockMasterSyncSend state machine of the ClockMaster entity. This variable is reset by this state machine.

10.2.7.1.2 rcvdPSSyncPtrSSS: A pointer to the received PortSyncSync structure indicated by rcvdPSSyncSSS.

10.2.7.1.3 txPSSyncPtrSSS: A pointer to the PortSyncSync structure transmitted by the state machine.

10.2.7.2 State machine functions

10.2.7.2.1 setPSSyncSend (rcvdPSSyncPtrSSS): Creates a PortSyncSync structure to be transmitted, and returns a pointer to this structure. The members are copied from the received PortSyncSync structure pointed to by rcvdPSSyncPtrSSS.

10.2.7.2.2 txPSSync (txPSSyncPtrSSS): Transmits a copy of the PortSyncSync structure pointed to by txPSSyncPtrSSS to the PortSyncSyncSend state machine of each PortSync entity and the ClockSlaveSync state machine of the ClockSlave entity of this PTP Instance.

10.2.7.3 State diagram

The SiteSyncSync state machine shall implement the function specified by the state diagram in Figure 10-3, the local variables specified in 10.2.7.1, the functions specified in 10.2.7.2, the structure specified in 10.2.2.3, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine receives time-synchronization information, accumulated rateRatio, and syncReceiptTimeoutTime from the PortSync entity (PortSyncSyncReceive state machine) of the current slave port or from the ClockMaster entity (ClockMasterSyncSend state machine). If the information was sent by a PortSync entity, the state machine also receives the portIdentity of the PTP Port on the upstream PTP Instance that sent the information to this PTP Instance (if the information was sent by the ClockMaster entity, the portIdentity is that of the ClockMaster entity, i.e., it has clockIdentity equal to the clockIdentity of this PTP Instance and portNumber 0). The state machine sends a PortSyncSync structure to the PortSync entities of all the ports and to the ClockSlave entity.

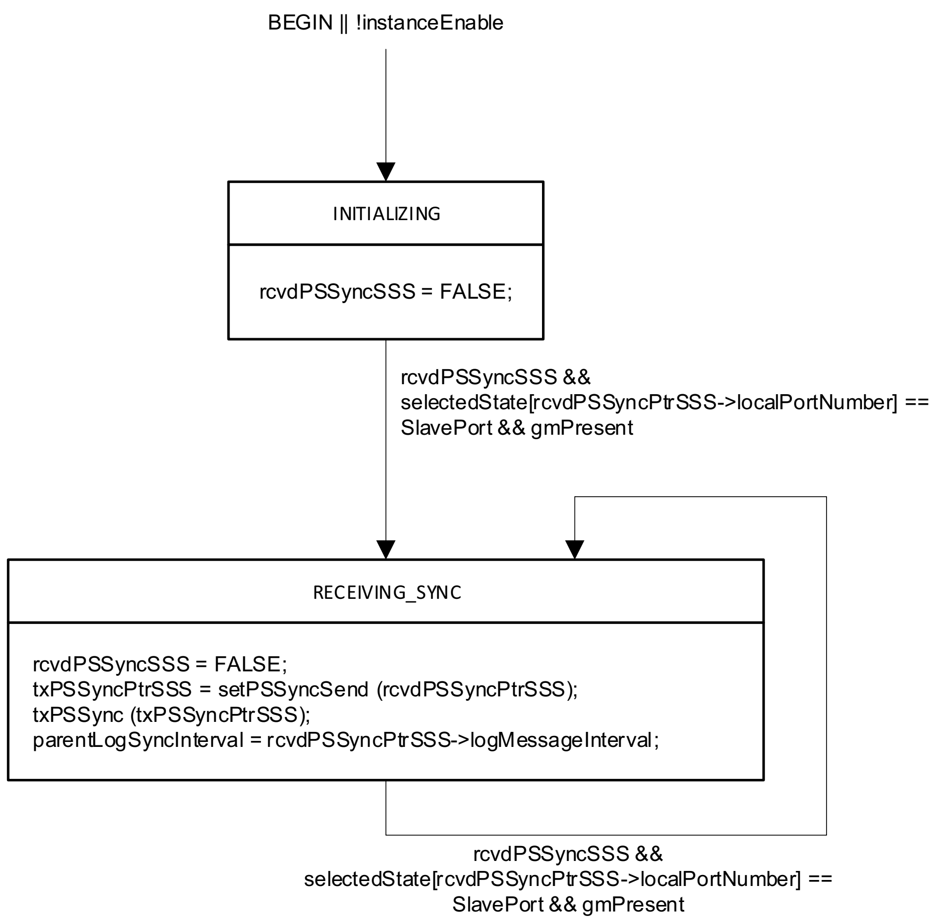


Figure 10-3—SiteSyncSync state machine

10.2.8 PortSyncSyncReceive state machine

10.2.8.1 State machine variables

The following variables are used in the state diagram in Figure 10-4 (in 10.2.8.3):

10.2.8.1.1 rcvdMDSyncPSSR: A Boolean variable that notifies the current state machine when an MDSyncReceive structure is received from an MD entity of the same PTP Port (see 10.2.2.1). This variable is reset by this state machine.

10.2.8.1.2 rcvdMDSyncPtrPSSR: A pointer to the received MDSyncReceive structure indicated by rcvdMDSyncPSSR.

10.2.8.1.3 txPSSyncPtrPSSR: A pointer to the PortSyncSync structure transmitted by the state machine.

10.2.8.1.4 rateRatio: A Float64 variable that holds the ratio of the frequency of the Grandmaster Clock to the frequency of the LocalClock entity. This frequency ratio is computed by:

a) Measuring the ratio of the Grandmaster Clock frequency to the LocalClock frequency at the Grandmaster PTP Instance and initializing rateRatio to this value in the ClockMasterSend state machine of the Grandmaster PTP Instance and
b) Accumulating, in the PortSyncSyncReceive state machine of each PTP Instance, the frequency offset of the LocalClock entity of the PTP Instance at the remote end of the link attached to that PTP Port to the frequency of the LocalClock entity of this PTP Instance.

10.2.8.2 State machine functions

10.2.8.2.1 setPSSyncPSSR (rcvdMDSyncPtrPSSR syncReceiptTimeoutTimeInterval, rateRatio): Creates a PortSyncSync structure to be transmitted, and returns a pointer to this structure. The members are set as follows:

a) localPortNumber is set equal to thisPort.
b) domainNumber, followUpCorrectionField, sourcePortIdentity, logMessageInterval, preciseOriginTimestamp, and upstreamTxTime are copied from the received MDSyncReceive structure pointed to by rcvdMDSyncPtrPSSR.
c) syncReceiptTimeoutTime is set equal to currentTime plus syncReceiptTimeoutTimeInterval (see 10.2.5.3).
d) The function argument rateRatio is set equal to the local variable rateRatio (computed just prior to invoking setPSSyncPSSR (see Figure 10-4)). The rateRatio member of the PortSyncSync structure is then set equal to the function argument rateRatio.

10.2.8.2.2 txPSSyncPSSR (txPSSyncPtrPSSR): Transmits a copy of the PortSyncSync structure pointed to by txPSSyncPtrPSSR to the SiteSyncSync state machine of this PTP Instance.

10.2.8.3 State diagram

The PortSyncSyncReceive state machine shall implement the function specified by the state diagram in Figure 10-4, the local variables specified in 10.2.8.1, the functions specified in 10.2.8.2, the structures specified in 10.2.2.1 and 10.2.2.3, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine receives time-synchronization information, accumulated rateRatio, and syncReceiptTimeoutTime from the MD entity of the same PTP Port. The state machine adds, to rateRatio, the fractional frequency offset of the LocalClock entity relative to the LocalClock entity of the upstream PTP Instance at the remote end of the link attached to this PTP Port. The state machine computes syncReceiptTimeoutTime. The state machine sends this information to the SiteSync entity (SiteSyncSync state machine).

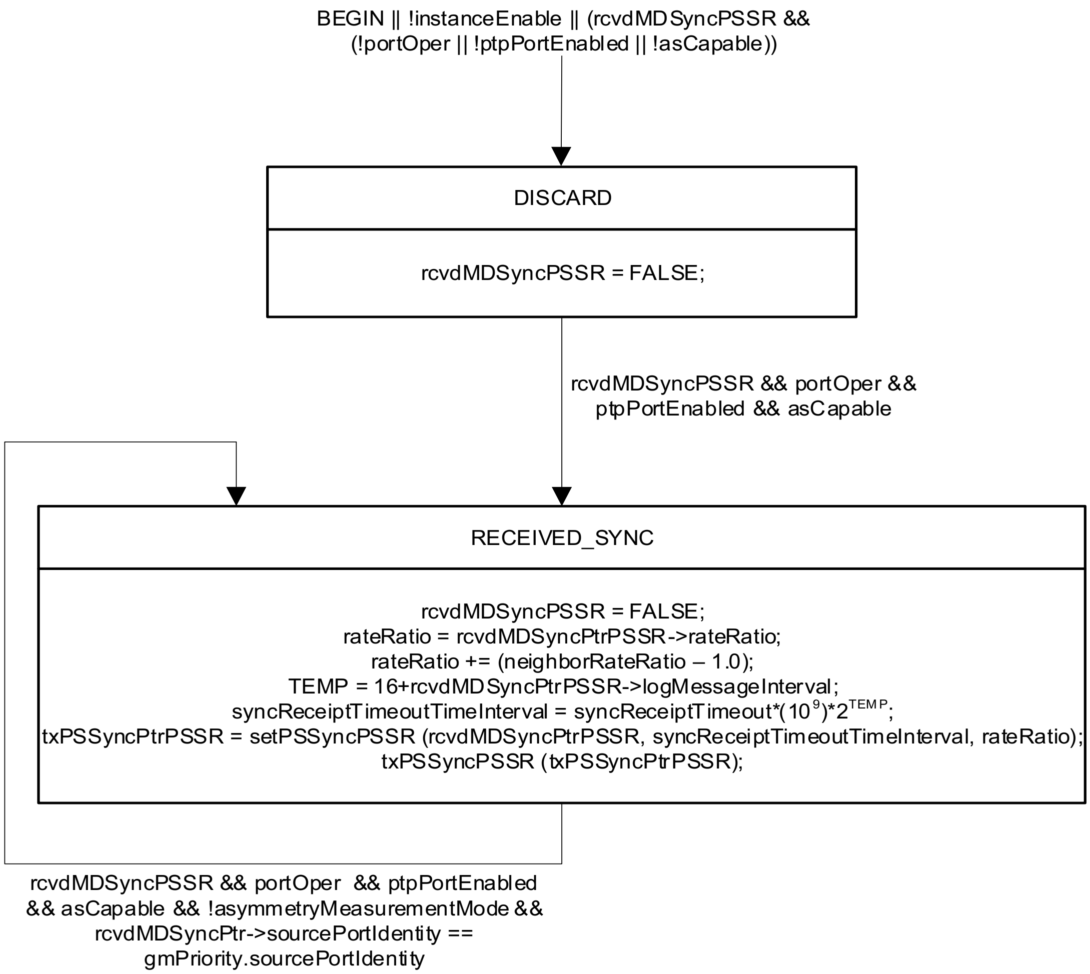


Figure 10-4—PortSyncSyncReceive state machine

10.2.9 ClockMasterSyncSend state machine

10.2.9.1 State machine variables

The following variables are used in the state diagram in Figure 10-5 (in 10.2.9.3):

10.2.9.1.1 syncSendTime: The time in seconds, relative to the LocalClock entity, when synchronization information will next be sent to the SiteSync entity, via a PortSyncSync structure. The data type for syncSendTime is UScaledNs.

10.2.9.1.2 txPSSyncPtrCMSS: A pointer to the PortSyncSync structure transmitted by the state machine.

10.2.9.2 State machine functions

10.2.9.2.1 setPSSyncCMSS (gmRateRatio): Creates a PortSyncSync structure to be transmitted, and returns a pointer to this structure. The members are set as follows:

a) localPortNumber is set to 0.
b) preciseOriginTimestamp is set equal to the masterTime, with any fractional nanoseconds truncated.
c) followUpCorrectionField is set equal to the sum of
   1) The fractional nanoseconds portion of masterTime.fractionalNanoseconds and
   2) The quantity gmRateRatio × (currentTime – localTime).
d) The clockIdentity member of sourcePortIdentity is set equal to the clockIdentity of this PTP Instance.
e) The portNumber member of the sourcePortIdentity is set to 0.

NOTE 1—This quantity and localPortNumber are redundant; both are retained so that the SiteSync entity can process PortSyncSync structures received from a PortSync entity or the ClockMaster entity in the same manner.

f) logMessageInterval is set to clockMasterLogSyncInterval.
g) upstreamTxTime is set equal to localTime.
h) syncReceiptTimeoutTime is set equal to $\text{FFFFFFFFFFFFFFFF}_{16}$, which indicates that there is no sync receipt timeout.

NOTE 2—A ClockMaster entity does not receive Sync messages, and there is no notion of sync receipt timeout.

i) rateRatio is set equal to gmRateRatio.
j) gmTimeBaseIndicator is set equal to clockSourceTimeBaseIndicator.
k) lastGmPhaseChange is set equal to clockSourcePhaseOffset.
l) lastGmFreqChange is set equal to clockSourceFreqOffset.
m) domainNumber is set equal to the domain number of this gPTP domain.

10.2.9.2.2 txPSSyncCMSS (txPSSyncPtrCMSS): Transmits a copy of the PortSyncSync structure pointed to by txPSSyncPtrCMSS to the SiteSync state machine.

10.2.9.2.3 computeClockMasterSyncInterval(): Computes the value of clockMasterSyncInterval (see 10.2.4.2) as $1000000000 \times 2^{\text{clockMasterLogSyncInterval}}$ ns, where clockMasterLogSyncInterval is the minimum currentLogSyncInterval value, taken over all the PTP Ports of the PTP Instance (see 10.7.2.4).

10.2.9.3 State diagram

The ClockMasterSyncSend state machine shall implement the function specified by the state diagram in Figure 10-5, the local variables specified in 10.2.9.1, the functions specified in 10.2.9.2, the structure specified in 10.2.2.3, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine receives masterTime and clockSourceTimeBaseIndicator from the ClockMasterSyncReceive state machine, and phase and frequency offset between masterTime and syncReceiptTime from the ClockMasterSyncOffset state machine. It provides masterTime (i.e., synchronized time) and the phase and frequency offset to the SiteSync entity via a PortSyncSync structure.

The ClockMasterSyncSend state machine is optional for PTP Instances that are not grandmaster-capable (see 8.6.2.1, 10.1.3, and 10.2.1). This state machine may be present in a PTP Instance that is not grandmaster-capable; however, any information supplied by it to the SiteSyncSync state machine is not used by the SiteSyncSync state machine if the PTP Instance is not grandmaster-capable.

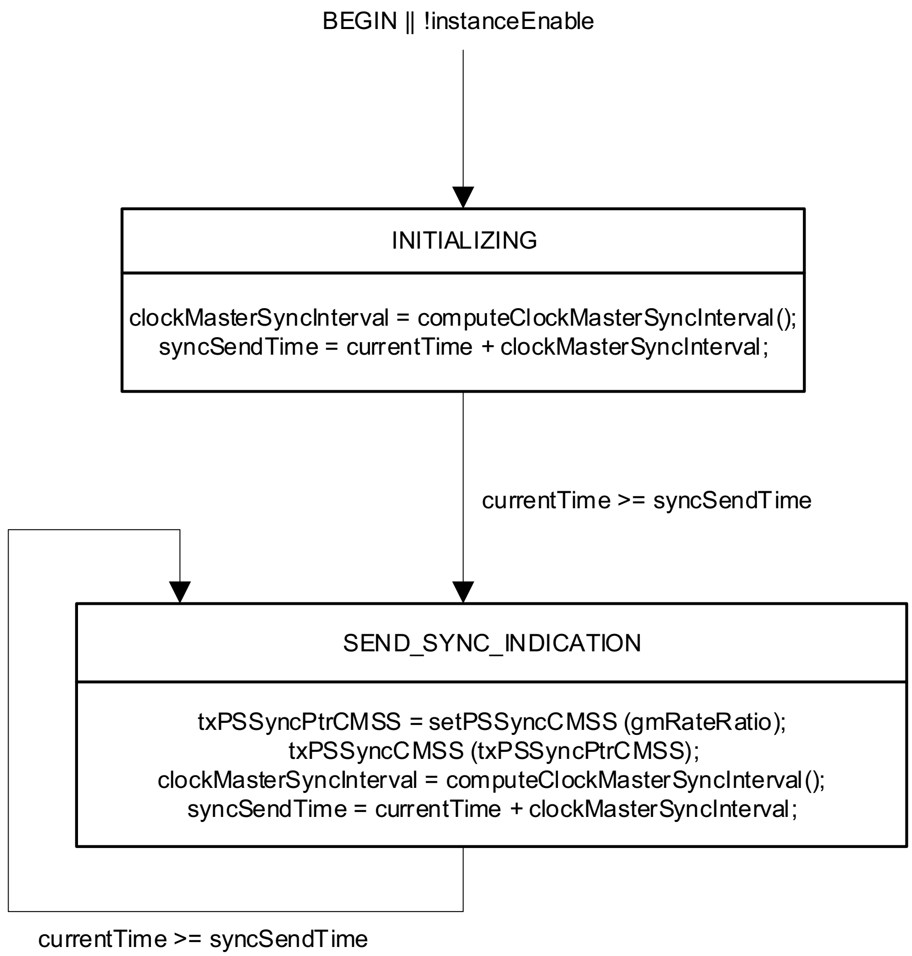


Figure 10-5—ClockMasterSyncSend state machine

10.2.10 ClockMasterSyncOffset state machine

10.2.10.1 State machine variables

The following variable is used in the state diagram in Figure 10-6 (in 10.2.10.3):

10.2.10.1.1 rcvdSyncReceiptTime: A Boolean variable that notifies the current state machine that syncReceiptTime has been updated by the ClockSlave entity. This variable is reset by this state machine.

10.2.10.2 State machine functions

10.2.10.2.1 computeClockSourceFreqOffset(): Computes and returns clockSourceFreqOffset (see 10.2.4.6), using successive values of masterTime computed by the ClockMasterSyncReceive state machine (see 10.2.11) and successive values of syncReceiptTime computed by the ClockSlaveSync state machine (see 10.2.13). The data type for the returned value is Float64. Any scheme that uses this information to compute clockSourceFreqOffset is acceptable as long as the performance requirements specified in B.2.4 are met.

NOTE—As one example, clockSourceFreqOffset can be estimated as the ratio of the duration of a time interval measured by the ClockSource entity to the duration of the same time interval computed from ClockSlaveTime values, minus 1.

10.2.10.3 State diagram

The ClockMasterSyncOffset state machine shall implement the function specified by the state diagram in Figure 10-6, the local variable specified in 10.2.10.1, the function specified in 10.2.10.2, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine receives syncReceiptTime from the ClockSlaveSync state machine and masterTime from the ClockMasterSyncReceive state machine. It computes clockSourcePhaseOffset and clockSourceFrequency offset if this PTP Instance is not currently the Grandmaster PTP Instance, i.e., if selectedState[0] is equal to PassivePort.

The ClockMasterSyncOffset state machine is optional for PTP Instances that are not grandmaster-capable (see 8.6.2.1, 10.1.3, and 10.2.1). This state machine may be present in a PTP Instance that is not grandmaster-capable; however, any information supplied by it, via the ClockMasterSyncSend state machine, to the SiteSyncSync state machine is not used by the SiteSyncSync state machine if the PTP Instance is not grandmaster-capable.

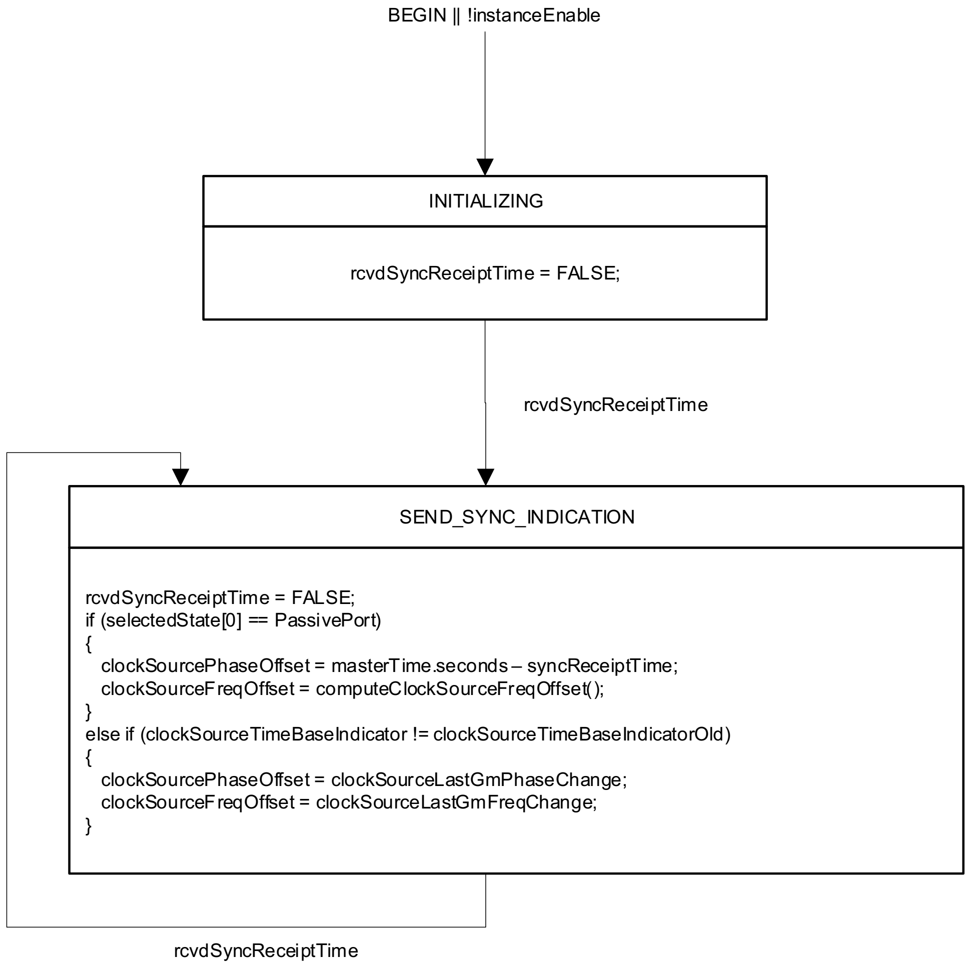


Figure 10-6—ClockMasterSyncOffset state machine

10.2.11 ClockMasterSyncReceive state machine

10.2.11.1 State machine variables

The following variables are used in the state diagram in Figure 10-7 (in 10.2.11.3):

10.2.11.1.1 rcvdClockSourceReq: A Boolean variable that notifies the current state machine when sourceTime is received from the ClockSource entity, due to the ClockSourceTime.invoke primitive having been invoked at that entity. This variable is reset by this state machine.

10.2.11.1.2 rcvdClockSourceReqPtr: A pointer to the received ClockSourceTime.invoke function parameters.

10.2.11.1.3 rcvdLocalClockTickCMSR: A Boolean variable that notifies the current state machine when the LocalClock entity updates its time. This variable is reset by this state machine.

10.2.11.2 State machine functions

10.2.11.2.1 computeGmRateRatio(): Computes gmRateRatio (see 10.2.4.14), using values of sourceTime conveyed by successive ClockSourceTime.invoke functions (see 9.2.2.1), and corresponding values of localTime (see 10.2.4.19). Any scheme that uses this information, along with any other information conveyed by the successive ClockSourceTime.invoke functions and corresponding values of localTime, to compute gmRateRatio is acceptable as long as the performance requirements specified in B.2.4 are met.

NOTE—As one example, gmRateRatio can be estimated as the ratio of the elapsed time of the ClockSource entity that supplies time to this PTP Instance, to the elapsed time of the LocalClock entity of this PTP Instance. This ratio can be computed for the time interval between a received ClockSourceTime.invoke function and a second received ClockSourceTime.invoke function some number of ClockSourceTime.invoke functions later, i.e.,

$$\frac{\text{ClockSource.invoke.sourceTime}_N - \text{ClockSource.invoke.sourceTime}_0}{\text{localTime}_N - \text{localTime}_0}$$

where the successive received ClockSourceTime.invoke functions are indexed from 0 to N, with the first such function indexed as 0, and localTime_j is the value of localTime when the ClockSourceTime.invoke function whose index is j is received.

10.2.11.2.2 updateMasterTime(): Updates the global variable masterTime (see 10.2.4.21), based on information received from the ClockSource and LocalClock entities. It is the responsibility of the application to filter master times appropriately. As one example, masterTime can be set equal to the sourceTime member of the ClockSourceTime.invoke function when this function is invoked at the ClockSource entity and can be incremented by localClockTickInterval (see 10.2.4.18) multiplied by gmRateRatio (see 10.2.4.14) when rcvdLocalClockTickCMSR is TRUE.

10.2.11.3 State diagram

The ClockMasterSyncReceive state machine shall implement the function specified by the state diagram in Figure 10-7, the local variables specified in 10.2.11.1, the functions specified in 10.2.11.2, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine updates the global variable masterTime with information received from the ClockSource entity via the ClockSourceTime.invoke function and information received from the LocalClock entity. It also computes gmRateRatio, i.e., the ratio of the ClockSource entity frequency and the LocalClock entity frequency.

The ClockMasterSyncReceive state machine is optional for PTP Instances that are not grandmaster-capable (see 8.6.2.1, 10.1.3, and 10.2.1). This state machine may be present in a PTP Instance that is not grandmaster-capable; however, any information supplied by it, via the ClockMasterSyncSend state machine, to the SiteSyncSync state machine is not used by the SiteSyncSync state machine if the PTP Instance is not grandmaster-capable.

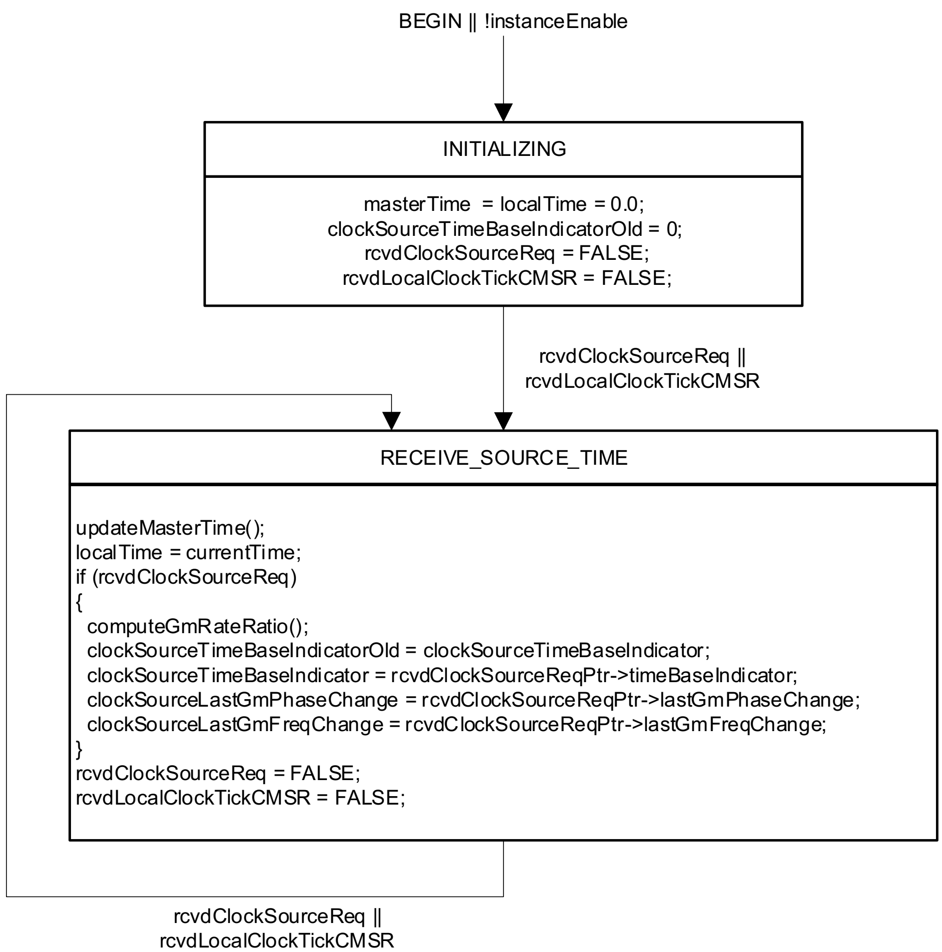


Figure 10-7—ClockMasterSyncReceive state machine

10.2.12 PortSyncSyncSend state machine

10.2.12.1 State machine variables

The following variables are used in the state diagram in Figure 10-8 (in 10.2.12.3):

10.2.12.1.1 rcvdPSSyncPSSS: A Boolean variable that notifies the current state machine when a PortSyncSync structure is received from the SiteSyncSync state machine of the SiteSync entity of the PTP Instance (see 10.2.2.3). This variable is reset by this state machine.

10.2.12.1.2 rcvdPSSyncPtrPSSS: A pointer to the received PortSyncSync structure indicated by rcvdPSSyncPSSS.

10.2.12.1.3 lastPreciseOriginTimestamp: The preciseOriginTimestamp member of the most recently received PortSyncSync structure. The data type for lastPreciseOriginTimestamp is Timestamp.

10.2.12.1.4 lastFollowUpCorrectionField: The followUpCorrectionField member of the most recently received PortSyncSync structure. The data type for lastFollowUpCorrectionField is ScaledNs.

10.2.12.1.5 lastRateRatio: The rateRatio member of the most recently received PortSyncSync structure. The data type for lastRateRatio is Float64.

10.2.12.1.6 lastUpstreamTxTime: The upstreamTxTime member of the most recently received PortSyncSync structure. The data type for lastUpstreamTxTime is UScaledNs.

10.2.12.1.7 lastSyncSentTime: The value of currentTime (i.e., the time relative to the LocalClock entity) when the most recent MDSyncSend structure was sent. The data type for lastSyncSentTime is UScaledNs.

NOTE—lastSyncSentTime is the time the abstract MDSyncSend structure was sent, NOT the time the corresponding Sync message (or equivalent) was sent on a physical link.

10.2.12.1.8 lastRcvdPortNum: The portNumber of the PTP Port on which time-synchronization information was most recently received. The data type for lastRcvdPortNum is UInteger16.

10.2.12.1.9 lastGmTimeBaseIndicator: The gmTimeBaseIndicator of the most recently received PortSyncSync structure. The data type for lastGmTimeBaseIndicator is UInteger16.

10.2.12.1.10 lastGmPhaseChangePSSS: The lastGmPhaseChange of the most recently received PortSyncSync structure. The data type for lastGmPhaseChange is ScaledNs.

10.2.12.1.11 lastGmFreqChangePSSS: The lastGmFreqChange of the most recently received PortSyncSync structure. The data type for lastGmPhaseChange is Float64.

10.2.12.1.12 txMDSyncPtr: A pointer to the MDSyncSend structure sent to the MD entity of this PTP Port.

10.2.12.1.13 numberSyncTransmissions: A count of the number of consecutive Sync message transmissions after the SyncIntervalSetting state machine (see Figure 10-20) has set syncSlowdown (see 10.2.5.17) to TRUE. The data type for numberSyncTransmissions is UInteger8.

10.2.12.1.14 interval1: A local variable that holds either syncInterval or oldSyncInterval. The data type for interval1 is UScaledNs.

10.2.12.2 State machine functions

10.2.12.2.1 setMDSync(): Creates an MDSyncSend structure, and returns a pointer to this structure. The members are set as follows:

a) sourcePortIdentity is set to the portIdentity of this PTP Port (see 8.5.2).
b) logMessageInterval is set equal to the value of currentLogSyncInterval for this PTP Port (see 10.7.2.3).
c) preciseOriginTimestamp is set equal to lastPreciseOriginTimestamp (see 10.2.12.1.3).
d) rateRatio is set equal to lastRateRatio (see 10.2.12.1.5).
e) followUpCorrectionField is set equal to lastFollowUpCorrectionField (see 10.2.12.1.4).
f) upstreamTxTime is set equal to lastUpstreamTxTime (see 10.2.12.1.6).
g) gmTimeBaseIndicator is set to lastGmTimeBaseIndicator (see 10.2.12.1.9).
h) lastGmPhaseChange (structure member) is set to lastGmPhaseChangePSSS (see 10.2.12.1.10).
i) lastGmFreqChange (structure member) is set to lastGmFreqChangePSSS (see 10.2.12.1.11).
j) domainNumber is set equal to the domain number of this gPTP domain (see 8.1).

10.2.12.2.2 txMDSync(txMDSyncPtr): Transmits the MDSyncSend structure pointed to by txMDSyncPtr, to the MD entity of this PTP Port.

10.2.12.3 State diagram

The PortSyncSyncSend state machine shall implement the function specified by the state diagram in Figure 10-8, the local variables specified in 10.2.12.1, the functions specified in 10.2.12.2, the structures specified in 10.2.2.1 through 10.2.2.3, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine receives time-synchronization information from the SiteSyncSync state machine, corresponding to the receipt of the most recent synchronization information on either the slave port, if this PTP Instance is not the Grandmaster PTP Instance, or from the ClockMasterSyncSend state machine, if this PTP Instance is the Grandmaster PTP Instance. The state machine causes time-synchronization information to be sent to the MD entity if this PTP Port is a MasterPort.

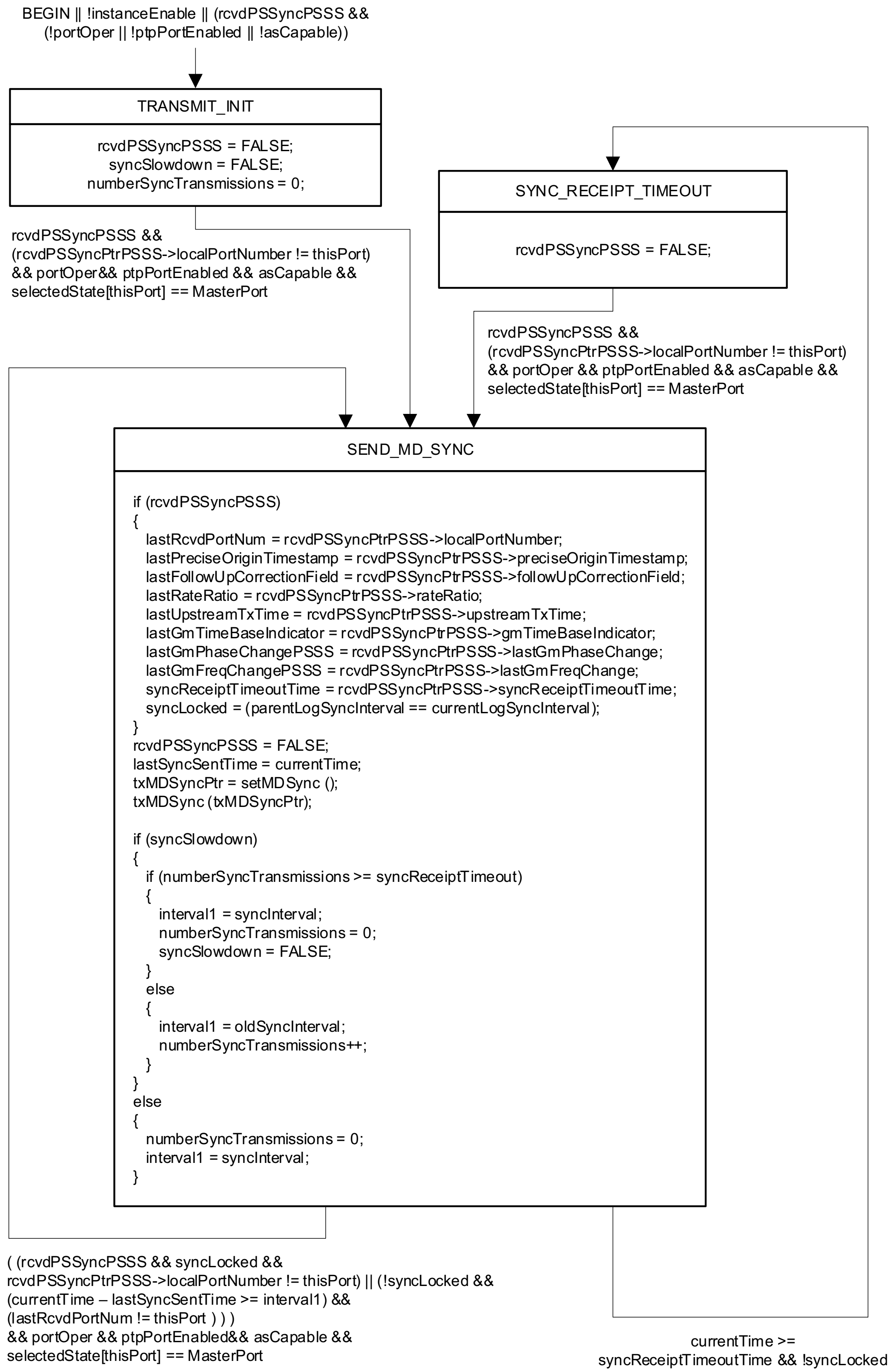


Figure 10-8—PortSyncSyncSend state machine

10.2.13 ClockSlaveSync state machine

10.2.13.1 State machine variables

The following variables are used in the state diagram in Figure 10-9 (in 10.2.13.3):

10.2.13.1.1 rcvdPSSyncCSS: A Boolean variable that notifies the current state machine when a PortSyncSync structure is received from the SiteSyncSync state machine of the SiteSync entity. This variable is reset by this state machine.

10.2.13.1.2 rcvdLocalClockTickCSS: A Boolean variable that notifies the current state machine when the LocalClock entity updates its time. This variable is reset by this state machine.

10.2.13.1.3 rcvdPSSyncPtrCSS: A pointer to the received PortSyncSync structure.

10.2.13.2 State machine functions

10.2.13.2.1 updateSlaveTime(): Updates the global variable clockSlaveTime (see 10.2.4.3), based on information received from the SiteSync and LocalClock entities. It is the responsibility of the application to filter slave times appropriately (see B.3 and B.4 for examples). As one example, clockSlaveTime can be:

a) Set to syncReceiptTime at every LocalClock update immediately after a PortSyncSync structure is received, and
b) Incremented by localClockTickInterval (see 10.2.4.18) multiplied by the rateRatio member of the previously received PortSyncSync structure during all other LocalClock updates.

If no PTP Instance is grandmaster-capable, i.e., gmPresent is FALSE, then clockSlaveTime is set to the time provided by the LocalClock. This function is invoked when rcvdLocalClockTickCSS is TRUE.

10.2.13.2.2 invokeApplicationInterfaceFunction (functionName): Invokes the application interface function whose name is functionName. For the ClockSlaveSync state machine, functionName is clockTargetPhaseDiscontinuity.result (see 9.6.2).

10.2.13.3 State diagram

The ClockSlaveSync state machine shall implement the function specified by the state diagram in Figure 10-9, the local variables specified in 10.2.13.1, the functions specified in 10.2.13.2, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine receives a PortSyncSync structure from the SiteSyncSync state machine. It computes syncReceiptTime and clockSlaveTime, and sets syncReceiptLocalTime (i.e., the time relative to the LocalClock entity corresponding to syncReceiptTime), GmTimeBaseIndicator, lastGmPhaseChange, and lastGmFreqChange. It provides clockSlaveTime to the ClockMasterSyncOffset state machine, and provides information to the ClockTarget entity (via the ClockTargetPhaseDiscontinuity interface; see 9.6) to enable that entity to determine if a phase or frequency discontinuity has occurred.

The per-PTP Port global variables used in the ClockSlaveSync state machine are determined based on rcvdPSSyncPtrCSS->localPortNumber, as follows:

a) If rcvdPSSyncPtrCSS->localPortNumber > 0, the per-PTP Port global variables of PTP Port number rcvdPSSyncPtrCSS->localPortNumber are used.
b) If rcvdPSSyncPtrCSS->localPortNumber == 0, the values of the used per-PTP Port global variables are fixed as follows:
   1) meanLinkDelay = 0
   2) delayAsymmetry = 0
   3) neighborRateRatio = 1.0

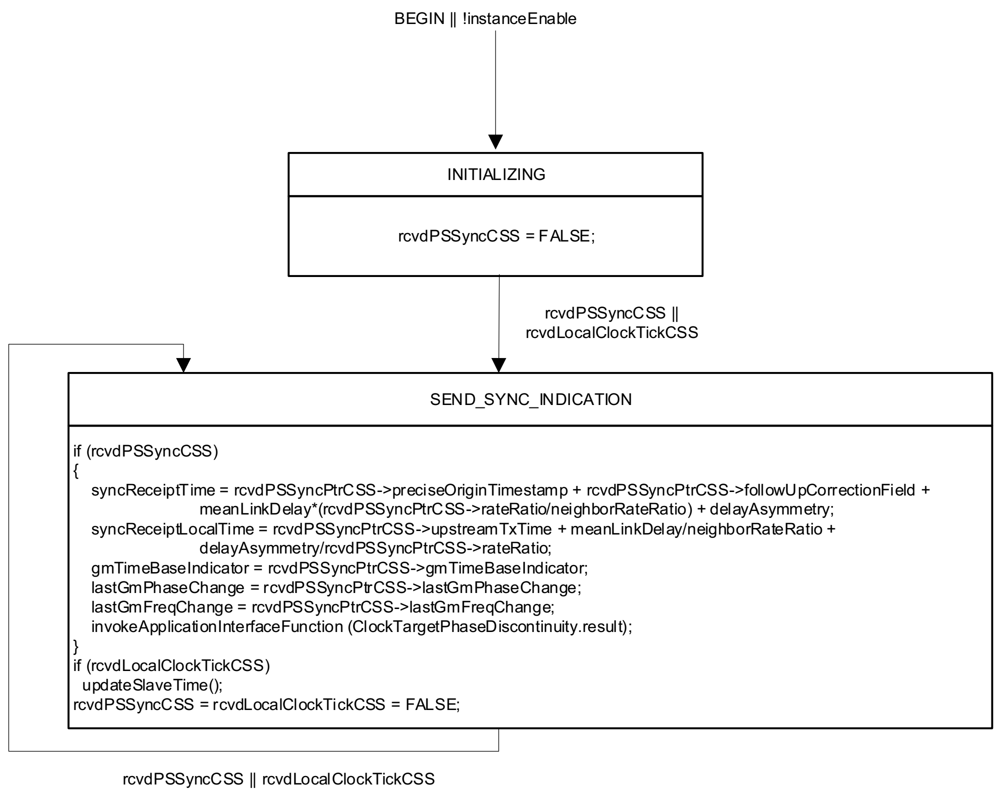


Figure 10-9—ClockSlaveSync state machine

10.3 Best master clock selection, external port configuration, and announce interval setting state machines

10.3.1 Best master clock selection and external port configuration overview

10.3.1.1 General

There are two methods for setting the Grandmaster PTP Instance and time-synchronization spanning tree for a gPTP domain:

a) The BMCA is used to determine the Grandmaster PTP Instance for a gPTP domain and construct the time-synchronization spanning tree with that Grandmaster PTP Instance as the root. In this case, the network is configured automatically, i.e., the PTP Port states are set, using the results of the BMCA.
b) The PTP Port states are configured to force a desired Grandmaster PTP Instance and to construct a desired time-synchronization spanning tree with the Grandmaster PTP Instance as the root.

The PTP Port state definitions are given in Table 10-2.

The per PTP Instance global variable externalPortConfigurationEnabled indicates whether method a) or b) is used; a value of TRUE indicates method b), and a value of FALSE indicates method a) (see 10.3.9.24). The data type of externalPortConfigurationEnabled is Boolean. Method a) is implemented and is the default mode of operation (i.e., externalPortConfigurationEnabled is FALSE) on domain 0 to maintain backward compatibility. For domains other than domain 0, the following statements apply:

c) At least one of the possibilities [method a) or b)] is implemented.
d) Both possibilities can be implemented.
e) If both possibilities are implemented, the default value of externalPortConfigurationEnabled is FALSE.

Once an Announce message is transmitted by a PTP Port, subsequent timing information (see 7.4) transmitted by that PTP Port is derived from the Grandmaster PTP Instance indicated in that Announce message.

Table 10-2—PTP Port state definitions

PTP Port state	Description
MasterPort	Any PTP Port, P, of the PTP Instance that is closer to the root than any other PTP Port of the gPTP communication path connected to P.
SlavePort	The one PTP Port of the PTP Instance that is closest to the root PTP Instance. If the root is grandmaster-capable, the SlavePort is also closest to the Grandmaster PTP Instance. The PTP Instance does not transmit Sync or Announce messages on the SlavePort.
PassivePort	Any PTP Port of the PTP Instance whose PTP Port state is not MasterPort, SlavePort, or DisabledPort.
DisabledPort	Any PTP Port of the PTP Instance for which the variables portOper, ptpPortEnabled, and asCapable are not all TRUE.
NOTE—PTP Port states are per PTP Port and per domain (i.e., per PTP Instance; see 8.1).	

NOTE—Information contained in Sync and associated Follow_Up messages received on PTP Ports whose PTP Port state is PassivePort is discarded; the SiteSyncSync state machine (see 10.2.7) uses only information received from a PTP Port whose PTP Port state is SlavePort.

An example master/slave hierarchy of PTP Instances is shown in Figure 10-10. The Grandmaster PTP Instance ports all have PTP Port state of MasterPort. All the other PTP Instances have exactly one slave port. The time-synchronization spanning tree is composed of the PTP Instances and the links that do not have an endpoint PTP Port whose PTP Port state is PassivePort.

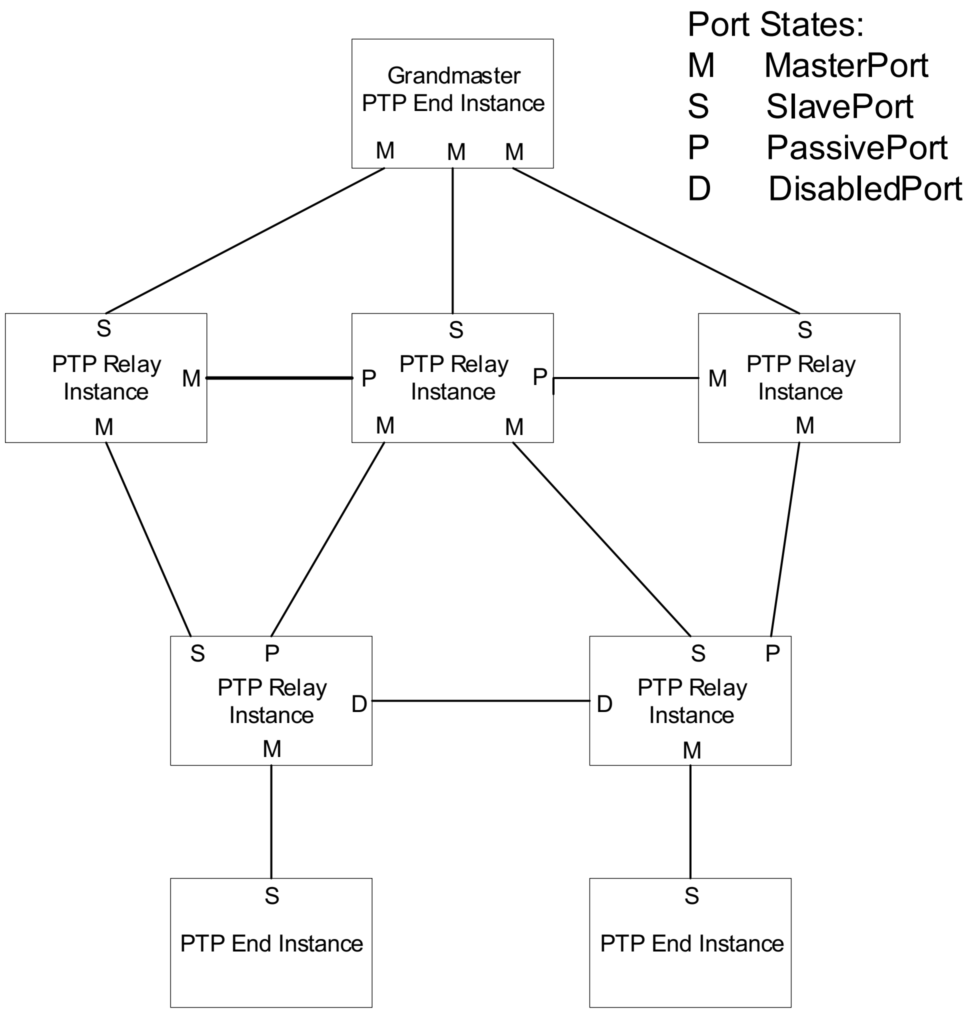


Figure 10-10—Example master/slave hierarchy of PTP Instances

10.3.1.2 Best master clock algorithm overview

In the BMCA (i.e., method a) of 10.3.1.1), best master selection information is exchanged between PTP Instances of time-aware systems via Announce messages (see 10.5 and 10.6). Each Announce message contains time-synchronization spanning tree vector information that identifies one PTP Instance as the root of the time-synchronization spanning tree and, if the PTP Instance is grandmaster-capable, the Grandmaster PTP Instance. Each PTP Instance in turn uses the information contained in the Announce messages it receives, along with its knowledge of itself, to compute which of the PTP Instances that it has knowledge of ought to be the root of the spanning tree and, if grandmaster-capable, the Grandmaster PTP Instance. As part of constructing the time-synchronization spanning tree, each PTP Port of each PTP Instance is assigned a PTP Port state from Table 10-2 by state machines associated with the ports and with the PTP Instance as a whole.

NOTE—The BMCA described in this standard is the default BMCA according to the specifications of 9.3 of IEEE Std 1588-2019. It is also equivalent to a subset of the Rapid Spanning Tree Protocol (RSTP) described in IEEE Std 802.1Q-2018 (though the full RSTP described in IEEE Std 802.1Q-2018 is not equivalent to the full BMCA described in IEEE Std 1588-2019). The BMCA description here uses the formalism of the RSTP description in IEEE Std 802.1Q-2018.

10.3.1.3 external port configuration overview

In external port configuration (i.e., method b) of 10.3.1.1), an external entity determines the synchronization spanning tree and sets the PTP Port states accordingly. The method used by the external entity to determine the synchronization spanning tree is outside the scope of this standard. However, as with the BMCA, Announce messages are used to transport information on the time-synchronization spanning tree and Grandmaster PTP Instance time properties information from one PTP Instance to the next in the tree. The external entity sets the state of a PTP Port by setting the value of externalPortConfigurationPortDS.desiredState to the desired state.

10.3.2 systemIdentity

The systemIdentity attribute of a PTP Instance is a UInteger112 (i.e., a 14-byte, unsigned integer) formed by concatenating the following attributes, in the following order, from most significant to least significant octet:

a) priority1 (1 octet; see 8.6.2.1)
b) clockClass (1 octet; see 8.6.2.2 and 6.4.3.8)
c) clockAccuracy (1 octet; see 8.6.2.3 and 6.4.3.8)
d) offsetScaledLogVariance (2 octets; see 8.6.2.4 and 6.4.3.8)
e) priority2 (1 octet; see 8.6.2.5)
f) clockIdentity (8 octets; see 8.5.2.2 and 6.4.3.6)

The systemIdentity attribute is defined for convenience when comparing two PTP Instances to determine, when using the BMCA (i.e., method a) of 10.3.1.1), which is a better candidate for root and if the PTP Instance is grandmaster-capable (i.e., the value of priority1 is less than 255; see 8.6.2.1). Two PTP Instances are compared as follows. Let the systemIdentity of PTP Instance A be S_A and the systemIdentity of PTP Instance B be S_B. Let the clockIdentity of A be C_A and the clockIdentity of B be C_B. Then, if $C_A \neq C_B$, i.e., A and B represent different PTP Instances,

g) A is better than B if and only if $S_A < S_B$, and
h) B is better than A if and only if $S_B < S_A$.

If $C_A = C_B$, i.e., A and B represent the same PTP Instance,

i) $S_A < S_B$ means that A represents an upgrading of the PTP Instance compared to B or, equivalently, B represents a downgrading of the PTP Instance compared to A,
j) $S_B < S_A$ means that B represents an upgrading of the PTP Instance compared to A or, equivalently, A represents a downgrading of the PTP Instance compared to B, and
k) $S_A = S_B$ means that A and B represent the same PTP Instance that has not changed.

Comparisons g) and h) in this subclause imply that, with the ordering of attributes in the systemIdentity, the clockIdentity is a tie-breaker when two different PTP Instances that have identical attributes a) through e) are compared.

Comparisons g) and h) also imply that a PTP Instance that is grandmaster-capable is always better than another PTP Instance that is not grandmaster-capable because the priority1 is less than 255 if the PTP Instance is grandmaster-capable and is equal to 255 if it is not grandmaster-capable (see 8.6.2.1).

The cases where A and B represent different PTP Instances and represent the same PTP Instance are handled separately in the BMCA. When comparing two different PTP Instances, the better PTP Instance is selected as the Grandmaster PTP Instance candidate. However, if A and B represent the same PTP Instance with attributes that have changed, the PTP Instance is considered as having the most recent attributes when doing subsequent comparisons with other PTP Instances.

10.3.3 stepsRemoved

Every PTP Instance has a stepsRemoved associated with it. For the root PTP Instance, and therefore the Grandmaster PTP Instance when the root is grandmaster-capable, it is zero. For all other PTP Instances, it is the number of gPTP communication paths in the path from the root to the respective PTP Instance.

NOTE—For example, stepsRemoved for a slave port on the same gPTP communication path as the Grandmaster PTP Instance will have a value of 1, indicating that a single path was traversed.

The stepsRemoved attributes of different ports of a PTP Instance are compared after comparisons of other attributes that take precedence (i.e., priority1, clockClass, clockAccuracy, offsetScaledLogVariance, priority2) do not result in one PTP Port being declared better than the other. Among the ports whose stepsRemoved attributes are compared, the PTP Port on the PTP Instance with the lowest stepsRemoved is assigned the state of SlavePort for that PTP Instance (the root PTP Instance does not have a SlavePort). This lowest stepsRemoved is also considered the stepsRemoved for the PTP Instance. If a PTP Instance has two or more ports with the same stepsRemoved, then the PTP Port with the smallest portNumber is selected as the SlavePort.

10.3.4 time-synchronization spanning tree priority vectors

PTP Instances send best master selection information to each other in Announce messages. The information is structured in a time-synchronization spanning tree priority vector. Time-synchronization spanning tree priority vectors provide the basis for a concise specification of the BMCA's determination of the time-synchronization spanning tree and Grandmaster PTP Instance. A priority vector is formed by concatenating the following attributes, in the following order, from most significant to least significant octet:

a) rootSystemIdentity (14 octets; see 10.3.2)
b) stepsRemoved (2 octets; see 10.3.3)
c) sourcePortIdentity (i.e., portIdentity of the transmitting PTP Instance; 10 octets; see 8.5.2 and 10.6.2)
d) portNumber of the receiving PTP Port (2 octets; see 8.5.2.3)

The first two components of a priority vector are significant throughout the gPTP domain; they are propagated via Announce messages and updated through invocation of BMCA state machines. The next component is assigned hop-by-hop for each gPTP communication path or PTP Instance and thus is of local significance only. It is used as a tie-breaker in decisions between time-synchronization spanning tree priority vectors that are otherwise equal. The fourth component is not conveyed in Announce messages, but is used as a tie-breaker within a PTP Instance.

The set of all time-synchronization spanning tree priority vectors is totally ordered. For all components, a lesser numerical value is better, and earlier components in the preceding list are more significant. In addition, as mentioned earlier, a priority vector that reflects a root PTP Instance that is grandmaster-capable is always better than a priority vector that reflects a root PTP Instance that is not grandmaster-capable. As each PTP Port receives a priority vector, via an Announce message, from ports closer to the root, additions are made to one or more components to yield a worse priority vector. This process of receiving information, adding to it, and passing it on, can be described in terms of the message priority vector received and a set of priority vectors used to facilitate the computation of a priority vector for each PTP Port, to be transmitted in further Announce Messages to PTP Instances further from the root.

10.3.5 Priority vector calculations

The portPriorityVector is the time-synchronization spanning tree priority vector held for the PTP Port when the reception of Announce messages and any pending update of information has been completed:

portPriorityVector = {rootSystemIdentity : stepsRemoved : sourcePortIdentity : portNumber}

A messagePriorityVector is the time-synchronization spanning tree priority vector conveyed in a received Announce Message. For a PTP Instance S receiving an Announce Message on PTP Port P_S with portNumber PN_S, from a MasterPort with portIdentity P_M on PTP Instance M claiming a rootSystemIdentity of R_M and a stepsRemoved of SR_M:

messagePriorityVector = $\{R_M : SR_M : P_M : PN_S\}$

This messagePriorityVector is superior to the portPriorityVector and will replace it if, and only if, the messagePriorityVector is better than the portPriorityVector, or the Announce message has been transmitted from the same master PTP Instance and MasterPort as the portPriorityVector, i.e., if the following is true:

((R_M < rootSystemIdentity)) ||

((R_M == rootSystemIdentity) && (SR_M < stepsRemoved)) ||

((R_M == rootSystemIdentity) && (SR_M == stepsRemoved) && (P_M < sourcePortIdentity (of current master PTP Instance))) ||

((R_M == rootSystemIdentity) && (SR_M == stepsRemoved)

&& (P_M == sourcePortIdentity (of current master PTP Instance)) && (PN_S < portNumber)) ||

((P_M.clockIdentity == sourcePortIdentity.clockIdentity (of current master PTP Instance)) &&
(P_M.portNumber == sourcePortIdentity.PortNumber (of the current master PTP Instance)))

A gmPathPriorityVector can be calculated from a received portPriorityVector by adding one to the stepsRemoved component:

gmPathPriorityVector = $\{R_M : SR_M + 1 : P_M : PN_S\}$

The systemPriorityVector for a PTP Instance S with systemIdentity S_S and clockIdentity C_S is the priority vector that would, with the portIdentity of the SlavePort set equal to the portIdentity of the transmitting PTP Port, be used as the message priority vector in Announce Messages transmitted on S's ports whose state is MasterPort if S was selected as the root:

systemPriorityVector = $\{S_S : 0 : \{C_S : 0\} : 0\}$

The gmPriorityVector for S is the best of the set comprising the systemPriorityVector vector plus every gmPathPriorityVector for which the clockIdentity of the master PTP Instance portIdentity is not the clockIdentity of S. If the systemPriorityVector is best, S has been selected as the root. When the best gmPathPriorityVector is that of PTP Port PN_S above, then:

gmPriorityVector = $\{S_S : 0 : \{C_S : 0\} : 0\}$if S is better than R_M, or

gmPriorityVector = $\{R_M : SR_M + 1 : P_M : PN_S\}$ if S is worse than R_M.

The masterPriorityVector for a PTP Port Q on PTP Instance S is the gmPriorityVector with S's clockIdentity C_S substituted for the clockIdentity of the master portIdentity, and Q's portNumber PN_Q substituted for the portNumber of the master portIdentity and for the portNumber of the receiving PTP Port:

masterPriorityVector = $\{S_S : 0 : \{C_S : PN_Q\} : PN_Q\}$if S is better than R_M, or
masterPriorityVector = $\{R_M : SR_M + 1 : \{C_S : PN_Q\} : PN_Q\}$ if S is worse than R_M.

If the masterPriorityVector is better than the portPriorityVector, the PTP Port will be the MasterPort for the attached gPTP communication path and the portPriorityVector will be updated. The messagePriorityVector information in Announce messages transmitted by a PTP Port always includes the first three components of the masterPriorityVector of the PTP Port.

NOTE—The consistent use of lower numerical values to indicate better information is deliberate as the MasterPort that is closest to the root, i.e., has a numerically lowest path cost component, is selected from amongst potential alternatives for any given gPTP communication path. Adopting the conventions that lower numerical values indicate better information, that where possible more significant priority components are encoded earlier in the octet sequence of an Announce message, and that earlier octets in the encoding of individual components are more significant allows concatenated octets that compose a priority vector to be compared as if they were a multiple octet encoding of a single number, without regard to the boundaries between the encoded components. To reduce the confusion that naturally arises from having the lesser of two numerical values represent the better of the two, i.e., the one to be chosen all other factors being equal, this clause uses the following consistent terminology. Relative numeric values are described as "least," "lesser," "equal," and "greater," and their comparisons as "less than," "equal to," or "greater than," while relative time-synchronization spanning tree priorities are described as "best," "better," "the same," "different," and "worse" and their comparisons as "better than," "the same as," "different from," and "worse than." The operators "<" and "==" represent less than and equal to, respectively. The terms "superior" and "inferior" are used for comparisons that are not simply based on priority, but can include the fact that the priority vector of a MasterPort can replace an earlier vector transmitted in an Announce message by the same PTP Port.

10.3.6 PTP Port state assignments

10.3.6.1 PTP Port state assignments when the BMCA is used

The BMCA assigns one of the following PTP Port states to each PTP Port: MasterPort, SlavePort, PassivePort, or DisabledPort.

The DisabledPort state is assigned if portOper is FALSE (see 10.2.5.12), ptpPortEnabled is FALSE (see 10.2.5.13), or asCapable is FALSE (see 10.2.5.1).

A PTP Port for which portOper, ptpPortEnabled, and asCapable are all TRUE has its PTP Port state assigned according to the source and relative priority of the time-synchronization spanning tree portPriorityVector (see 10.3.4 and 10.3.5) as follows:

a) If the PTP Instance is not the root, the source of the gmPriorityVector is the SlavePort.
b) Each PTP Port whose portPriorityVector is its masterPriorityVector is a MasterPort.
c) Each PTP Port, other than the SlavePort, whose portPriorityVector has been received from another PTP Instance or another PTP Port on this PTP Instance is a PassivePort.

10.3.6.2 PTP Port state assignments when external port configuration is used

If external port configuration is used, one of the states MasterPort, SlavePort, PassivePort, or DisabledPort is assigned to each PTP Port by an external entity, as described in this subclause.

The DisabledPort state is assigned if portOper is FALSE (see 10.2.5.12), ptpPortEnabled is FALSE (see 10.2.5.13), or asCapable is FALSE (see 10.2.5.1).

The member externalPortConfigurationPortDS.desiredState (see 14.12.2) is used by an external entity to set the state of the respective PTP Port to MasterPort, SlavePort, or PassivePort. When this member is set, its value is copied to the per PTP Port local variable portStateInd (see 10.3.15.1.5). If portOper, ptpPortEnabled, and asCapable are all TRUE for this PTP Port, the PTP Port state is set equal to the value of externalPortConfigurationPortDS.desiredState by copying the value of this member to the element of the selectedState array (see 10.2.4.20) for this PTP Port.

10.3.7 Overview of best master clock selection, external port configuration, and announce interval setting state machines

10.3.7.1 Best master clock selection state machines overview

The best master clock selection function in a PTP Instance is specified by a number of cooperating state machines. Figure 10-11 is not itself a state machine, but illustrates the machines, their local variables, their interrelationships, their performance parameters, and the global variables and structures used to communicate between them.

NOTE—The BMCA state machines are all invoked by the media-independent layer, i.e., by the SiteSync and PortSync entities. The media-dependent layer, i.e., the MD entity, simply takes an Announce message received from the PortSync entity of the same PTP Port and gives it to the next lower layer (e.g., IEEE 802.3, IEEE 802.11). It is the PortSync entity that generates and consumes Announce messages.

The following media-independent layer state machines are in Figure 10-11:

a) PortAnnounceReceive (one instance per PTP Instance, per PTP Port): receives Announce information from the MD entity of the same PTP Port, determines if the Announce message is qualified and, if so, sets the rcvdMsg variable. This state machine is invoked by the PortSync entity of the PTP Port.
b) PortAnnounceInformation (one instance per PTP Instance, per PTP Port): receives new qualified Announce information from the PortAnnounceReceive state machine, determines if the Announce information is better than the current best master information it knows about, and updates the current best master information when it receives updated PTP Port state information from the PortStateSelection state machine and when announce receipt timeout or, when gmPresent is TRUE, sync receipt timeout occurs. This state machine is invoked by the PortSync entity of the PTP Port.
c) PortStateSelection (one instance per PTP Instance): updates the gmPathPriority vector for each PTP Port of the PTP Instance, the gmPriorityVector for the PTP Instance, and the masterPriorityVector for each PTP Port of the PTP Instance; determines the PTP Port state for each PTP Port; and updates gmPresent. This state machine is invoked by the SiteSync entity of the PTP Instance.
d) PortAnnounceTransmit (one instance per PTP Instance, per PTP Port): if the PTP Port state is MasterPort, transmits Announce information to the MD entity when an announce interval has elapsed, PTP Port states have been updated, and portPriority and portStepsRemoved information has been updated with newly determined masterPriority and masterStepsRemoved information. This state machine is invoked by the PortSync entity of the PTP Port and is also used when external port configuration is used.

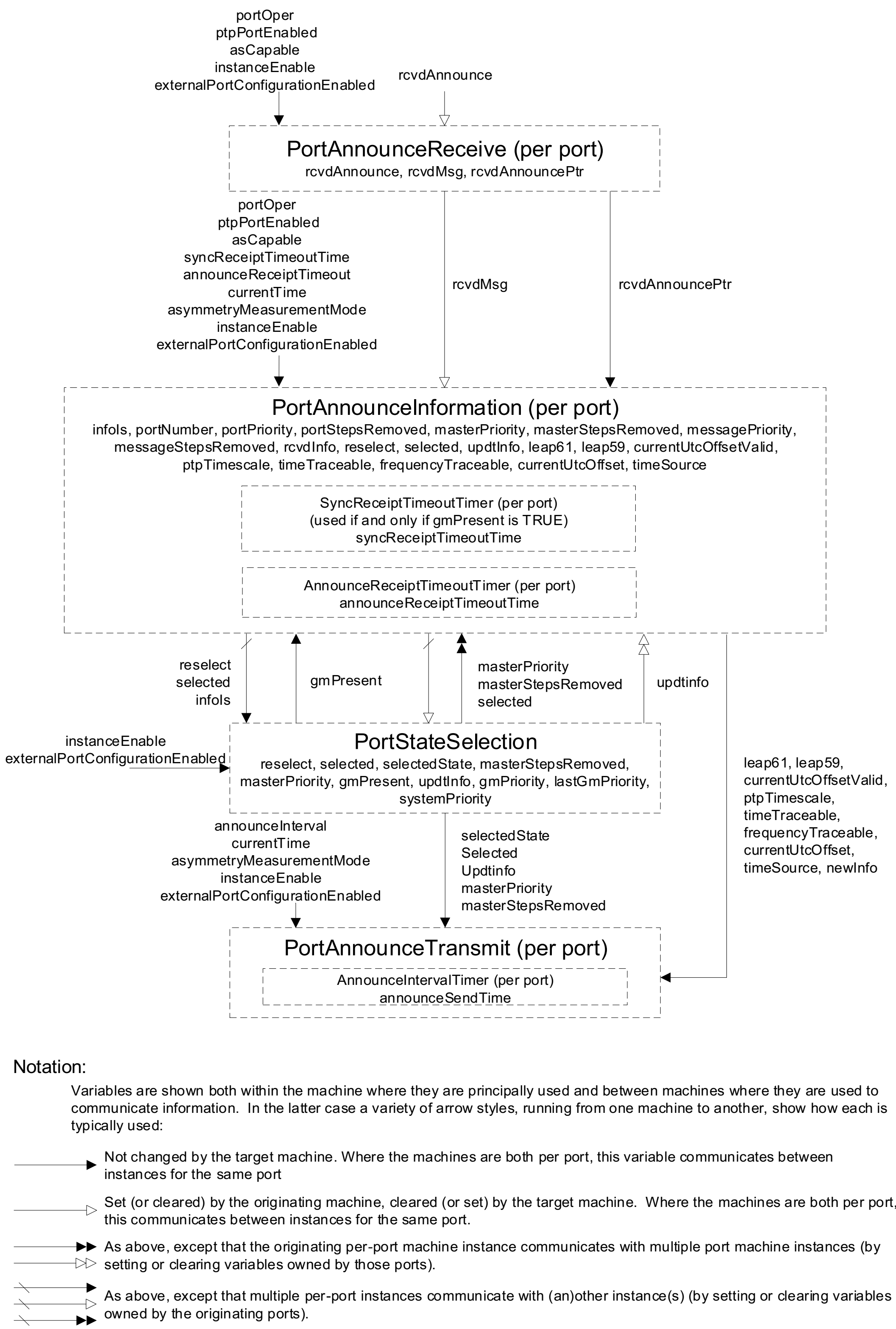


Figure 10-11—Best master clock selection state machines—overview and interrelationships

10.3.7.2 External port configuration state machines overview

The external port configuration function in a PTP Instance is specified by a number of cooperating state machines. Figure 10-12 is not itself a state machine, but illustrates the machines, their local variables, their interrelationships, their performance parameters, and the global variables and structures used to communicate between them.

NOTE—The external port configuration state machines are all invoked by the media-independent layer and are per PTP Port, i.e., they are invoked by the PortSync entity for the respective PTP Port. The media-dependent layer, i.e., the MD entity, simply takes an Announce message received from the PortSync entity of the same PTP Port and gives it to the next lower layer (e.g., IEEE 802.3, IEEE 802.11). It is the PortSync entity that generates and consumes Announce messages.

The following media-independent layer state machines are in Figure 10-12:

a) PortAnnounceInformationExt (one instance per PTP Instance, per PTP Port): Receives and stores new Announce information received in Announce messages.
b) PortStateSettingExt (one instance per PTP Instance): Copies the desired PTP Port state for the PTP Port to the respective selectedState array element, updates gmPresent, computes masterStepsRemoved, stores the time properties information in the respective global variables, and computes the gmPriorityVector and masterPriorityVector.
c) PortAnnounceTransmit (one instance per PTP Instance, per PTP Port): If the PTP Port state is MasterPort, transmits Announce information to the MD entity when an announce interval has elapsed, PTP Port states have been updated, and portPriority and portStepsRemoved information has been updated with newly determined masterPriority and masterStepsRemoved information. This state machine is invoked by the PortSync entity of the PTP Port and is also used when the BMCA is used.

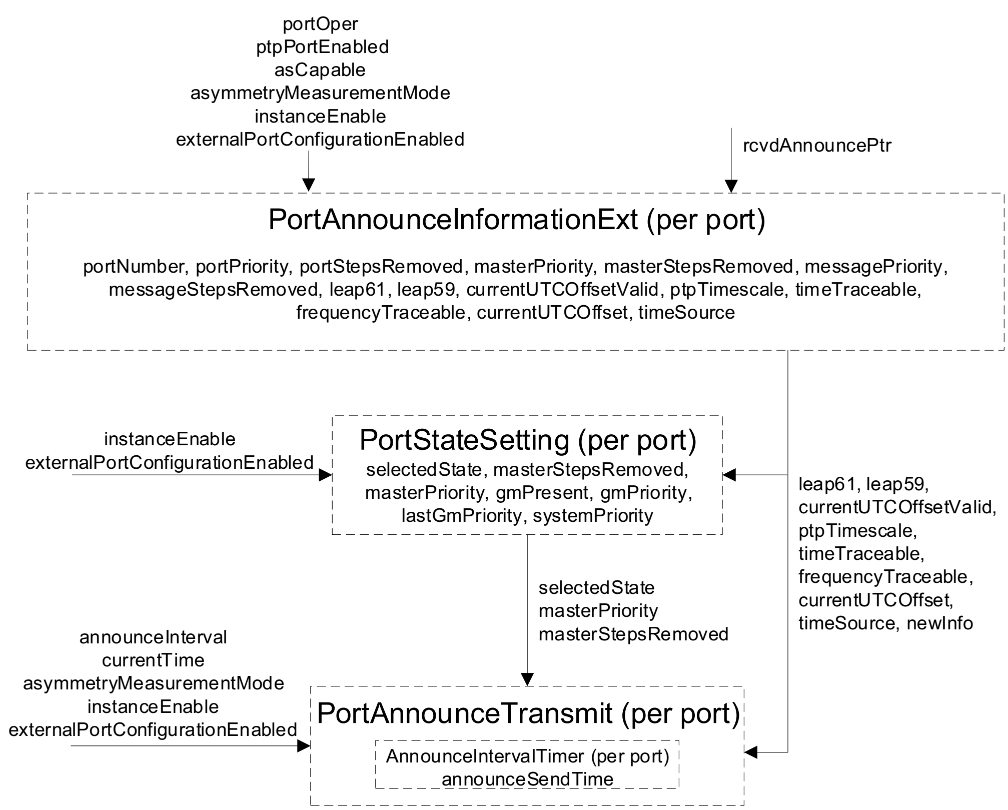


Figure 10-12—External port configuration state machines—overview and interrelationships

10.3.7.3 AnnounceIntervalSetting state machine overview

An additional state machine, the AnnounceIntervalSetting state machine, receives a Signaling message that contains a Message Interval Request TLV (see 10.6.4.3) and sets the global variables that give the duration of the mean interval between successive Announce messages.

10.3.8 Overview of global variables used by best master clock selection, external port configuration, and announce interval setting state machines

Subclauses 10.3.9 and 10.3.10 define global variables used by best master clock selection, external port configuration, and announce interval setting state machines whose scopes are as follows:

- Per PTP Instance (i.e., per domain)
- Per PTP Instance, per PTP Port
- Instance used by CMLDS (see 11.2.17) (i.e., variable is common across all LinkPorts)
- Instance used by CMLDS, per LinkPort

Table 10-3 summarizes the scope of each global variable of 10.3.9 and 10.3.10.

Table 10-3—Summary of scope of global variables used by best master clock selection, external port configuration, and announce interval setting state machines (see 10.3.9 and 10.3.10)

Variable name	Subclause of definition	Per PTP Instance (i.e., per domain)	Per PTP Instance, per PTP Port	Instance used by CMLDS (i.e., variable is common across all LinkPorts)	Instance used by CMLDS, per LinkPort
reselect	10.3.9.1	Yes	No	No	No
selected	10.3.9.2	Yes	No	No	No
masterStepsRemoved	10.3.9.3	Yes	No	No	No
leap61	10.3.9.4	Yes	No	No	No
leap59	10.3.9.5	Yes	No	No	No
currentUtcOffsetValid	10.3.9.6	Yes	No	No	No
ptpTimescale	10.3.9.7	Yes	No	No	No
timeTraceable	10.3.9.8	Yes	No	No	No
frequencyTraceable	10.3.9.9	Yes	No	No	No
currentUtcOffset	10.3.9.10	Yes	No	No	No
timeSource	10.3.9.11	Yes	No	No	No
sysLeap61	10.3.9.12	Yes	No	No	No
sysLeap59	10.3.9.13	Yes	No	No	No
sysCurrentUtcOffsetValid	10.3.9.14	Yes	No	No	No
sysPtpTimescale	10.3.9.15	Yes	No	No	No
sysTimeTraceable	10.3.9.16	Yes	No	No	No
sysFrequencyTraceable	10.3.9.17	Yes	No	No	No

Table 10-3—Summary of scope of global variables used by best master clock selection, external port configuration, and announce interval setting state machines (see 10.3.9 and 10.3.10) *(continued)*

Variable name	Subclause of definition	Per PTP Instance (i.e., per domain)	Per PTP Instance, per PTP Port	Instance used by CMLDS (i.e., variable is common across all LinkPorts)	Instance used by CMLDS, per LinkPort
sysCurrentUtcOffset	10.3.9.18	Yes	No	No	No
sysTimeSource	10.3.9.19	Yes	No	No	No
systemPriority	10.3.9.20	Yes	No	No	No
gmPriority	10.3.9.21	Yes	No	No	No
lastGmPriority	10.3.9.22	Yes	No	No	No
pathTrace	10.3.9.23	Yes	No	No	No
externalPortConfigurationEnabled	10.3.9.24	Yes	No	No	No
lastAnnouncePort	10.3.9.25	Yes	No	No	No
announceReceiptTimeoutTimeInterval	10.3.10.1	No	Yes	No	No
announceSlowdown	10.3.10.2	No	Yes	No	No
oldAnnounceInterval	10.3.10.3	No	Yes	No	No
infoIs	10.3.10.4	No	Yes	No	No
masterPriority	10.3.10.5	No	Yes	No	No
currentLogAnnounceInterval	10.3.10.6	No	Yes	No	No
initialLogAnnounceInterval	10.3.10.7	No	Yes	No	No
announceInterval	10.3.10.8	No	Yes	No	No
messageStepsRemoved	10.3.10.9	No	Yes	No	No
newInfo	10.3.10.10	No	Yes	No	No
portPriority	10.3.10.11	No	Yes	No	No
portStepsRemoved	10.3.10.12	No	Yes	No	No
rcvdAnnouncePtr	10.3.10.13	No	Yes	No	No
rcvdMsg	10.3.10.14	No	Yes	No	No
updtInfo	10.3.10.15	No	Yes	No	No
annLeap61	10.3.10.16	No	Yes	No	No
annLeap59	10.3.10.17	No	Yes	No	No
annCurrentUtcOffsetValid	10.3.10.18	No	Yes	No	No
annPtpTimescale	10.3.10.19	No	Yes	No	No
annTimeTraceable	10.3.10.20	No	Yes	No	No
annFrequencyTraceable	10.3.10.21	No	Yes	No	No

Table 10-3—Summary of scope of global variables used by best master clock selection, external port configuration, and announce interval setting state machines (see 10.3.9 and 10.3.10) *(continued)*

Variable name	Subclause of definition	Per PTP Instance (i.e., per domain)	Per PTP Instance, per PTP Port	Instance used by CMLDS (i.e., variable is common across all LinkPorts)	Instance used by CMLDS, per LinkPort
annCurrentUtcOffset	10.3.10.22	No	Yes	No	No
annTimeSource	10.3.10.23	No	Yes	No	No
receivedPathTrace	10.3.10.24	No	Yes	No	No

10.3.9 Per PTP Instance global variables

10.3.9.1 reselect: A Boolean array of length numberPorts+1 (see 8.6.2.8). Setting reselect[j], where $0 \le j \le$ numberPorts, to TRUE causes the STATE_SELECTION block of the PortStateSelection state machine (see 10.3.13) to be re-entered, which in turn causes the PTP Port state of each PTP Port of the PTP Instance to be updated (via the function updtStatesTree(); see 10.3.13.2.4). This variable is used only by the BMCA, i.e., not by the explicit port state configuration option.

10.3.9.2 selected: A Boolean array of length numberPorts+1 (see 8.6.2.8). selected[j], where $0 \le j \le$ numberPorts, is set to TRUE immediately after the PTP Port states of all the ports are updated. This value indicates to the PortAnnounceInformation state machine (see 10.3.12) that it can update the portPriorityVector and other variables for each PTP Port. This variable is used by both the BMCA and the explicit port state configuration option; however, its value does not impact the explicit port state configuration option (see the NOTE in 10.3.16.3).

NOTE—Array elements 0 of the reselect and selected arrays are not used, except that the function clearReselectTree() sets reselect[0] to FALSE when it sets the entire array to zero and the function setSelectedTree() sets selected[0] to TRUE when it sets the entire array to TRUE. This action is taken only for convenience, so that array element j can correspond to PTP Port j. Note also that, in contrast, selectedState[0] is not used (see 10.2.4.20).

10.3.9.3 masterStepsRemoved: The value of stepsRemoved for the PTP Instance, after the PTP Port states of all the ports have been updated (see 10.3.13.2.4 for details on the computation of masterStepsRemoved). The data type for masterStepsRemoved is UInteger16. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.9.4 leap61: A Boolean variable whose value is TRUE if the last minute of the current UTC day, relative to the current Grandmaster Clock, contains 61 s and FALSE if the last minute of the current UTC day does not contain 61 s. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.9.5 leap59: A Boolean variable whose value is TRUE if the last minute of the current UTC day, relative to the current Grandmaster Clock, contains 59 s and FALSE if the last minute of the current UTC day does not contain 59 s. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.9.6 currentUtcOffsetValid: A Boolean variable whose value is TRUE if currentUtcOffset (see 10.3.9.10), relative to the current Grandmaster Clock, is known to be correct and FALSE if currentUtcOffset is not known to be correct. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.9.7 ptpTimescale: A Boolean variable whose value is TRUE if the timescale of the current Grandmaster Clock is PTP (see 8.2.1) and FALSE if the timescale is ARB. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.9.8 timeTraceable: A Boolean variable whose value is TRUE if both clockSlaveTime [i.e., the synchronized time maintained at the slave (see 10.2.4.3)] and currentUtcOffset (see 10.3.9.10), relative to the current Grandmaster Clock, are traceable to a primary reference and FALSE if one or both are not traceable to a primary reference. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.9.9 frequencyTraceable: A Boolean variable whose value is TRUE if the frequency that determines clockSlaveTime, i.e., the frequency of the LocalClockEntity multiplied by the most recently computed rateRatio by the PortSyncSyncReceive state machine (see 10.2.8.1.4), is traceable to a primary reference and FALSE if this frequency is not traceable to a primary reference. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.9.10 currentUtcOffset: The difference between TAI time and UTC time, i.e., TAI time minus UTC time, in seconds, and relative to the current Grandmaster Clock, when known. Otherwise, the value has no meaning (see 10.3.9.6). The data type for currentUtcOffset is Integer16. This variable is used by both the BMCA and the explicit port state configuration option.

NOTE—For example, 2006-01-01 00:00:00 UTC and 2006-01-01 00:00:33 TAI represent the same instant of time. At this time, currentUtcOffset was equal to 33 s.[13]

10.3.9.11 timeSource: The value of the timeSource attribute of the current Grandmaster PTP Instance. The data type for timeSource is TimeSource (see 8.6.2.7). This variable is used by both the BMCA and the explicit port state configuration option.

10.3.9.12 sysLeap61: A Boolean variable whose value is TRUE if the last minute of the current UTC day, relative to the ClockMaster entity of this PTP Instance, contains 61 s and FALSE if the last minute of the current UTC day does not contain 61 s. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.9.13 sysLeap59: A Boolean variable whose value is TRUE if the last minute of the current UTC day, relative to the ClockMaster entity of this PTP Instance, contains 59 s and FALSE if the last minute of the current UTC day does not contain 59 s. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.9.14 sysCurrentUtcOffsetValid: A Boolean variable whose value is TRUE if currentUtcOffset (see 10.3.9.10), relative to the ClockMaster entity of this PTP Instance, is known to be correct and FALSE if currentUtcOffset is not known to be correct. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.9.15 sysPtpTimescale: A Boolean variable whose value is TRUE if the timescale of the ClockMaster entity of this PTP Instance is PTP (see 8.2.1) and FALSE if the timescale of the ClockMaster entity of this PTP Instance is ARB. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.9.16 sysTimeTraceable: A Boolean variable whose value is TRUE if both masterTime [i.e., the time maintained by the ClockMaster entity of this PTP Instance (see 10.2.4.21)] and currentUtcOffset (see 10.3.9.10), relative to the ClockMaster entity of this PTP Instance, are traceable to a primary reference and

[13] Note also that a leap second was not added at the end of the last UTC minute of 2005-12-31.

FALSE if one or both are not traceable to a primary reference. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.9.17 sysFrequencyTraceable: A Boolean variable whose value is TRUE if the frequency that determines masterTime of the ClockMaster entity of this PTP Instance, i.e., the frequency of the LocalClockEntity multiplied by the most recently computed gmRateRatio by the ClockMasterSyncReceive state machine (see 10.2.4.14 and 10.2.11), is traceable to a primary reference and FALSE if this frequency is not traceable to a primary reference. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.9.18 sysCurrentUtcOffset: The difference between TAI time and UTC time, i.e., TAI time minus UTC time, in seconds, and relative to the ClockMaster entity of this PTP Instance, when known. Otherwise, the value has no meaning (see 10.3.9.14). The data type for sysCurrentUtcOffset is Integer16. This variable is used by both the BMCA and the explicit port state configuration option.

NOTE—See the NOTE in 10.3.9.10 for more detail on the sign convention.

10.3.9.19 sysTimeSource: The value of the timeSource attribute of the ClockMaster entity of this PTP Instance (see 8.6.2.7). The data type for sysTimeSource is TimeSource.

10.3.9.20 systemPriority: The systemPriority vector for this PTP Instance. The data type for systemPriority is UInteger224 (see 10.3.5).

10.3.9.21 gmPriority: The current gmPriorityVector for the PTP Instance. The data type for gmPriority is UInteger224 (see 10.3.5).

10.3.9.22 lastGmPriority: The previous gmPriorityVector for the PTP Instance, prior to the most recent invocation of the PortStateSelection state machine. The data type for lastGmPriority is UInteger224 (see 10.3.4). lastGmPriority is used only by the BMCA, i.e., not by the explicit port state configuration option.

10.3.9.23 pathTrace: An array that contains the clockIdentities of the successive PTP Instances that receive, process, and send Announce messages. The data type for pathTrace is ClockIdentity[N], where N is the number of PTP Instances, including the Grandmaster PTP Instance, that the Announce information has traversed. This variable is used by both the BMCA and the explicit port state configuration option.

NOTE 1—N is equal to stepsRemoved+1 (see 10.6.3.2.6). The size of the pathTrace array can change after each reception of an Announce message, up to the maximum size for the respective medium. For example, the maximum value of N for a full-duplex IEEE 802.3 medium is 179. This is obtained from the fact that the number of PTP octets in an Announce message is 68 + 8N, where N is the number of entries in the pathTrace array (see 10.6.3.1 and Table 10-11), and the maximum payload size for full-duplex IEEE 802.3 media is 1500 octets. Setting 68 + 8N = 1500, and solving for N gives N = 179.

NOTE 2—The current behavior for the path trace feature is documented in 10.3.11.2.1 and 10.3.16.2.1 and is as follows:

— Item c) of 10.3.11.2.1, the description of the qualifyAnnounce() function of the PortAnnounceReceive state machine, indicates that if a path trace TLV is present and one of the elements of the pathSequence array field is equal to the clockIdentity of the clock where the TLV is being processed, the Announce message is not qualified.
— Item d) of 10.3.11.2.1 (qualifyAnnounce() function) indicates that if the Announce message is qualified and a path trace TLV is present, the pathSequence array of the TLV is copied to the pathTrace array (described in this subclause) and the clockIdentity of the PTP Instance that processes the Announce message is appended to the array. However, if a path trace TLV is not present, the path trace array is empty.

— Item f) of 10.3.16.2.1, the description of the txAnnounce() function of the PortAnnounceTransmit state machine, indicates that a path trace TLV is constructed and appended to an Announce message just before the Announce message is transmitted only if the pathTrace array is not empty and appending the TLV does not cause the media-dependent layer frame to exceed any respective maximum size. If appending the TLV does cause a respective maximum frame size to be exceeded or if the pathTrace array is empty, the TLV is not appended.
— As a result of the behaviors of the qualifyAnnounce() and txAnnounce() functions described in this note, the path trace feature is not used, i.e., a path trace TLV is not appended to an Announce message and the pathTrace array is empty, once appending a clockIdentity to the TLV would cause the frame carrying the Announce message to exceed its maximum size.

NOTE 3—Once the value of stepsRemoved of an Announce message reaches 255, the Announce message is not qualified [see item b) of 10.3.11.2.1].

10.3.9.24 externalPortConfigurationEnabled: A variable whose value indicates whether PTP Port states are externally configured or determined by the BMCA. The data type shall be Boolean. The value TRUE indicates that the PTP Port states are externally configured; the value FALSE indicates that the PTP Port states are determined by the BMCA. This variable is used by both the BMCA and the external port configuration option.

10.3.9.25 lastAnnouncePort: The PTP Port number of the PTP Port on which the most recent Announce message was received. This variable is used by the PortAnnounceInformationExt and PortStateSettingExt state machines for the external port configuration option. This variable is not used by the BMCA. The data type for this variable is UInteger16.

10.3.10 Per-port global variables

10.3.10.1 announceReceiptTimeoutTimeInterval: The time interval after which announce receipt timeout occurs if an Announce message has not been received during the interval. The value of announceReceiptTimeoutTimeInterval is equal to announceReceiptTimeout (see 10.7.3.2) multiplied by the announceInterval (see 10.3.10.8) for the PTP Port at the other end of the link to which this PTP Port is attached. The value of announceInterval for the PTP Port at the other end of the link is computed from logMessageInterval of the received Announce message (see 10.6.2.2.14). The data type for announceReceiptTimeoutTimeInterval is UScaledNs. This variable is used only by the BMCA, i.e., not by the explicit port state configuration option.

10.3.10.2 announceSlowdown: A Boolean that is set to TRUE if the AnnounceIntervalSetting state machine (see Figure 10-19 in item 10.3.17.3) receives a TLV that requests a larger Announce message transmission interval (see 10.7.2.2) and FALSE otherwise. When announceSlowdown is set to TRUE, the PortAnnounceTransmit state machine (see Figure 10-18) continues to send Announce messages at the old (i.e., faster) rate until a number of Announce messages equal to announceReceiptTimeout (see 10.7.3.2) have been sent, but with the logMessageInterval field of the PTP common header set equal to the new announce interval (i.e., corresponding to the slower rate). After announceReceiptTimeout Announce messages have been sent, subsequent Announce messages are sent at the new (i.e., slower) rate and with the logMessageInterval field of the PTP common header set to the new announce interval. This variable is used by both the BMCA and the explicit port state configuration option. When announceSlowdown is set to FALSE, the PortAnnounceTransmit state machine immediately sends Announce messages at the new (i.e., slower) rate.

NOTE—If a receiver of Announce messages requests a slower rate, the receiver will continue to use the upstream announceInterval value, which it obtains from the logMessageInterval field of received Announce messages, until it receives an Announce message where that value has changed. If, immediately after requesting a slower Announce message rate, up to announceReceiptTimeout minus one consecutive Announce messages sent to the receiver are lost, announce receipt timeout could occur if the sender had changed to the slower rate immediately. Delaying the slowing down of the sending rate of Announce messages for announceReceiptTimeout messages prevents announce receipt timeout from occurring until at least announceReceiptTimeout Announce messages have been lost. Note that networks with high packet loss can still experience announce receipt timeout under high-packet-loss conditions; however, the announce receipt timeout condition occurs only after at least announceReceiptTimeout Announce messages have been lost.

10.3.10.3 oldAnnounceInterval: The saved value of the previous announce interval, when a new announce interval is requested via a Signaling message that contains a message interval request TLV. The data type for oldAnnounceInterval is UScaledNs. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.4 infoIs: An Enumeration2 that takes the values Received, Mine, Aged, or Disabled to indicate the origin and state of the PTP Port’s time-synchronization spanning tree information:

a) If infoIs is Received, the PTP Port has received current information (i.e., announce receipt timeout has not occurred and, if gmPresent is TRUE, sync receipt timeout also has not occurred) from the master PTP Instance for the attached gPTP communication path.
b) If infoIs is Mine, information for the PTP Port has been derived from the SlavePort for the PTP Instance (with the addition of SlavePort stepsRemoved). This includes the possibility that the SlavePort is the PTP Port whose portNumber is 0, i.e., the PTP Instance is the root of the gPTP domain.
c) If infoIs is Aged, announce receipt timeout or, when gmPresent is TRUE, sync receipt timeout has occurred.
d) If portOper, ptpPortEnabled, and asCapable are not all TRUE, infoIs is Disabled.

The variable infoIs is used only by the BMCA, i.e., not by the explicit port state configuration option.

10.3.10.5 masterPriority: The masterPriorityVector for the PTP Port. The data type for masterPriority is UInteger224 (see 10.3.4). This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.6 currentLogAnnounceInterval: The current value of the logarithm to base 2 of the mean time interval, in seconds, between the sending of successive Announce messages (see 10.7.2.2). This value is set in the AnnounceIntervalSetting state machine (see 10.3.17). The data type for currentLogAnnounceInterval is Integer8. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.7 initialLogAnnounceInterval: The initial value of the logarithm to base 2 of the mean time interval, in seconds, between the sending of successive Announce messages (see 10.7.2.2). The data type for initialLogAnnounceInterval is Integer8. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.8 announceInterval: A variable containing the mean Announce message transmission interval for the PTP Port. This value is set in the AnnounceIntervalSetting state machine (see 10.3.17). The data type for announceInterval is UScaledNs. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.9 messageStepsRemoved: The value of stepsRemoved contained in the received Announce information. The data type for messageStepsRemoved is UInteger16. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.10 newInfo: A Boolean variable that is set to cause a PTP Port to transmit Announce information; specifically, it is set when an announce interval has elapsed (see Figure 10-18), PTP Port states have been updated, and portPriority and portStepsRemoved information has been updated with newly determined masterPriority and masterStepsRemoved information. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.11 portPriority: The portPriorityVector for the PTP Port. The data type for portPriority is UInteger224 (see 10.3.4). This variable is used only by the BMCA, i.e., not by the explicit port state configuration option.

10.3.10.12 portStepsRemoved: The value of stepsRemoved for the PTP Port. portStepsRemoved is set equal to masterStepsRemoved (see 10.3.9.3) after masterStepsRemoved is updated. The data type for portStepsRemoved is UInteger16. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.13 rcvdAnnouncePtr: A pointer to a structure that contains the fields of a received Announce message. This variable is used by both the BMCA and the explicit PTP Port state configuration option.

10.3.10.14 rcvdMsg: A Boolean variable that is TRUE if a received Announce message is qualified and FALSE if it is not qualified. This variable is used only by the BMCA, i.e., not by the explicit port state configuration option.

10.3.10.15 updtInfo: A Boolean variable that is set to TRUE to indicate that the PortAnnounceInformation state machine (see 10.3.12) should copy the newly determined masterPriority and masterStepsRemoved to portPriority and portStepsRemoved, respectively. This variable is used by both the BMCA and the explicit port state configuration option; however, its value does not impact the explicit port state configuration option (see the NOTE in 10.3.16.3).

10.3.10.16 annLeap61: A global variable in which the leap61 flag (see 10.6.2.2.8) of a received Announce message is saved. The data type for annLeap61 is Boolean. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.17 annLeap59: A global variable in which the leap59 flag (see 10.6.2.2.8) of a received Announce message is saved. The data type for annLeap59 is Boolean. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.18 annCurrentUtcOffsetValid: A global variable in which the currentUtcOffsetValid flag (see 10.6.2.2.8) of a received Announce message is saved. The data type for annCurrentUtcOffsetValid is Boolean. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.19 annPtpTimescale: A global variable in which the ptpTimescale flag (see 10.6.2.2.8) of a received Announce message is saved. The data type for annPtpTimescale is Boolean. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.20 annTimeTraceable: A global variable in which the timeTraceable flag (see 10.6.2.2.8) of a received Announce message is saved. The data type for annTimeTraceable is Boolean. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.21 annFrequencyTraceable: A global variable in which the frequencyTraceable flag (see 10.6.2.2.8) of a received Announce message is saved. The data type for annFrequencyTraceable is Boolean. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.22 annCurrentUtcOffset: A global variable in which the currentUtcOffset field (see 10.6.3.2.1) of a received Announce message is saved. The data type for annCurrentUtcOffset is Integer16. This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.23 annTimeSource: A global variable in which the timeSource field (see 10.6.3.2.1) of a received Announce message is saved. The data type for annTimeSource is TimeSource (see 8.6.2.7). This variable is used by both the BMCA and the explicit port state configuration option.

10.3.10.24 receivedPathTrace: An array in which the pathSequence array field of the path trace TLV of the most recently received Announce message is saved. The data type for receivedPathTrace is clockIdentity[N], where N is the number of entries in the pathSequence array field.

10.3.11 PortAnnounceReceive state machine

10.3.11.1 State machine variables

The following variable is used in the state diagram in Figure 10-13 (in 10.3.11.3):

10.3.11.1.1 rcvdAnnouncePAR: A Boolean variable that notifies the current state machine when Announce message information is received from the MD entity of the same PTP Port. This variable is reset by this state machine.

10.3.11.2 State machine functions

10.3.11.2.1 qualifyAnnounce (rcvdAnnouncePtr): Qualifies the received Announce message pointed to by rcvdAnnouncePtr as follows:

a) If the Announce message was sent by the current PTP Instance, i.e., if sourcePortIdentity.clockIdentity (see 10.6.2.2.11 and 8.5.2) is equal to thisClock (see 10.2.4.22), the Announce message is not qualified, and FALSE is returned;
b) If the stepsRemoved field is greater than or equal to 255, the Announce message is not qualified, and FALSE is returned;
c) If a path trace TLV is present and one of the elements of the pathSequence array field of the path trace TLV is equal to thisClock (i.e., the clockIdentity of the current PTP Instance; see 10.2.4.22), the Announce message is not qualified, and FALSE is returned;
d) Otherwise, the Announce message is qualified, and TRUE is returned. If a path trace TLV is present, it is saved in the per port global variable receivedPathTrace. If a path trace TLV is not present, the per port global variable receivedPathTrace is set to the empty array.

10.3.11.3 State diagram

The PortAnnounceReceive state machine shall implement the function specified by the state diagram in Figure 10-13, the local variable specified in 10.3.11.1, the function specified in 10.3.11.2, and the relevant global variables specified in 10.2.4, 10.2.5, 10.3.9, 10.3.10, and 11.2.13. The state machine is not used if externalPortConfigurationEnabled is TRUE. The state machine receives Announce information from the MD entity of the same PTP Port, determines if the Announce message is qualified, and if so, sets the rcvdMsg variable.

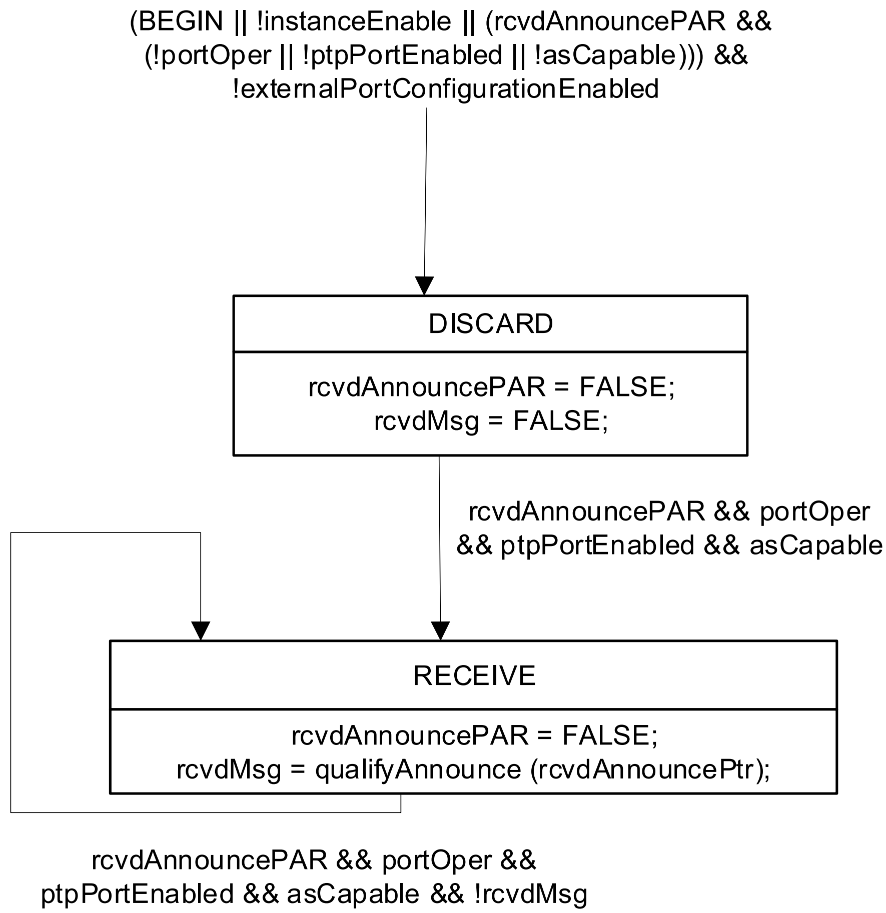


Figure 10-13—PortAnnounceReceive state machine

10.3.12 PortAnnounceInformation state machine

10.3.12.1 State machine variables

The following variables are used in the state diagram in Figure 10-14 (in 10.3.12.3):

10.3.12.1.1 announceReceiptTimeoutTime: A variable used to save the time at which announce receipt timeout occurs. The data type for announceReceiptTimeoutTime is UScaledNs.

10.3.12.1.2 messagePriorityPAI: The messagePriorityVector corresponding to the received Announce information. The data type for messagePriorityPAI is UInteger224 (see 10.3.4).

10.3.12.1.3 rcvdInfo: An Enumeration2 that holds the value returned by rcvInfo() (see 10.3.12.2.1).

10.3.12.2 State machine functions

10.3.12.2.1 rcvInfo (rcvdAnnouncePtr): Decodes the messagePriorityVector (see 10.3.4 and 10.3.5) and stepsRemoved 10.6.3.2.6) field from the Announce information pointed to by rcvdAnnouncePtr (see 10.3.10.13), and then:

a) Stores the messagePriorityVector and stepsRemoved field value in messagePriorityPAI and messageStepsRemoved, respectively, and then:
    1) If the received message conveys the PTP Port state MasterPort and the messagePriorityVector is the same as the portPriorityVector of the PTP Port, returns RepeatedMasterInfo; else
    2) If the received message conveys the PTP Port state MasterPort and the messagePriorityVector is superior to the portPriorityVector of the PTP Port, returns SuperiorMasterInfo; else
    3) If the received message conveys the PTP Port state MasterPort, and the messagePriorityVector is worse than the portPriorityVector of the PTP Port, returns InferiorMasterInfo; else
    4) Returns OtherInfo.

NOTE—In accordance with 10.3.5, the messagePriorityVector is superior to the portPriorityVector of the PTP Port if, and only if, the messagePriorityVector is better than the portPriorityVector, or the Announce message has been transmitted from the same master PTP Instance and MasterPort as the portPriorityVector. In steps a) 1) to a) 4) in this subclause, rcvInfo() first checks whether the messagePriorityVector and portPriorityVector are the same (and the received message conveys the PTP Port state MasterPort), before checking whether the messagePriorityVector is superior to the portPriorityVector. The reason for this sequence is that RepeatedMasterInfo needs to be returned if the messagePriorityVector and portPriorityVector are the same, while SuperiorMasterInfo needs to be returned in other instances where the Announce message has been transmitted from the same master PTP Instance and MasterPort as the portPriorityVector (if the test for SuperiorMasterInfo were done before the test for RepeatedMasterInfo, SuperiorMasterInfo would be returned when RepeatedMasterInfo is desired).

10.3.12.2.2 recordOtherAnnounceInfo(): Saves the flags leap61, leap59, currentUtcOffsetValid, ptpTimescale, timeTraceable, and frequencyTraceable, and the fields currentUtcOffset and timeSource, of the received Announce message for this PTP Port. The values are saved in the per-PTP Port global variables annLeap61, annLeap59, annCurrentUtcOffsetValid, annPtpTimescale, annTimeTraceable, annFrequencyTraceable, annCurrentUtcOffset, and annTimeSource (see 10.3.10.16 through 10.3.10.23).

10.3.12.3 State diagram

The PortAnnounceInformation state machine shall implement the function specified by the state diagram in Figure 10-14, the local variables specified in 10.3.12.1, the functions specified in 10.3.12.2, and the relevant global variables specified in 10.2.4, 10.2.5, 10.3.9, 10.3.10, and 11.2.13. This state machine is used only if externalPortConfigurationEnabled is FALSE (if this variable is TRUE, the PortAnnounceInformationExt state machine is used instead). The state machine receives new qualified Announce information from the PortAnnounceReceive state machine (see 10.3.11) of the same PTP Port and determines if the Announce information is better than the current best master information it knows. The state machine also updates the current best master information when it receives updated PTP Port state information from the PortStateSelection state machine (see 10.3.13) and when announce receipt timeout or, when gmPresent is TRUE, sync receipt timeout occurs.

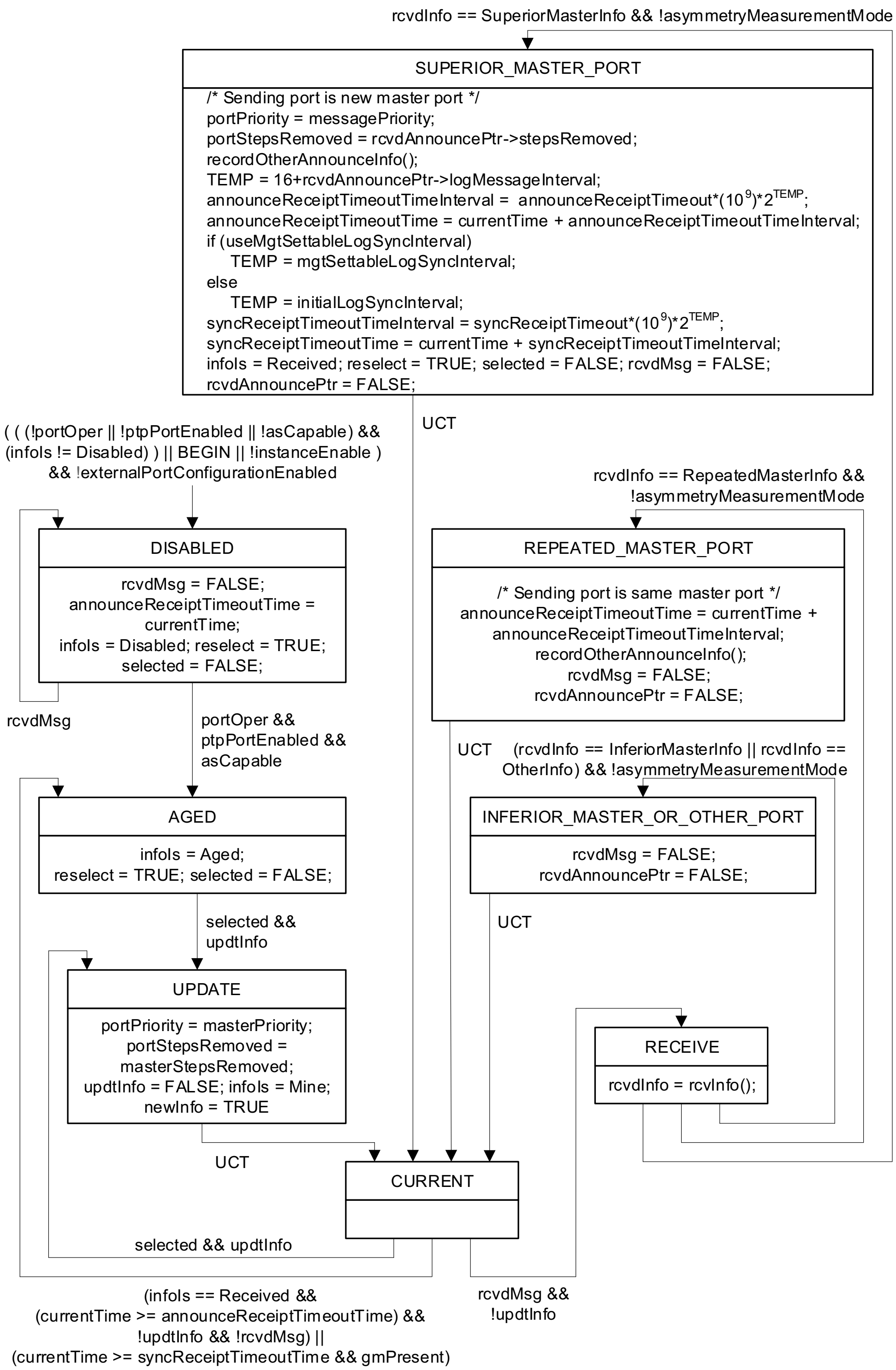


Figure 10-14—PortAnnounceInformation state machine

10.3.13 PortStateSelection state machine

10.3.13.1 State machine variables

The following variables are used in the state diagram in Figure 10-15 (in 10.3.13.3):

10.3.13.1.1 systemIdentityChange: A Boolean variable that notifies the current state machine when the systemIdentity (see 10.3.2) of this PTP Instance has changed. This variable is reset by this state machine.

The systemIdentity changes when at least one of the attributes priority1, clockClass, clockAccuracy, offsetScaledLogVariance, and priority2 changes (e.g., due to management action, degradation or loss of the ClockSource). The systemIdentity also includes the attribute clockIdentity, but this attribute does not change.

10.3.13.1.2 asymmetryMeasurementModeChange: A Boolean variable that notifies the current state machine when the per-port variable asymmetryMeasurementMode (see 10.2.5.2) changes on any port. This variable is reset by this state machine. There is one instance of asymmetryMeasurementModeChange for all the domains (per port). The variable is accessible by all the domains.

10.3.13.2 State machine functions

10.3.13.2.1 updtStateDisabledTree(): Sets all the elements of the selectedState array (see 10.2.4.20) to DisabledPort. Sets lastGmPriority to all ones. Sets the pathTrace array (see 10.3.9.23) to contain the single element thisClock (see 10.2.4.22).

10.3.13.2.2 clearReselectTree(): Sets all the elements of the reselect array (see 10.3.9.1) to FALSE.

10.3.13.2.3 setSelectedTree(): Sets all the elements of the selected array (see 10.3.9.2) to TRUE.

10.3.13.2.4 updtStatesTree(): Performs the following operations (see 10.3.4 and 10.3.5 for details on the priority vectors):

a) Computes the gmPathPriorityVector for each PTP Port that has a portPriorityVector and for which neither announce receipt timeout nor, if gmPresent is TRUE, sync receipt timeout have occurred,
b) Saves gmPriority (see 10.3.9.21) in lastGmPriority (see 10.3.9.22), computes the gmPriorityVector for the PTP Instance and saves it in gmPriority, chosen as the best of the set consisting of the systemPriorityVector (for this PTP Instance) and the gmPathPriorityVector for each PTP Port for which the clockIdentity of the master port is not equal to thisClock (see 10.2.4.22),
c) Sets the per PTP Instance global variables leap61, leap59, currentUtcOffsetValid, ptpTimescale, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource as follows:
    1) If the gmPriorityVector was set to the gmPathPriorityVector of one of the ports, then leap61, leap59, currentUtcOffsetValid, ptpTimescale, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource are set to annLeap61, annLeap59, annCurrentUtcOffsetValid, annPtpTimescale, annTimeTraceable, annFrequencyTraceable, annCurrentUtcOffset, and annTimeSource, respectively, for that PTP Port.
    2) If the gmPriorityVector was set to the systemPriorityVector, then leap61, leap59, currentUtcOffsetValid, ptpTimescale, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource are set to sysLeap61, sysLeap59, sysCurrentUtcOffsetValid, sysPtpTimescale, sysTimeTraceable, sysFrequencyTraceable, sysCurrentUtcOffset, and sysTimeSource, respectively.

d) Computes the masterPriorityVector for each PTP Port.

e) Computes masterStepsRemoved, which is equal to one of the following:

   1) messageStepsRemoved (see 10.3.10.9) for the PTP Port associated with the gmPriorityVector, incremented by 1, if the gmPriorityVector is not the systemPriorityVector, or

   2) 0 if the gmPriorityVector is the systemPriorityVector.

f) Assigns the PTP Port state for PTP Port j, and sets selectedState[j] equal to this PTP Port state, as follows, for j = 1, 2, ..., numberPorts:

   1) If the PTP Port is disabled (infoIs == Disabled), then selectedState[j] is set to DisabledPort.

   2) If asymmetryMeasurementMode is TRUE, then selectedState[j] is set to PassivePort, and updtInfo is set to FALSE.

   3) If announce receipt timeout, or sync receipt timeout with gmPresent set to TRUE, has occurred (infoIs = Aged), then selectedState[j] is set to MasterPort, and updtInfo is set to TRUE.

   4) If the portPriorityVector was derived from another PTP Port on the PTP Instance or from the PTP Instance itself as the root (infoIs == Mine), then selectedState[j] is set to MasterPort. In addition, updtInfo is set to TRUE if the portPriorityVector differs from the masterPriorityVector or portStepsRemoved differs from masterStepsRemoved.

   5) If the portPriorityVector was received in an Announce message, announce receipt timeout, or sync receipt timeout with gmPresent TRUE, has not occurred (infoIs == Received), and the gmPriorityVector is now derived from the portPriorityVector, then selectedState[j] is set to SlavePort, and updtInfo is set to FALSE. The per port global variable receivedPathTrace, for this port, is copied to the per PTP Instance global array pathTrace, and, if it is not empty, thisClock is appended to pathTrace.

   6) If the portPriorityVector was received in an Announce message, announce receipt timeout, or sync receipt timeout with gmPresent TRUE, has not occurred (infoIs == Received), the gmPriorityVector is not now derived from the portPriorityVector, the masterPriorityVector is not better than the portPriorityVector, and the sourcePortIdentity component of the portPriorityVector *does not* reflect another PTP Port on the PTP Instance, then selectedState[j] is set to PassivePort, and updtInfo is set to FALSE.

   7) If the portPriorityVector was received in an Announce message, announce receipt timeout, or sync receipt timeout with gmPresent TRUE, has not occurred (infoIs == Received), the gmPriorityVector is not now derived from the portPriorityVector, the masterPriorityVector is not better than the portPriorityVector, and the sourcePortIdentity component of the portPriorityVector *does* reflect another PTP Port on the PTP Instance, then selectedState[j] set to PassivePort, and updtInfo is set to FALSE.

   8) If the portPriorityVector was received in an Announce message, announce receipt timeout, or sync receipt timeout with gmPresent TRUE, has not occurred (infoIs == Received), the gmPriorityVector is not now derived from the portPriorityVector, and the masterPriorityVector is better than the portPriorityVector, then selectedState[j]is set to MasterPort, and updtInfo is set to TRUE.

g) Updates gmPresent as follows:

   1) gmPresent is set to TRUE if the priority1 field of the rootSystemIdentity of the gmPriorityVector is less than 255.

   2) gmPresent is set to FALSE if the priority1 field of the rootSystemIdentity of the gmPriorityVector is equal to 255.

h) Assigns the PTP Port state for PTP Port 0 (see 8.5.2.3), and sets selectedState[0] as follows:

   1) if selectedState[j] is set to SlavePort for any PTP Port with portNumber j, j = 1, 2, ..., numberPorts, selectedState[0] is set to PassivePort.

   2) if selectedState[j] is *not* set to SlavePort for any PTP Port with portNumber j, j = 1, 2, ..., numberPorts, selectedState[0] is set to SlavePort.

i) If the clockIdentity member of the systemIdentity (see 10.3.2) member of gmPriority (see 10.3.9.21) is equal to thisClock (see 10.2.4.22), i.e., if the current PTP Instance is the Grandmaster PTP Instance, the pathTrace array is set to contain the single element thisClock (see 10.2.4.22).

10.3.13.3 State diagram

The PortStateSelection state machine shall implement the function specified by the state diagram in Figure 10-15, the functions specified in 10.3.13.1, and the relevant global variables specified in 10.2.4, 10.2.5, 10.3.9, 10.3.10, and 11.2.13. This state machine is used only if externalPortConfigurationEnabled is FALSE (if this variable is TRUE, the PortStateSettingExt state machine is used instead). The state machine updates the gmPathPriority vector for each PTP Port of the PTP Instance, the gmPriorityVector for the PTP Instance, and the masterPriorityVector for each PTP Port of the PTP Instance. The state machine determines the PTP Port state for each PTP Port and updates gmPresent.

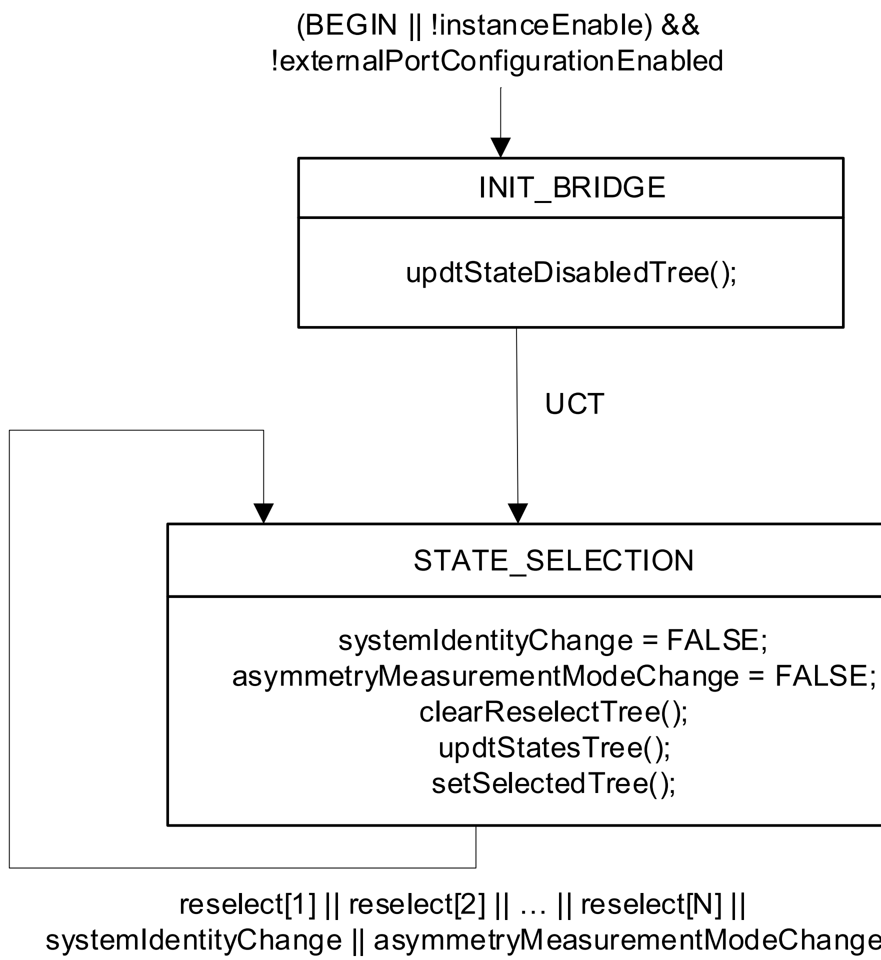


Figure 10-15—PortStateSelection state machine

10.3.14 PortAnnounceInformationExt state machine

10.3.14.1 State machine variables

The following variables are used in the state diagram in Figure 10-16 (in 10.3.14.3):

10.3.14.1.1 rcvdAnnouncePAIE: A Boolean variable that notifies the current state machine when Announce message information is received from the MD entity of the same PTP Port. This variable is reset by this state machine.

10.3.14.1.2 messagePriorityPAIE: The messagePriorityVector corresponding to the received Announce information. The data type for messagePriorityPAIE is UInteger224 (see 10.3.4).

10.3.14.2 State machine functions

10.3.14.2.1 rcvInfoExt (rcvdAnnouncePtr): Decodes the messagePriorityVector (see 10.3.4 and 10.3.5) and stepsRemoved 10.6.3.2.6) field from the Announce information pointed to by rcvdAnnouncePtr (see 10.3.10.13), and then stores the messagePriorityVector and stepsRemoved field value in messagePriorityPAIE and messageStepsRemoved, respectively. If a path trace TLV is present in the Announce message and the portState of the PTP Port is SlavePort, the pathSequence array field of the TLV is copied to the global array pathTrace, and thisClock is appended to pathTrace (i.e., is added to the end of the array).

10.3.14.2.2 recordOtherAnnounceInfo(): Saves the flags leap61, leap59, currentUtcOffsetValid, ptpTimescale, timeTraceable, and frequencyTraceable, and the fields currentUtcOffset and timeSource, of the received Announce message for this PTP Port. The values are saved in the per-PTP Port global variables annLeap61, annLeap59, annCurrentUtcOffsetValid, annPtpTimescale, annTimeTraceable, annFrequencyTraceable, annCurrentUtcOffset, and annTimeSource (see 10.3.10.16 through 10.3.10.23).

10.3.14.3 State diagram

The PortAnnounceInformationExt state machine shall implement the function specified by the state diagram in Figure 10-16, the local variables specified in 10.3.14.1, the functions specified in 10.3.14.2, and the relevant global variables specified in 10.2.4, 10.2.5, 10.3.9, 10.3.10, and 11.2.13. This state machine is used only if externalPortConfigurationEnabled is TRUE (if this variable is FALSE, the PortAnnounceInformation state machine of 10.3.12.3 is used instead). The state machine receives Announce information from the MD entity of the same PTP Port and saves the information.

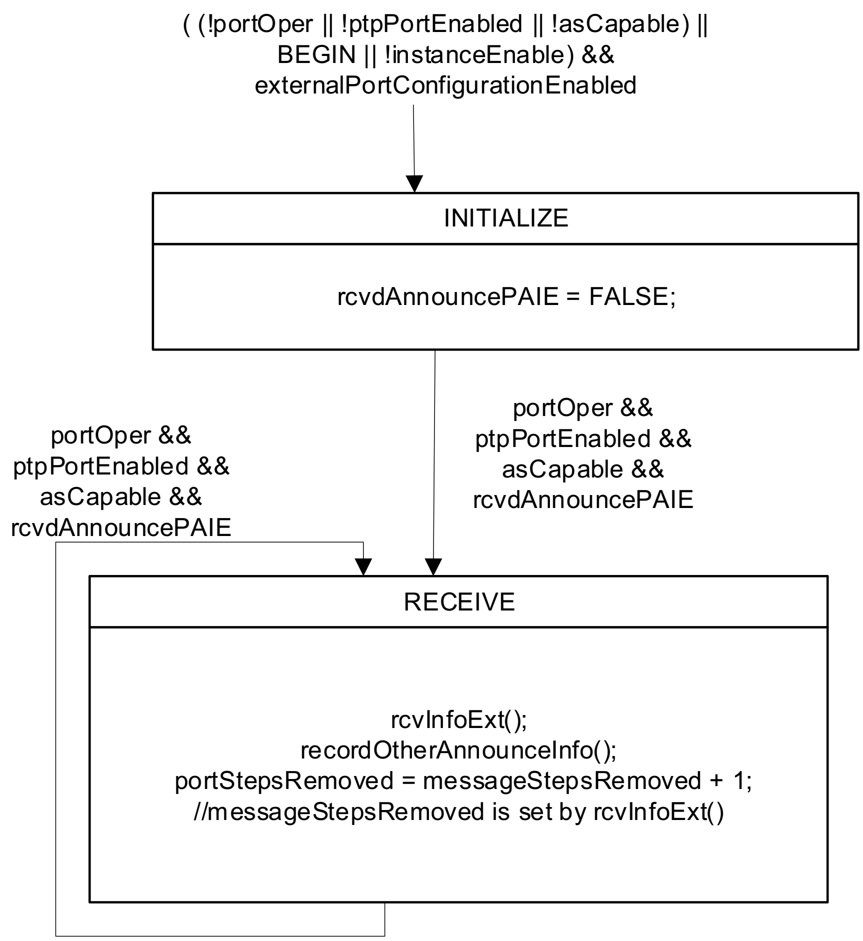


Figure 10-16—PortAnnounceInformationExt state machine

10.3.15 PortStateSettingExt state machine

10.3.15.1 State machine variables

The following variables are used in the state diagram in Figure 10-17 (in 10.3.15.3):

10.3.15.1.1 disabledExt: A Boolean variable that notifies the current state machine (i.e., when it is set to TRUE) when at least one of the variables portOper, ptpPortEnabled, or asCapable, for this PTP Port, has changed from TRUE to FALSE. This variable is reset by this state machine.

10.3.15.1.2 reenabledExt: A Boolean variable that notifies the current state machine (i.e., when it is set to TRUE) when all of the variables portOper, ptpPortEnabled, and asCapable, for this PTP Port, that are FALSE have changed to TRUE. This variable is reset by this state machine.

10.3.15.1.3 a**symmetryMeasurementModeChangeThisPort:** A Boolean variable that notifies the current state machine when the per-port variable asymmetryMeasurementMode (see 10.2.5.2) changes on this port. This variable is reset by this state machine. There is one instance of asymmetryMeasurementModeChangeThisPort for all the domains (per port). The variable is accessible by all the domains.

10.3.15.1.4 rcvdPortStateInd: A Boolean variable that notifies the current state machine (i.e., when it is set to TRUE) when the PTP Port state of this PTP Port has been externally set. This variable is reset by this state machine.

10.3.15.1.5 portStateInd: An Enumeration2 that indicates the PTP Port state that has been set. The values are MasterPort, SlavePort, and PassivePort.

NOTE—The PTP Port state can be externally set to DisabledPort by setting portOper or ptpPortEnabled to FALSE. The PTP Port state is set to DisabledPort when asCapable becomes FALSE.

10.3.15.2 State machine functions

10.3.15.2.1 resetStateTree(j): Sets selectedState[j] (see 10.2.4.20) to externalPortConfigurationPortDS.desiredState. Sets the pathTrace array (see 10.3.9.23) to contain the single element thisClock (see 10.2.4.22) if no PTP Port of the PTP Instance has the PTP Port state SlavePort.

10.3.15.2.2 updtPortState(j): Performs the following operations for PTP Port j (see 10.3.4 and 10.3.5 for details on the priority vectors):

a) Sets the per PTP Instance global variables leap61, leap59, currentUtcOffsetValid, ptpTimescale, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource as follows:
    1) If the PTP Port state of any PTP Port of this PTP Instance other than PTP Port 0 (see 8.5.2.3) is SlavePort, then leap61, leap59, currentUtcOffsetValid, ptpTimescale, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource are set to annLeap61, annLeap59, annCurrentUtcOffsetValid, annPtpTimescale, annTimeTraceable, annFrequencyTraceable, annCurrentUtcOffset, and annTimeSource, respectively, for that PTP Port.

2) If no PTP Port of this PTP Instance other than PTP Port 0 has the PTP Port state SlavePort, then leap61, leap59, currentUtcOffsetValid, ptpTimescale, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource are set to sysLeap61, sysLeap59, sysCurrentUtcOffsetValid, sysPtpTimescale, sysTimeTraceable, sysFrequencyTraceable, sysCurrentUtcOffset, and sysTimeSource, respectively.

b) Computes masterStepsRemoved as follows:
   1) If the PTP Port state of any PTP Port of this PTP Instance other than PTP Port 0 is SlavePort, then masterStepsRemoved is set equal to portStepsRemoved for that PTP Port.
   2) If no PTP Port of this PTP Instance other than PTP Port 0 has the PTP Port state SlavePort, then masterStepsRemoved is set equal to 0.
c) Assigns the PTP Port state for PTP Port j, and sets selectedState[j] equal to this PTP Port state, as follows:
   1) If disabledExt is TRUE, selectedState[j] is set to DisabledPort, else
   2) If asymmetryMeasurementMode is TRUE, selectedState[j] is set to PassivePort, else
   3) selectedState[j] is set to portStateInd.
d) Updates gmPresent as follows:
   1) If the PTP Port state of any PTP Port of this PTP Instance other than PTP Port 0 is SlavePort and the priority1 field of the rootSystemIdentity of the messagePriorityPAIE of the slave port is less than 255, gmPresent is set to TRUE, else
   2) If the PTP Port state of any PTP Port of this PTP Instance other than PTP Port 0 is SlavePort and the priority1 field of the rootSystemIdentity of the messagePriorityPAIE of the slave PTP Port is equal to 255, gmPresent is set to FALSE, else
   3) If no PTP Port of this PTP Instance other than PTP Port 0 has the PTP Port state SlavePort, gmPresent is set to TRUE if priority1 for this PTP Instance is less than 255 and FALSE if priority1 for this PTP Instance is equal to 255.
e) Assigns the PTP Port state for PTP Port 0, and sets selectedState[0] as follows:
   1) If selectedState[j] is set to SlavePort, selectedState[0] is set to PassivePort.
   2) If selectedState[j] is *not* set to SlavePort and selectedState[k] is not equal to SlavePort for every k not equal to 0 or j, selectedState[0] is set to SlavePort.
f) Computes the gmPriorityVector as follows:
   1) If selectedState[j] is set to SlavePort, the gmPriorityVector is set equal to messagePriorityPAIE for PTP Port j.
   2) If selectedState[j] is *not* set to SlavePort and selectedState[k] is not equal to SlavePort for every k not equal to 0 or j, the gmPriorityVector is set equal to the systemPriorityVector.
g) Computes the masterPriorityVector for PTP Port j.
h) If no PTP Port of this PTP Instance has the PTP Port state SlavePort, the pathTrace array is set to contain the single element thisClock (see 10.2.4.22).

10.3.15.3 State diagram

The PortStateSettingExt state machine shall implement the function specified by the state diagram in Figure 10-17, the local variables specified in 10.3.15.1, the functions specified in 10.3.15.2, and the relevant global variables specified in 10.2.4, 10.2.5, 10.3.9, 10.3.10, and 11.2.13. This state machine is used only if externalPortConfigurationEnabled is TRUE (if this variable is FALSE, the PortStateSelection state machine of 10.3.13.3 is used instead). A separate instance of this state machine runs on each PTP Port (unlike the PortStateSelection state machine, for which a single instance runs in the PTP Instance and performs operations on all the ports).

The state machine updates the gmPriorityVector for the PTP Instance and the masterPriorityVector for each PTP Port of the PTP Instance. The state machine determines the PTP Port state for each PTP Port and updates gmPresent.

NOTE—It is possible to use the external port configuration mechanism to misconfigure the network, e.g., to produce a configuration where one or more PTP Instances have more than one slave port. Detecting and correcting misconfiguations is outside the scope of this standard.

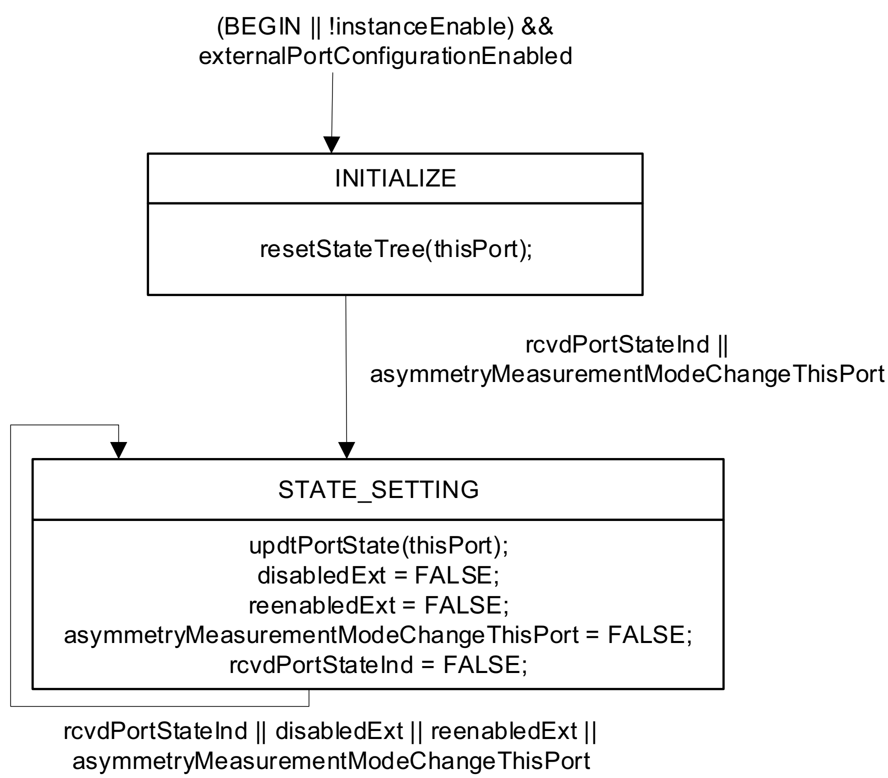


Figure 10-17—PortStateSettingExt state machine

10.3.16 PortAnnounceTransmit state machine

10.3.16.1 State machine variables

The following variables are used in the state diagram in Figure 10-18 (in 10.3.16.3):

10.3.16.1.1 announceSendTime: The time, relative to the LocalClock, at which the next transmission of Announce information is to occur. The data type for announceSendTime is UScaledNs.

10.3.16.1.2 numberAnnounceTransmissions: A count of the number of consecutive Announce message transmissions after the AnnounceIntervalSetting state machine (see Figure 10-19 in 10.3.17.3) has set announceSlowdown (see 10.3.10.2) to TRUE. The data type for numberAnnounceTransmissions is UInteger8.

10.3.16.1.3 interval2: A local variable that holds either announceInterval or oldAnnounceInterval. The data type for interval2 is UScaledNs.

10.3.16.2 State machine functions

10.3.16.2.1 txAnnounce (): Transmits Announce information to the MD entity of this PTP Port. The Announce information is set as follows:

a) The components of the messagePriorityVector are set to the values of the respective components of the masterPriorityVector of this PTP Port.
b) The grandmasterIdentity, grandmasterClockQuality, grandmasterPriority1, and grandmasterPriority2 fields of the Announce message are set equal to the corresponding components of the messagePriorityVector.
c) The value of the stepsRemoved field of the Announce message is set equal to masterStepsRemoved.
d) The Announce message flags leap61, leap59, currentUtcOffsetValid, ptpTimescale, timeTraceable, and frequencyTraceable, and the Announce message fields currentUtcOffset and timeSource, are set equal to the values of the global variables leap61, leap59, currentUtcOffsetValid, ptpTimescale, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource, respectively (see 10.3.9.4 through 10.3.9.11).
e) The sequenceId field of the Announce message is set in accordance with 10.5.7.
f) A path trace TLV (see 10.6.3.3) is constructed, with its pathSequence field (see 10.6.3.3.4) set equal to the pathTrace array (see 10.3.9.23). If appending the path trace TLV to the Announce message does not cause the media-dependent layer frame to exceed any respective maximum size, the path trace TLV is appended to the Announce message; otherwise, it is not appended. If the pathTrace array is empty, the path trace TLV is not appended. See 10.3.9.23 for a description of the path trace feature.

10.3.16.3 State diagram

The PortAnnounceTransmit state machine shall implement the function specified by the state diagram in Figure 10-18, the local variables specified in 10.3.16.1, the functions specified in 10.3.16.2, and the relevant global variables specified in 10.2.4, 10.2.5, 10.3.9, 10.3.10, and 11.2.13. The state machine transmits Announce information to the MD entity when an announce interval has elapsed, PTP Port states have been updated, and portPriority and portStepsRemoved information has been updated with newly determined masterPriority and masterStepsRemoved information.

NOTE—When the external port configuration option is used (i.e., externalPortConfigurationEnabled is TRUE; see 10.3.9.24) the values of the variables updtInfo and selected do not affect the operation of the PortAnnounceTransmit state machine because the term of the conditions in which they appear, i.e., (selected && !updtInfo) || externalPortConfiguartionEnabled, evaluates to TRUE when externalPortConfigurationEnabled is TRUE.

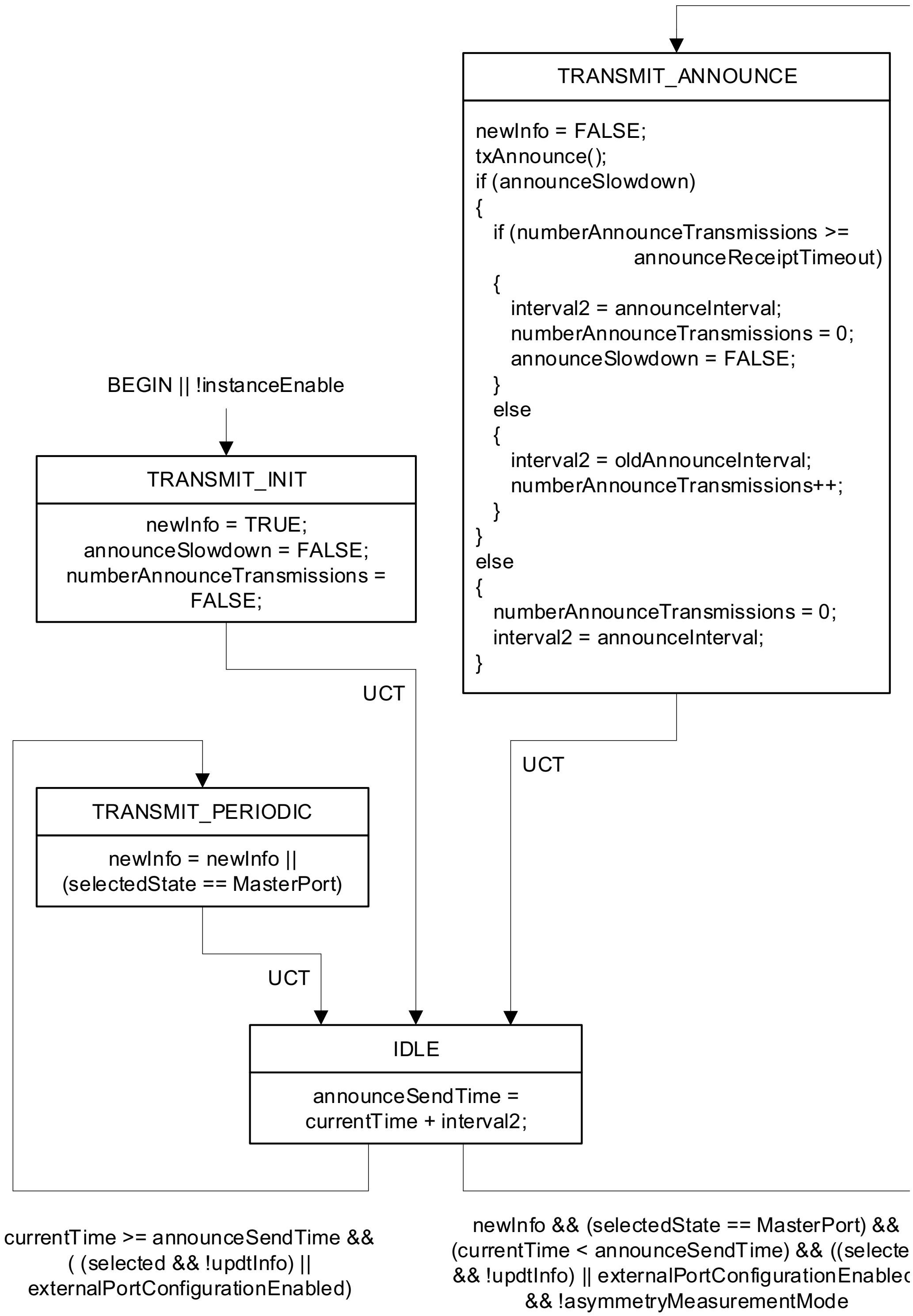


Figure 10-18—PortAnnounceTransmit state machine

10.3.17 AnnounceIntervalSetting state machine

10.3.17.1 State machine variables

The following variables are used in the state diagram in Figure 10-19 (in 10.3.17.3):

10.3.17.1.1 rcvdSignalingMsg2: A Boolean variable that notifies the current state machine when a Signaling message that contains a Message Interval Request TLV (see 10.6.4.3) is received. This variable is reset by the current state machine.

10.3.17.1.2 rcvdSignalingPtrAIS: A pointer to a structure whose members contain the values of the fields of the received Signaling message that contains a Message Interval Request TLV (see 10.6.4.3).

10.3.17.1.3 logSupportedAnnounceIntervalMax: The maximum supported logarithm to base 2 of the announce interval. The data type for logSupportedAnnounceIntervalMax is Integer8.

10.3.17.1.4 logSupportedClosestLongerAnnounceInterval: The logarithm to base 2 of the announce interval, such that logSupportedClosestLongerAnnounceInterval > logRequestedAnnounceInterval, that is numerically closest to logRequestedAnnounceInterval, where logRequestedAnnounceInterval is the argument of the function computeLogAnnounceInterval() (see 10.3.17.2.2). The data type for logSupportedClosestLongerAnnounceInterval is Integer8.

10.3.17.1.5 computedLogAnnounceInterval: A variable used to hold the result of the function computeLogAnnounceInterval(). The data type for computedLogAnnounceInterval is Integer8.

10.3.17.2 State machine functions

10.3.17.2.1 isSupportedLogAnnounceInterval (logAnnounceInterval): A Boolean function that returns TRUE if the announce interval given by the argument logAnnounceInterval is supported by the PTP Port and FALSE if the announce interval is not supported by the PTP Port. The argument logAnnounceInterval has the same data type and format as the field logAnnounceInterval of the message interval request TLV (see 10.6.4.3.8).

10.3.17.2.2 computeLogAnnounceInterval (logRequestedAnnounceInterval): An Integer8 function that computes and returns the logAnnounceInterval, based on the logRequestedAnnounceInterval. This function is defined as indicated below. It is defined here so that the detailed code that it invokes does not need to be placed into the state machine diagram.

```
Integer8 computeLogAnnounceInterval (logRequestedAnnounceInterval)
Integer8 logRequestedAnnounceInterval;
{
        Integer8 logSupportedAnnounceIntervalMax,
                        logSupportedClosestLongerAnnounceInterval;
        if (isSupportedLogAnnounceInterval (logRequestedAnnounceInterval))
                // The requested Announce Interval is supported and returned
                return (logRequestedAnnounceInterval)
        else
        {
                if (logRequestedAnnounceInterval > logSupportedAnnounceIntervalMax)
                        // Return the fasted supported rate, even if faster than the requested rate
                        return (logSupportedAnnounceIntervalMax);
                else
                        // Return the fastest supported rate that is still slower than
                        // the requested rate.
                        return (logSupportedClosestLongerAnnounceInterval);
        }
}
```

10.3.17.3 State diagram

The AnnounceIntervalSetting state machine shall implement the function specified by the state diagram in Figure 10-19, the local variables specified in 10.3.17.1, the functions specified in 10.3.17.2, the messages specified in 10.6, the relevant global variables specified in 10.2.5 and 10.3.10, the relevant managed objects specified in 14.8, and the relevant timing attributes specified in 10.7. This state machine is responsible for setting the global variables that give the duration of the mean interval between successive Announce messages, both at initialization and in response to the receipt of a Signaling message that contains a Message Interval Request TLV (see 10.6.4.3).

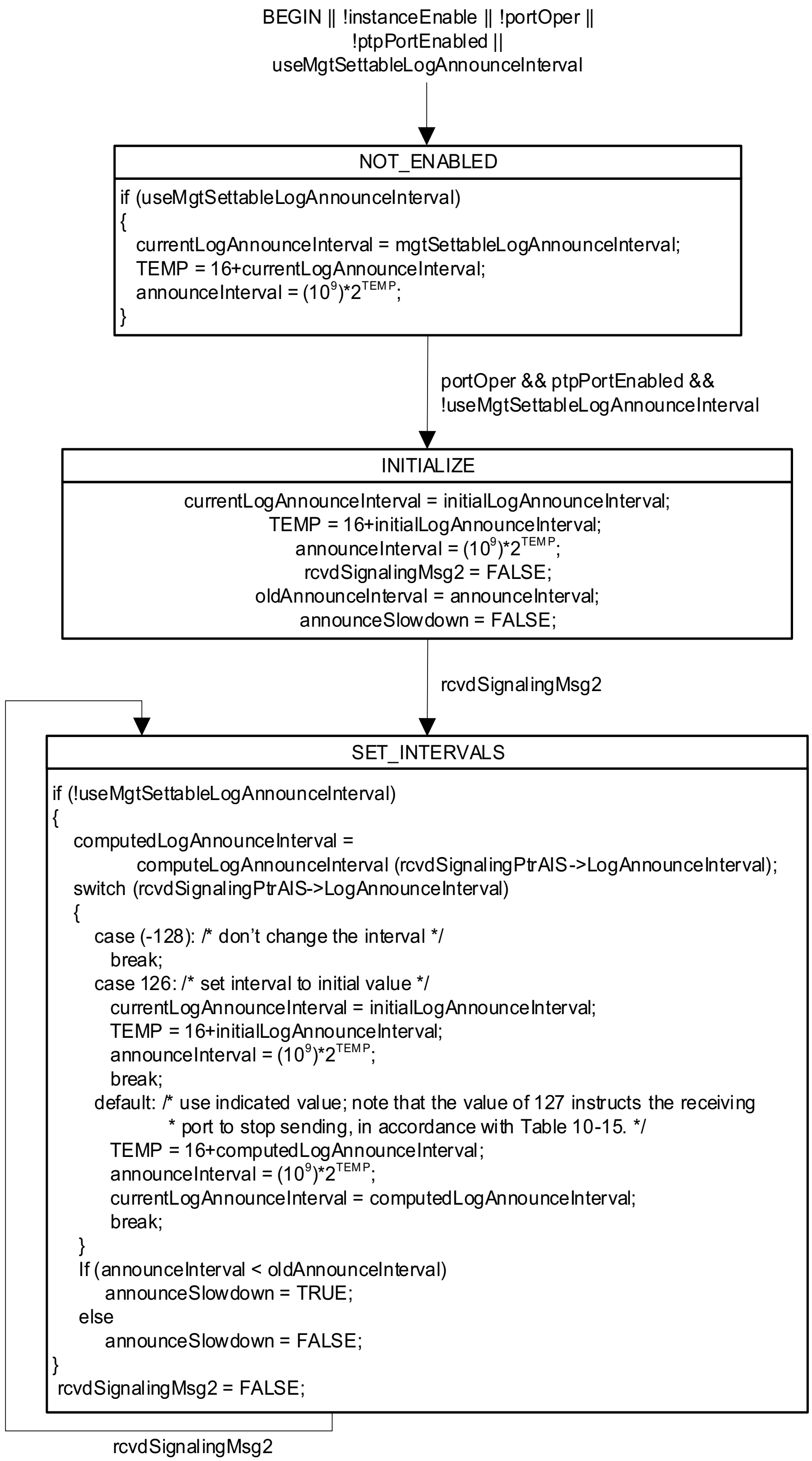


Figure 10-19—AnnounceIntervalSetting state machine

10.3.18 SyncIntervalSetting state machine

10.3.18.1 State machine variables

The following variables are used in the state diagram in Figure 10-20 (in 10.3.18.3):

10.3.18.1.1 rcvdSignalingMsg3: A Boolean variable that notifies the current state machine when a Signaling message that contains a Message Interval Request TLV (see 10.6.4.3) is received. This variable is reset by the current state machine.

10.3.18.1.2 rcvdSignalingPtrSIS: A pointer to a structure whose members contain the values of the fields of the received Signaling message that contains a Message Interval Request TLV (see 10.6.4.3).

10.3.18.1.3 logSupportedSyncIntervalMax: The maximum supported logarithm to base 2 of the sync interval. The data type for logSupportedSyncIntervalMax is Integer8.

10.3.18.1.4 logSupportedClosestLongerSyncInterval: The logarithm to base 2 of the sync interval, such that logSupportedClosestLongerSyncInterval > logRequestedSyncInterval, that is numerically closest to logRequestedSyncInterval, where logRequestedSyncInterval is the argument of the function computeLogSyncInterval() (see 10.3.18.2.2). The data type for logSupportedClosestLongerSyncInterval is Integer8.

10.3.18.1.5 computedLogSyncInterval: A variable used to hold the result of the function computeLogSyncInterval(). The data type for computedLogSyncInterval is Integer8.

10.3.18.2 State machine functions

10.3.18.2.1 isSupportedLogSyncInterval (logSyncInterval): A Boolean function that returns TRUE if the sync interval given by the argument logSyncInterval is supported by the PTP Port and FALSE if the sync interval is not supported by the PTP Port. The argument logSyncInterval has the same data type and format as the field syncInterval of the message interval request TLV (see 10.6.4.3.7).

10.3.18.2.2 computeLogSyncInterval (logRequestedSyncInterval): An Integer8 function that computes and returns the logSyncInterval, based on the logRequestedSyncInterval. This function is defined as indicated below. It is defined here so that the detailed code that it invokes does not need to be placed into the state machine diagram.

```
Integer8 computeLogSyncInterval (logRequestedSyncInterval)
Integer8 logRequestedSyncInterval;
{
        Integer8 logSupportedSyncIntervalMax, logSupportedClosestLongerSyncInterval;
        if (isSupportedLogSyncInterval (logRequestedSyncInterval))
                // The requested Sync Interval is supported and returned
                return (logRequestedSyncInterval)
        else
        {
                if (logRequestedSyncInterval > logSupportedSyncIntervalMax)
                        // Return the fasted supported rate, even if faster than the requested rate
                        return (logSupportedSyncIntervalMax);
                else
                        // Return the fastest supported rate that is still slower than
                        // the requested rate.
                        return (logSupportedClosestLongerSyncInterval);
        }
}
```

10.3.18.3 State diagram

The SyncIntervalSetting state machine shall implement the function specified by the state diagram in Figure 10-20, the local variables specified in 10.3.18.1, the functions specified in 10.3.18.2, the messages specified in 10.6, the relevant global variables specified in 10.2.5, the relevant managed objects specified in 14.8, and the relevant timing attributes specified in 10.7. This state machine is responsible for setting the global variables that give the duration of the mean intervals between successive Sync messages, both at initialization and in response to the receipt of a Signaling message that contains a Message Interval Request TLV (see 10.6.4.3).

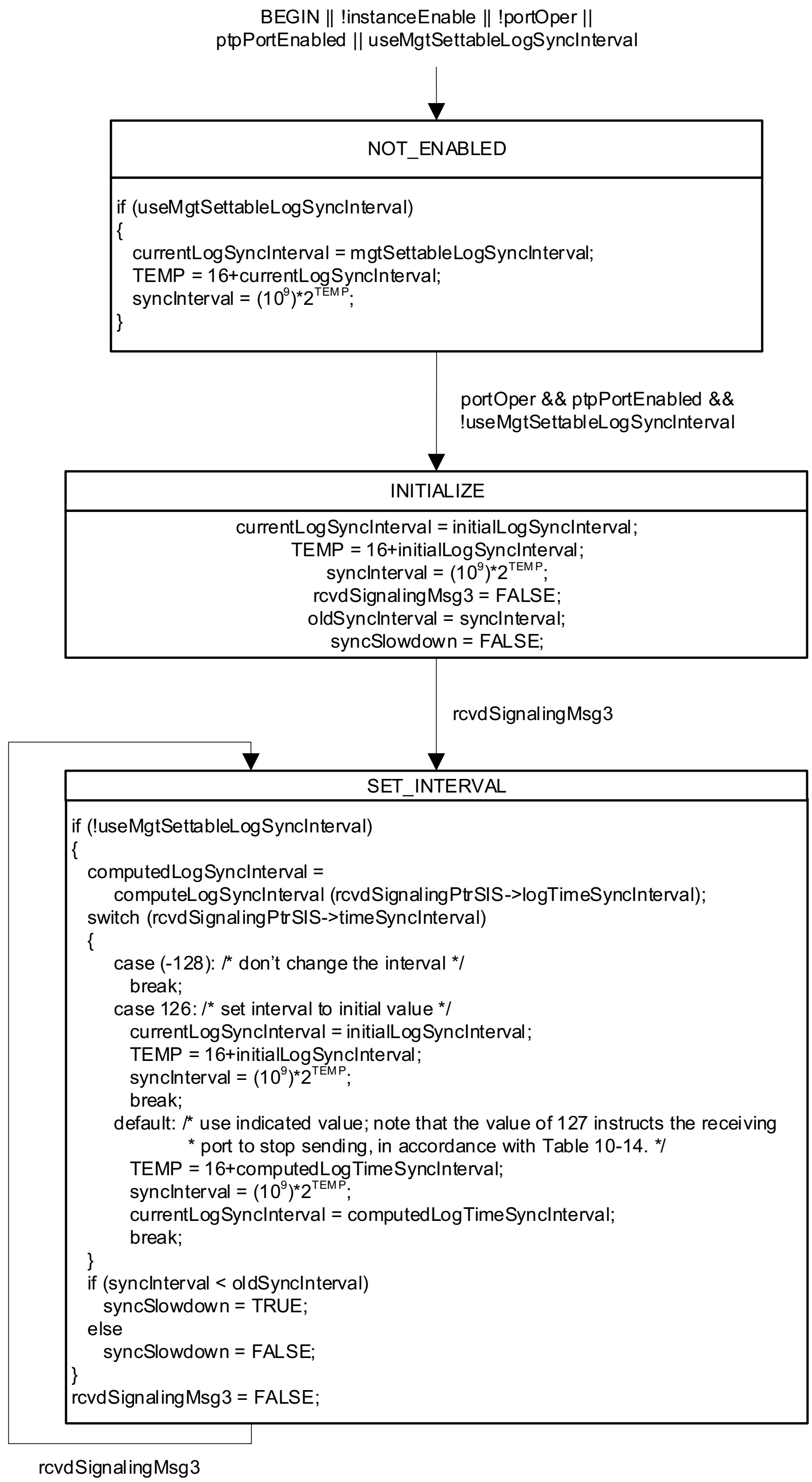


Figure 10-20—SyncIntervalSetting state machine

10.4 State machines related to signaling gPTP capability

10.4.1 GptpCapableTransmit state machine

10.4.1.1 State machine variables

The following variables are used in the state diagram in Figure 10-21 (in 10.4.1.3):

10.4.1.1.1 intervalTimer: A variable used to save the time at which the gPTP-capable message interval timer is set (see Figure 10-21). A Signaling message containing a gPTP-capable TLV is sent when this timer expires. The data type for intervalTimer is UScaledNs.

10.4.1.1.2 txSignalingMsgPtr: A pointer to a structure whose members contain the values of the fields of a Signaling message to be transmitted, which contains a gPTP-capable TLV (see 10.6.4.4).

10.4.1.1.3 interval3: A local variable that holds either gPtpCapableMessageInterval or oldGptpCapableMessageInterval. The data type for interval3 is UScaledNs.

10.4.1.1.4 numberGptpCapableMessageTransmissions: A count of the number of consecutive transmissions of Signaling messages that contain a gPTP-capable TLV, after the GptpCapableIntervalSetting state machine (see Figure 10-23 in 10.4.3.3) has set gPtpCapableMessageSlowdown (see 10.2.5.19) to TRUE. The data type for numberGptpCapableMessageTransmissions is UInteger8.

10.4.1.2 State machine functions

10.4.1.2.1 setGptpCapableTlv(): Creates a structure containing the parameters of a Signaling message that contains a gPTP-capable TLV, to be transmitted (see 10.6.4), and returns a pointer to this structure. The parameters are set as follows:

a) logGptpCapableMessageInterval is set to the value of the managed object currentLogGptpCapableMessageInterval (see 14.8.28).
b) The remaining parameters are set as specified in 10.6.4 and 10.6.4.4.

10.4.1.3 State diagram

The GptpCapableTransmit state machine shall implement the function specified by the state diagram in Figure 10-21, the local variables specified in 10.4.1.1, the functions specified in 10.4.1.2, the relevant parameters specified in 10.6.4 and 10.6.4.4, and the relevant timing attributes specified in 10.7. This state machine is responsible for setting the parameters of each Signaling message that contains the gPTP-capable TLV, and causing these Signaling messages to be transmitted at a regular rate.

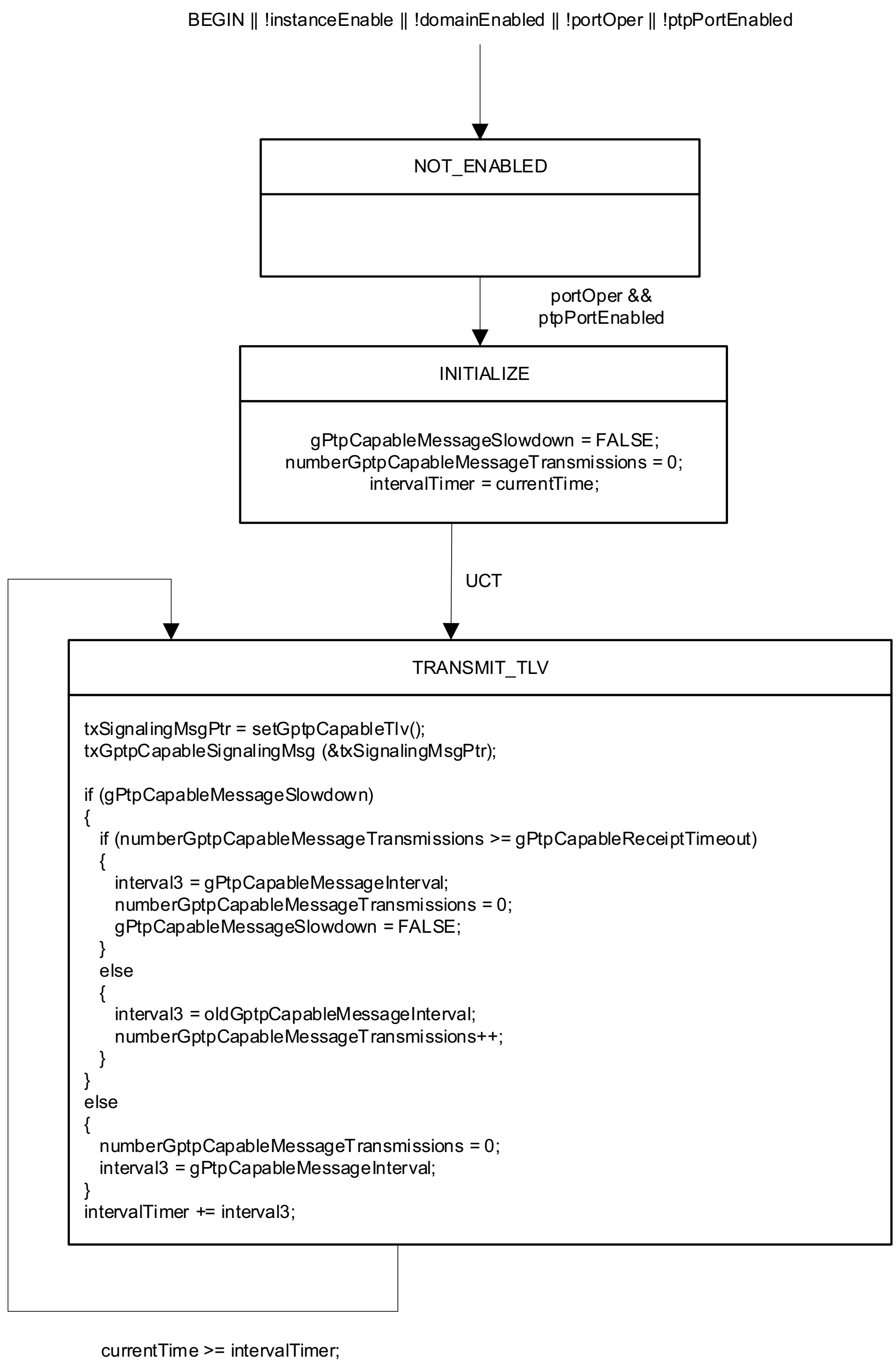


Figure 10-21—GptpCapableTransmit state machine

10.4.2 GptpCapableReceive state machine

10.4.2.1 State machine variables

The following variables are used in the state diagram in Figure 10-22 (in 10.4.2.2):

10.4.2.1.1 rcvdGptpCapableTlv: A Boolean variable that notifies the current state machine when a Signaling message containing a gPTP-capable TLV is received. This variable is reset by the current state machine.

10.4.2.1.2 rcvdSignalingMsgPtr: A pointer to a structure whose members contain the values of the fields of a Signaling message whose receipt is indicated by rcvdGptpCapableTlv (see 10.4.2.1.1).

10.4.2.1.3 gPtpCapableReceiptTimeoutTimeInterval: The time interval after which, if a Signaling message containing a gPTP-capable TLV is not received, the neighbor of this PTP Port is considered to no longer be invoking gPTP. The data type for gPtpCapableReceiptTimeoutTimeInterval is UScaledNs.

10.4.2.1.4 timeoutTime: A variable used to save the time at which the neighbor of this PTP Port is considered to no longer be invoking gPTP if a Signaling message containing a gPTP-capable TLV is not received. The data type for timeoutTime is UScaledNs.

10.4.2.2 State diagram

The GptpCapableReceive state machine shall implement the function specified by the state diagram in Figure 10-22, the local variables specified in 10.4.2.1, the relevant parameters specified in 10.6.4 and 10.6.4.4, and the relevant timing attributes specified in 10.7. This state machine is responsible for setting neighborGptpCapable to TRUE on receipt of a Signaling message containing the gPTP-capable TLV, and setting the timeout time after which neighborGptpCapable is set to FALSE.

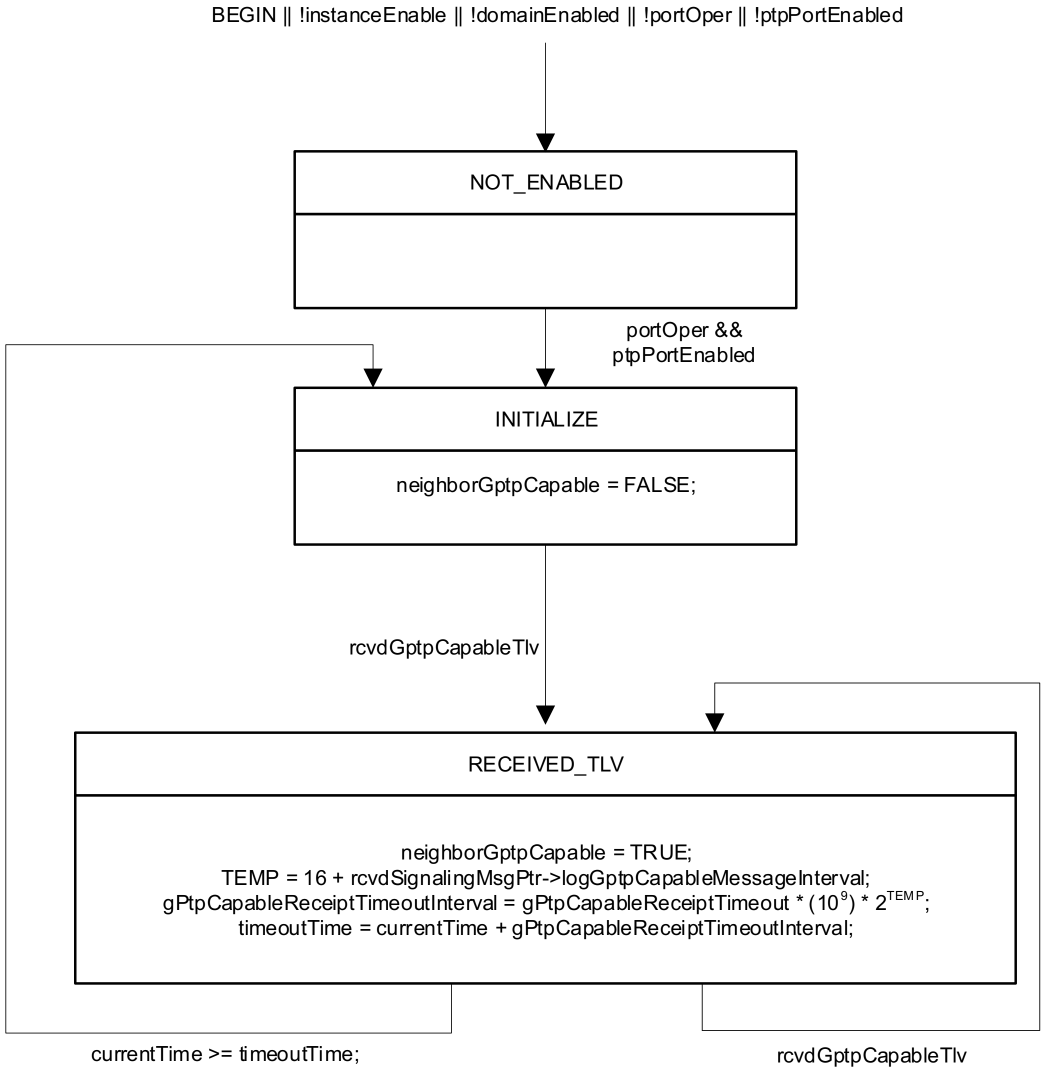


Figure 10-22—GptpCapableReceive state machine

10.4.3 GptpCapableIntervalSetting state machine

10.4.3.1 State machine variables

The following variables are used in the state diagram in Figure 10-23 (in 10.4.3.3):

10.4.3.1.1 rcvdSignalingMsg4: A Boolean variable that notifies the current state machine when a Signaling message that contains a gPTP-capable Message Interval Request TLV (see 10.6.4.5) is received. This variable is reset by the current state machine.

10.4.3.1.2 rcvdSignalingPtrGIS: A pointer to a structure whose members contain the values of the fields of the received Signaling message that contains a gPTP-capable Message Interval Request TLV (see 10.6.4.5).

10.4.3.1.3 logSupportedGptpCapableMessageIntervalMax: The maximum supported logarithm to base 2 of the gPTP-capable message interval. The data type for logSupportedGptpCapableMessageIntervalMax is Integer8.

10.4.3.1.4 logSupportedClosestLongerGptpCapableMessageInterval: The logarithm to base 2 of the gPTP-capable message interval, such that logSupportedClosestLongerGptpCapableMessageInterval > logRequestedGptpCapableMessageInterval, that is numerically closest to logRequestedGptpCapableMessageInterval, where logRequestedGptpCapableMessageInterval is the argument of the function computeLogGptpCapableMessageInterval() (see 10.4.3.2.2). The data type for logSupportedClosestLongerGptpCapableMessageInterval is Integer8.

10.4.3.1.5 computedLogGptpCapableMessageInterval: A variable used to hold the result of the function computeLogGptpCapableMessageInterval(). The data type for computedLogGptpCapableMessageInterval is Integer8.

10.4.3.2 State machine functions

10.4.3.2.1 isSupportedLogGptpCapableMessageInterval (logGptpCapableMessageInterval): A Boolean function that returns TRUE if the gPTP-capable message interval given by the argument logGptpCapableMessageInterval is supported by the PTP Port and FALSE if the gPTP-capable message interval is not supported by the PTP Port. The argument logGptpCapableMessageInterval has the same data type and format as the field logGptpCapableMessageInterval of the gPTP-capable message interval request TLV (see 10.6.4.5.6).

10.4.3.2.2 computeLogGptpCapableMessageInterval (logRequestedGptpCapableMessageInterval): An Integer8 function that computes and returns the logGptpCapableMessageInterval, based on the logRequestedGptpCapableMessageInterval. This function is defined as indicated below. It is defined here so that the detailed code that it invokes does not need to be placed into the state machine diagram.

```
Integer8 computeLogGptpCapableMessageInterval (logRequestedGptpCapableMessageInterval)
Integer8 logRequestedGptpCapableMessageInterval;
{
	Integer8 logSupportedGptpCapableMessageIntervalMax,
			logSupportedClosestLongerGptpCapableMessageInterval;
	if (isSupportedLogGptpCapableMessageInterval
					(logRequestedGptpCapableMessageInterval))
		// The requested gPTP-capable Message Interval is supported and returned
		return (logRequestedGptpCapableMessageInterval)
	else
	{
		if (logRequestedGptpCapableMessageInterval >
					logSupportedGptpCapableMessageIntervalMax)
			// Return the fasted supported rate, even if faster than the requested rate
			return (logSupportedGptpCapableMessageIntervalMax);
		else
			// Return the fastest supported rate that is still slower than
			// the requested rate.
			return (logSupportedClosestLongerGptpCapableMessageInterval);
	}
}
```

10.4.3.3 State diagram

The GptpCapableIntervalSetting state machine shall implement the function specified by the state diagram in Figure 10-23, the local variables specified in 10.4.3.1, the functions specified in 10.4.3.2, the messages specified in 10.6, the relevant global variables specified in 10.2.5, the relevant managed objects specified in 14.8, and the relevant timing attributes specified in 10.7. This state machine is responsible for setting the global variables that give the duration of the mean intervals between successive Signaling messages containing the gPTP-capable TLV, both at initialization and in response to the receipt of a Signaling message that contains a gPTP-capable Message Interval Request TLV (see 10.6.4.5).

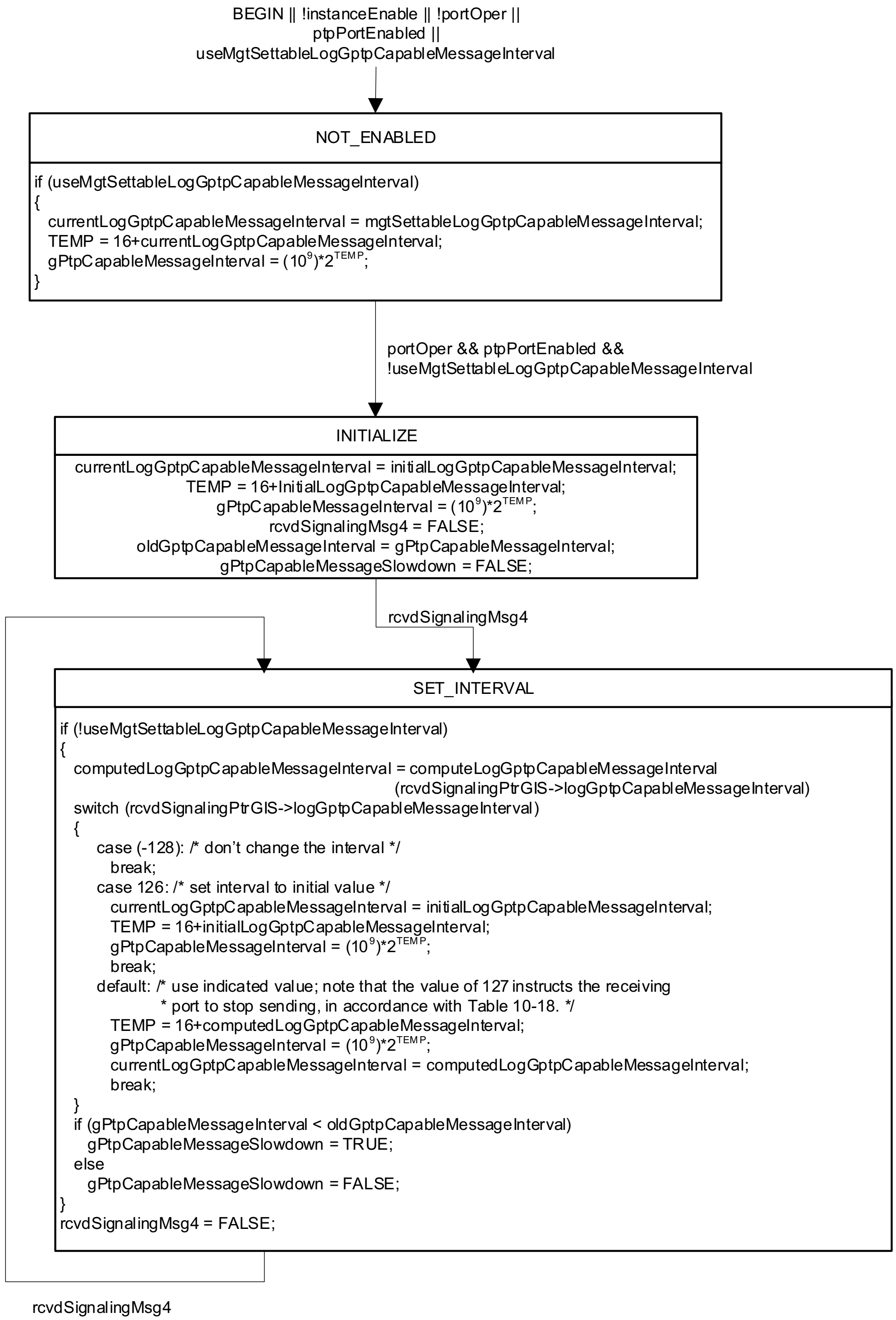


Figure 10-23—GptpCapableIntervalSetting state machine

10.5 Message attributes

10.5.1 General

This subclause describes media-independent attributes of the Announce message and the Signaling message that are not described in 8.4.2 and whose descriptions are not generic to all messages used in this standard. This subclause also describes media-independent attributes of all time-synchronization event messages.

10.5.2 Message class

The Announce message is a general message, i.e., it is not timestamped. An Announce message provides status and characterization information of the PTP Instance that transmitted the message and the Grandmaster PTP Instance. This information is used by the receiving PTP Instance when executing the BMCA.

The Signaling message is a general message, i.e., it is not timestamped. A Signaling message carries information, requests, and/or commands between PTP Instances, via one or more TLVs.

NOTE—In this standard, the Signaling message is used by a port of a PTP Instance to request that the port at the other end of the link send time-synchronization event messages, link delay measurement messages, or Announce messages at desired intervals; to indicate whether the port at the other end of the link should compute neighborRateRatio and/or meanLinkDelay; and to indicate whether a PTP Port can receive and correctly process one-step Syncs. The message interval request TLV is defined to carry this information (see 10.6.4.3). One usage of this functionality is to allow a time-aware system in power-saving mode to remain connected to a gPTP domain via the port on which the Signaling message is sent.

10.5.3 Addresses

The destination address of the Announce and Signaling messages shall be the reserved multicast address given in Table 10-4 unless otherwise specified in a media-dependent clause (see 12.2 and 16.2).

Table 10-4—Destination address for Announce and Signaling messages

Destination address
01-80-C2-00-00-0E
NOTE—This address is taken from Table 8-1, Table 8-2, and Table 8-3 of IEEE Std 802.1Q-2018.

NOTE—Frames whose destination address is the address of Table 10-4 are never forwarded, according to IEEE 802.1Q protocol. Use of this address is shared by IEEE 802.1AS and other IEEE 802.1 protocols.

10.5.4 EtherType

The EtherType of the Announce and Signaling messages shall be the EtherType given in Table 10-5.

Table 10-5—EtherType for Announce and Signaling messages

EtherType
88-F7

NOTE—This EtherType is used for all PTP messages.

10.5.5 Subtype

The subtype of the Announce and Signaling messages is indicated by the majorSdoId field (see 10.6.2.2.1).

NOTE—The subtype for all PTP messages is indicated by the majorSdoId field.

10.5.6 Source port identity

The Announce message, Signaling message, and all time-synchronization messages contain a sourcePortIdentity field (see 10.6.2.2.11), which identifies the egress port (see 8.5) on which the respective message is sent.

10.5.7 Sequence number

Each PortSync entity of a PTP Instance maintains a separate sequenceId pool for each of the message types Announce and Signaling, respectively, transmitted by the MD entity of the PTP Port.

Each Announce and Signaling message contains a sequenceId field (see 10.6.2.2.12), which carries the message sequence number. The sequenceId of an Announce message shall be one greater than the sequenceId of the previous Announce message sent by the transmitting PTP Port, subject to the constraints of the rollover of the UInteger16 data type used for the sequenceId field. The sequenceId of a Signaling message shall be one greater than the sequenceId of the previous Signaling message sent by the transmitting PTP Port, subject to the constraints of the rollover of the UInteger16 data type used for the sequenceId field.

10.6 Message formats

10.6.1 General

The PTP messages Announce and Signaling each have a header, body, and, if present, a suffix that contains one or more TLVs (see 10.6.2, 10.6.3, and 10.6.4 of this standard and Clause 14 of IEEE Std 1588-2019). Reserved fields shall be transmitted with all bits of the field 0 and ignored by the receiver, unless otherwise specified. The data type of the field shall be the type indicated in brackets in the title of each subclause.

Subclause 10.6 defines the path trace TLV, which is carried by the Announce message (see 10.6.3.2.8), and the message interval request TLV, which is carried by the Signaling message (see 10.6.4.3).

PTP Management Messages are not used in this standard. They are specified in IEEE Std 1588-2019.

IEEE Std 1588-2019 specifies various optional features that have associated TLVs. These optional features, including the associated TLVs, may be supported by an implementation of this standard. IEEE Std 1588-2019 also specifies that certain TLVs are propagated by a Boundary Clock if they are attached to an Announce message and are not supported (see 14.2.2.2 and Table 52 of IEEE Std 1588-2019). These TLVs are listed in Table 10-6. The TLV Propagate requirement in IEEE Std 1588-2019 means that a Propagate TLV is propagated through a PTP Relay Instance (e.g., from an ingress PTP Port in the Slave state to an egress PTP Port in the Master state, even when the TLV is unsupported by the PTP Relay Instance). If the corresponding optional feature is not supported by the PTP Relay Instance, the PTP Relay Instance shall propagate the TLV unchanged.

If a PTP Instance cannot parse a non-forwarding TLV, it shall ignore it and attempt to parse the next TLV (see 14.1 of IEEE Std 1588-2019).

NOTE—Any overhead specific to the respective medium is added to each message.

Table 10-6—Propagate TLVs of IEEE Std 1588-2019

tlvType values	Value (hex)	TLV defined in
PATH_TRACE	0008	16.2 of IEEE Std 1588-2019, and required by 10.6.3.3 of this standard
ALTERNATE_TIME_OFFSET_INDICATOR	0009	16.3 of IEEE Std 1588-2019
ORGANIZATION_EXTENSION_PROPAGATE	4000	14.3 of IEEE Std 1588-2019
ENHANCED_ACCURACY_METRICS	4001	16.12 of IEEE Std 1588-2019
Reserved for assignment by the IEEE 1588 Working Group for TLVs that propagate	4002–7EFF	
Experimental values (see 4.2.9 of IEEE Std 1588-2019)	7FF0–7FFF	

10.6.2 Header

10.6.2.1 General header specifications

The common header for all PTP messages shall be as specified in Table 10-7 and 10.6.2.2.

Table 10-7—PTP message header

Bits 7–4	Bits 3–0	Octets	Offset
majorSdoId	messageType	1	0
minorVersionPTP	versionPTP	1	1
messageLength		2	2
domainNumber		1	4
minorSdoId		1	5
flags		2	6
correctionField		8	8
messageTypeSpecific		4	16
sourcePortIdentity		10	20
sequenceId		2	30
controlField		1	32
logMessageInterval		1	33

10.6.2.2 Header field specifications

10.6.2.2.1 majorSdoId (Nibble)

The value is specified in 8.1 for all transmitted PTP messages of a gPTP domain. The value is specified in 11.2.17 for all transmitted PTP messages of the Common Mean Link Delay Service. Any PTP message received for which the value is not one of the values specified in those subclauses shall be ignored.

10.6.2.2.2 messageType (Enumeration4)

The value indicates the type of the message, as defined in Table 10-8.

The most significant bit of the messageType field divides this field in half between event and general messages, i.e., it is 0 for event messages and 1 for general messages.

Table 10-8—Values for messageType field

Message type	Message class	Value
Announce	General	0xB
Signaling	General	0xC
NOTE—Values for the messageType field for other PTP messages that are used only for specific media are defined in the respective media-dependent clause(s).		

10.6.2.2.3 minorVersionPTP (UInteger4)

For transmitted messages, the value shall be 1 (see 7.5.4 and 13.3.2.5 of IEEE Std 1588-2019). For received messages, the value is ignored.

NOTE—minorVersionPTP indicates the minor version number of IEEE 1588 PTP used in the PTP profile contained in this standard for information only.

10.6.2.2.4 versionPTP (UInteger4)

For transmitted messages, the value shall be 2 (see 7.5.4 and 13.3.2.4 of IEEE Std 1588-2019). For received messages, if the value is not 2, the entire message shall be ignored.

NOTE—versionPTP indicates the version number of IEEE 1588 PTP used in the PTP profile contained in this standard.

10.6.2.2.5 messageLength (UInteger16)

The value is the total number of octets that form the PTP message. The counted octets start with, and include, the first octet of the header and terminate with, and include, the last octet of the last TLV or, if there are no TLVs, the last octet of the message as defined in this clause.

NOTE—For example, the Follow_Up message (see 11.4.4) contains a PTP header (34 octets), preciseOriginTimestamp (10 octets), and Follow_Up information TLV (32 octets). The value of the messageLength field is 34+10+32 = 76.

10.6.2.2.6 domainNumber (UInteger8)

The value is the gPTP domain number specified in 8.1.

10.6.2.2.7 minorSdoId (UInteger8)

The value is specified in 8.1 for all transmitted PTP messages of a gPTP domain. The value is specified in 11.2.17 for all transmitted PTP messages of the Common Mean Link Delay Service. Any PTP message received for which the value is not one of the values specified in those subclauses shall be ignored.

10.6.2.2.8 flags (Octet2)

The value of the bits of the array are defined in Table 10-9. For message types where the bit is not defined in Table 10-9, the value of the bit is set to FALSE.

Table 10-9—Values of flag bits

Octet	Bit	Message types	Name	Value
0	0	All	alternateMasterFlag in Announce, Sync, Follow_Up, and Delay_Resp messages	Not used in this standard; transmitted as FALSE and ignored on reception
0	1	Sync, Pdelay_Resp	twoStepFlag	*For Sync messages:* a) For a one-step transmitting PTP Port (see 11.1.3 and 11.2.13.9), the value is FALSE. b) For a two-step transmitting PTP Port, the value is TRUE. *For Pdelay_Resp messages:* The value is transmitted as TRUE and ignored on reception.
0	2	All	unicastFlag	Not used in this standard; transmitted as FALSE and ignored on reception
0	3	All	Reserved	Not used by IEEE Std 1588-2019; reserved as FALSE and ignored on reception
0	4	All	Reserved	Not used by IEEE Std 1588-2019; reserved as FALSE and ignored on reception
0	5	All	PTP profileSpecific 1	Not used in this standard; transmitted as FALSE and ignored on reception
0	6	All	PTP profileSpecific 2	Not used in this standard; transmitted as FALSE and ignored on reception
0	7	All	Reserved	Not used in this standard; transmitted as FALSE and ignored on reception
1	0	Announce	leap61	The value of the global variable leap61 (see 10.3.9.4)
1	1	Announce	leap59	The value of the global variable leap59 (see 10.3.9.5)
1	2	Announce	currentUtcOffsetValid	The value of the global variable currentUtcOffsetValid (see 10.3.9.6)
1	3	Announce	ptpTimescale	The value of the global variable ptpTimescale (see 10.3.9.7)

Table 10-9—Values of flag bits *(continued)*

Octet	Bit	Message types	Name	Value
1	4	Announce	timeTraceable	The value of the global variable timeTraceable (see 10.3.9.8)
1	5	Announce	frequencyTraceable	The value of the global variable frequencyTraceable (see 10.3.9.9)
1	6	All	Reserved	Not used by IEEE Std 1588-2019; reserved as FALSE and ignored on reception
1	7	All	Reserved	Not used in this standard; reserved as FALSE and ignored on reception

10.6.2.2.9 correctionField (Integer64)

The value is 0.

10.6.2.2.10 messageTypeSpecific (Octet4)

The value of the messageTypeSpecific field varies, based on the value of the messageType field, as described in Table 10-10.

For Event messages only, the four octets of the messageTypeSpecific field may be used for internal implementation of a PTP Instance and its ports. For example, if the clock consists of multiple hardware components that are not synchronized, messageTypeSpecific can be used to transfer an internal timestamp between components (e.g., a physical layer chip and the clock's processor).

The messageTypeSpecific field is not used for features of this standard, and it has no meaning from one clock to another. In the on-the-wire format at each PTP Port, for all messageType values, the messageTypeSpecific field is transmitted with all bits of the field 0 and ignored on receive.

Table 10-10—messageTypeSpecific semantics

Value of messageType	Description
Follow_Up, Pdelay_Resp_Follow_Up, Announce, Signaling, Management	For the General message class, this field is reserved; it is transmitted as 0 and ignored on reception.
Sync, Delay_Req, Pdelay_Resp	For the Event message class, this field may be used for internal implementation as specified in this subclause.

10.6.2.2.11 sourcePortIdentity (PortIdentity)

The value is the PTP Port identity attribute (see 8.5.2) of the PTP Port that transmits the PTP message.

10.6.2.2.12 sequenceId (UInteger16)

The sequenceId field is assigned as specified in 10.5.7.

10.6.2.2.13 controlField (UInteger8)

The value is 0.

10.6.2.2.14 logMessageInterval (Integer8)

For an Announce message, the value is the value of currentLogAnnounceInterval (see 10.3.10.6) for the PTP Port that transmits the Announce message. For a Signaling message, the value is transmitted as 0x7F and ignored on reception.

10.6.3 Announce message

10.6.3.1 General Announce message specifications

The fields of the body of the Announce message shall be as specified in Table 10-11 and 10.6.3.2.

Table 10-11—Announce message fields

Bits (7 6 5 4 3 2 1 0)	Octets	Offset
header (see 10.6.2)	34	0
reserved	10	34
currentUtcOffset	2	44
reserved	1	46
grandmasterPriority1	1	47
grandmasterClockQuality	4	48
grandmasterPriority2	1	52
grandmasterIdentity	8	53
stepsRemoved	2	61
timeSource	1	63
path trace TLV	4+8N	64

10.6.3.2 Announce message field specifications

10.6.3.2.1 currentUtcOffset (Integer16)

The value is the value of currentUtcOffset (see 10.3.9.10) for the PTP Instance that transmits the Announce message.

10.6.3.2.2 grandmasterPriority1 (UInteger8)

The value is the value of the priority1 component of the rootSystemIdentity of the gmPriorityVector (see 10.3.5) of the PTP Instance that transmits the Announce message.

10.6.3.2.3 grandmasterClockQuality (ClockQuality)

The value is the clockQuality formed by the clockClass, clockAccuracy, and offsetScaledLogVariance of the rootSystemIdentity of the gmPriorityVector (see 10.3.5) of the PTP Instance that transmits the Announce message.

10.6.3.2.4 grandmasterPriority2 (UInteger8)

The value is the value of the priority2 component of the rootSystemIdentity of the gmPriorityVector (see 10.3.5) of the PTP Instance that transmits the Announce message.

10.6.3.2.5 grandmasterIdentity (ClockIdentity)

The value is the value of the clockIdentity component of the rootSystemIdentity of the gmPriorityVector (see 10.3.5) of the PTP Instance that transmits the Announce message.

10.6.3.2.6 stepsRemoved (UInteger16)

The value is the value of masterStepsRemoved (see 10.3.9.3) for the PTP Instance that transmits the Announce message.

10.6.3.2.7 timeSource (TimeSource)

The value is the value of timeSource (see 8.6.2.7 and 10.3.9.11) for the PTP Instance that transmits the Announce message.

10.6.3.2.8 Path trace TLV

The Announce message carries the path trace TLV, defined in 10.6.3.3.

10.6.3.3 Path trace TLV definition

10.6.3.3.1 General

The fields of the path-trace TLV shall be as specified in Table 10-12 and in 10.6.4.3.2 through 10.6.4.3.9. This TLV and its use are defined in IEEE Std 1588-2019 (see 16.2 and Table 52 of IEEE Std 1588-2019).

Table 10-12—Path trace TLV

Bits								Octets	Offset from start of TLV
7	6	5	4	3	2	1	0		
tlvType								2	0
lengthField								2	2
pathSequence								8N	4

10.6.3.3.2 tlvType (Enumeration16)

The value of the tlvType field is 0x8.

NOTE—This value indicates the TLV is a path trace TLV, as specified in 16.2.5.1 and Table 52 of IEEE Std 1588-2019. The value 0x8 is specified in that standard as PATH_TRACE.

10.6.3.3.3 lengthField (UInteger16)

The value of the lengthField is 8N.

10.6.3.3.4 pathSequence (ClockIdentity[N])

The value of pathSequence is a ClockIdentity array. The array elements are the clockIdentities of the successive PTP Instances that receive and send an Announce message. The quantity N is the number of PTP Instances, including the Grandmaster PTP Instance, that the Announce information has traversed.

NOTE—N is equal to stepsRemoved+1 (see 10.6.3.2.6). The size of the pathSequence array increases by 1 for each PTP Instance that the Announce information traverses.

10.6.4 Signaling message

10.6.4.1 General Signaling message specifications

The fields of the body of the Signaling message shall be as specified in Table 10-13 and 10.6.4.2.

Table 10-13—Signaling message fields

Bits: 7 6 5 4 3 2 1 0	Octets	Offset
header (see 10.6.2)	34	0
targetPortIdentity	10	34
message interval request TLV, gPTP-capable TLV, or gPTP-capable message interval request TLV	16	44

10.6.4.2 Signaling message field specifications

10.6.4.2.1 targetPortIdentity (PortIdentity)

The value of targetPortIdentity.clock identity is all ones, i.e., 0xFFFFFFFFFFFFFFFF. The value of targetPortIdentity.portNumber is all ones, i.e., 0xFFFF.

10.6.4.2.2 Message interval request TLV or gPTP-capable TLV

The Signaling message carries either the message interval request TLV, defined in 10.6.4.3, or the gPTP-capable TLV, defined in 10.6.4.4, but not both. If it is desired to send both TLVs, two Signaling messages must be sent.

10.6.4.3 Message interval request TLV definition

10.6.4.3.1 General

The fields of the message interval request TLV shall be as specified in Table 10-14 and in 10.6.4.3.2 through 10.6.4.3.9. This TLV is a standard organization extension TLV for the Signaling message, as specified in 14.3 of IEEE Std 1588-2019.

Table 10-14—Message interval request TLV

Bits (7 6 5 4 3 2 1 0)	Octets	Offset from start of TLV
tlvType	2	0
lengthField	2	2
organizationId	3	4
organizationSubType	3	7
logLinkDelayInterval	1	10
logTimeSyncInterval	1	11
logAnnounceInterval	1	12
flags	1	13
reserved	2	14

10.6.4.3.2 tlvType (Enumeration16)

The value of the tlvType field is 0x3.

NOTE—This value indicates the TLV is a vendor and standard organization extension TLV, as specified in 14.3.2.1 and Table 52 of IEEE Std 1588-2019. The tlvType is specified in that standard as ORGANIZATION_EXTENSION with a value of 0x3.

10.6.4.3.3 lengthField (UInteger16)

The value of the lengthField is 12.

10.6.4.3.4 organizationId (Octet3)

The value of organizationId is 00-80-C2.

10.6.4.3.5 organizationSubType (Enumeration24)

The value of organizationSubType is 2.

10.6.4.3.6 logLinkDelayInterval (Integer8)

The value is the logarithm to base 2 of the mean time interval, desired by the port that sends this TLV, between successive Pdelay_Req messages sent by the port at the other end of the link. The format and allowed values of logLinkDelayInterval are the same as the format and allowed values of initialLogPdelayReqInterval (see 11.5.2.2).

The values 127, 126, and –128 are interpreted as defined in Table 10-15.

Table 10-15—Interpretation of special values of logLinkDelayInterval

Value	Instruction to PTP Instance that receives this TLV
127	Instructs the port that receives this TLV to stop sending link delay measurement messages.
126	Instructs the port that receives this TLV to set currentLogPdelayReqInterval to the value of initialLogPdelayReqInterval (see 11.5.2.2).
–128	Instructs the port that receives this TLV not to change the mean time interval between successive Pdelay_Req messages.
All values in the ranges [–127, –25] and [25, 125] are reserved.	

10.6.4.3.7 logTimeSyncInterval (Integer8)

The value is the logarithm to base 2 of the mean time interval, desired by the PTP Port that sends this TLV, between successive time-synchronization event messages sent by the PTP Port at the other end of the link. The format and allowed values of logTimeSyncInterval are the same as the format and allowed values of initialLogSyncInterval (see 10.7.2.3, 11.5.2.3, 12.8, and 13.9.2).

The values 127, 126, and –128 are interpreted as defined in Table 10-16.

Table 10-16—Interpretation of special values of logTimeSyncInterval

Value	Instruction to PTP Instance that receives this TLV
127	Instructs the PTP Port that receives this TLV to stop sending time-synchronization event messages.
126	Instructs the PTP Port that receives this TLV to set currentLogSyncInterval to the value of initialLogSyncInterval (see 10.7.2.3, 11.5.2.3, 12.8, and 13.9.2).
–128	Instructs the PTP Port that receives this TLV not to change the mean time interval between successive time-synchronization event messages.
All values in the ranges [–127, –25] and [25, 125] are reserved.	

When a Signaling message that contains this TLV is sent by a PTP Port, the value of syncReceiptTimeoutTimeInterval for that PTP Port (see 10.2.5.3) shall be set equal to syncReceiptTimeout (see 10.7.3.1) multiplied by the value of the interval, in seconds, reflected by logTimeSyncInterval.

10.6.4.3.8 logAnnounceInterval (Integer8)

The value is the logarithm to base 2 of the mean time interval, desired by the PTP Port that sends this TLV, between successive Announce messages sent by the PTP Port at the other end of the link. The format and allowed values of logAnnounceInterval are the same as the format and allowed values of initialLogAnnounceInterval (see 10.7.2.2).

The values 127, 126, and –128 are interpreted as defined in Table 10-17.

Table 10-17—Interpretation of special values of logAnnounceInterval

Value	Instruction to PTP Instance that receives this TLV
127	Instructs the PTP Port that receives this TLV to stop sending Announce messages.
126	Instructs the PTP Port that receives this TLV to set currentLogAnnounceInterval to the value of initialLogAnnounceInterval (see 10.7.2.2).
–128	Instructs the PTP Port that receives this TLV not to change the mean time interval between successive Announce messages.
All values in the ranges [–127, –25] and [25, 125] are reserved.	

When a Signaling message that contains this TLV is sent by a PTP Port, the value of announceReceiptTimeoutTimeInterval for that PTP Port (see 10.3.10.1) shall be set equal to announceReceiptTimeout (see 10.7.3.2) multiplied by the value of the interval, in seconds, reflected by logAnnounceInterval.

10.6.4.3.9 flags (Octet)

Bits 0 through 2 of the octet are defined in Table 10-18 and take on the values TRUE and FALSE. Bits not defined in Table 10-18 are set to FALSE and ignored on receipt.

Table 10-18—Definitions of bits of flags field of message interval request TLV

Bit	Name
0	computeNeighborRateRatio
1	computeMeanLinkDelay
2	oneStepReceiveCapable

NOTE—For full-duplex point-to-point links (see Clause 11), it is expected that the PTP Port sending this TLV will set bits 0 and/or 1 to FALSE if this PTP Port will not provide valid timing information in its subsequent responses (Pdelay_Resp and Pdelay_Resp_Follow_Up) to Pdelay_Req messages. Similarly, it is expected that the PTP Port sending this TLV will set bit 2 to TRUE if it is capable of receiving and correctly processing one-step Sync messages.

10.6.4.4 gPTP-capable TLV definition

10.6.4.4.1 General

The fields of the gPTP-capable TLV shall be as specified in Table 10-19 and in 10.6.4.4.2 through 10.6.4.4.7. This TLV is a standard organization extension TLV for the Signaling message, as specified in 14.3 of IEEE Std 1588-2019.

Table 10-19—gPTP-capable TLV

Bits (7 6 5 4 3 2 1 0)	Octets	Offset from start of TLV
tlvType	2	0
lengthField	2	2
organizationId	3	4
organizationSubType	3	7
logGptpCapableMessageInterval	1	10
flags	1	11
reserved	4	12

10.6.4.4.2 tlvType (Enumeration16)

The value of the tlvType field is 0x8000.

NOTE—This value indicates the TLV is a vendor and standard organization extension TLV that is not propagated, as specified in 14.3.2.1 and Table 52 of IEEE Std 1588-2019. The tlvType is specified in that standard as ORGANIZATION_EXTENSION_DO_NOT_PROPAGATE with a value of 0x8000.

10.6.4.4.3 lengthField (UInteger16)

The value of the lengthField is 12.

10.6.4.4.4 organizationId (Octet3)

The value of organizationId is 00-80-C2.

10.6.4.4.5 organizationSubType (Enumeration24)

The value of organizationSubType is 4.

10.6.4.4.6 logGptpCapableMessageInterval (Integer8)

The value of logGptpCapableMessageInterval is the logarithm to base 2 of the mean gPTP-capable message interval in seconds (see 10.7.2.1 and 10.7.2.5)

10.6.4.4.7 flags (Octet)

The flag bits shall be transmitted as FALSE and ignored on receipt.

10.6.4.5 gPTP-capable message interval request TLV definition

10.6.4.5.1 General

The fields of the gPTP-capable message interval request TLV shall be as specified in Table 10-20 and in 10.6.4.5.2 through 10.6.4.5.6. This TLV is a standard organization extension TLV for the Signaling message, as specified in 14.3 of IEEE Std 1588-2019.

Table 10-20—gPTP-capable message interval request TLV

Bits (7 6 5 4 3 2 1 0)	Octets	Offset from start of TLV
tlvType	2	0
lengthField	2	2
organizationId	3	4
organizationSubType	3	7
logGptpCapableMessageInterval	1	10
reserved	3	11

10.6.4.5.2 tlvType (Enumeration16)

The value of the tlvType field is 0x8000.

NOTE—This value indicates the TLV is a vendor and standard organization extension TLV that is not propagated, as specified in 14.3.2.1 and Table 52 of IEEE Std 1588-2019. The tlvType is specified in that standard as ORGANIZATION_EXTENSION_DO_NOT_PROPAGATE with a value of 0x8000.

10.6.4.5.3 lengthField (UInteger16)

The value of the lengthField is 10.

10.6.4.5.4 organizationId (Octet3)

The value of organizationId is 00-80-C2.

10.6.4.5.5 organizationSubType (Enumeration24)

The value of organizationSubType is 5.

10.6.4.5.6 logGptpCapableMessageInterval (Integer8)

The value is the logarithm to base 2 of the mean time interval, desired by the PTP Port that sends this TLV, between successive Signaling messages that contain the gPTP-capable TLV (see 10.6.4.4), sent by the PTP Port at the other end of the link. The format and allowed values of logGptpCapableMessageInterval are the same as the format and allowed values of initialLogGptpCapableMessageInterval (see 10.7.2.5).

The values 127, 126, and –128 are interpreted as defined in Table 10-21.

Table 10-21—Interpretation of special values of logGptpCapableMessageInterval

Value	Instruction to PTP Instance that receives this TLV
127	Instructs the PTP Port that receives this TLV to stop sending Signaling messages that contain the gPTP-capable TLV.
126	Instructs the PTP Port that receives this TLV to set currentlogGptpCapableMessageInterval to the value of initialLogGptpCapableMessageInterval (see 10.7.2.4).
–128	Instructs the PTP Port that receives this TLV not to change the mean time interval between successive Signaling messages that contain the gPTP-capable TLV.
All values in the ranges [–127, –25] and [25, 125] are reserved.	

When a Signaling message that contains this TLV is sent by a PTP Port, the value of gPtpCapableReceiptTimeoutTimeInterval for that PTP Port (see 10.3.10.1) shall be set equal to pPtpCapableReceiptTimeout (see 10.7.3.3) multiplied by the value of the interval, in seconds, reflected by logGptpCapableMessageInterval.

10.7 Protocol timing characterization

10.7.1 General

This subclause specifies timing and timeout attributes for the media-independent sublayer state machines.

10.7.2 Message transmission intervals

10.7.2.1 General interval specification

The mean time interval between the sending of successive Announce messages, known as the *announce interval*, shall be as specified in 10.7.2.2.

The mean time interval between the sending of successive time-synchronization event messages for full-duplex point-to-point, IEEE 802.11, and CSN links, and successive general messages containing time-synchronization information for IEEE 802.3 EPON links, is known as the *sync interval*. The sync interval shall be as specified in 10.7.2.3.

The mean time interval between the sending of successive Signaling messages that contain the gPTP-capable TLV, known as the *gPTP-capable message interval*, shall be as specified in 10.7.2.5.

10.7.2.2 Announce message transmission interval

The logarithm to the base 2 of the announce interval (in seconds) is carried in the logMessageInterval field of the Announce message.

When useMgtSettableLogAnnounceInterval (see 14.8.14) is FALSE, the initialLogAnnounceInterval specifies the announce interval when the PTP Port is initialized and the value to which the announce interval is set when a message interval request TLV is received with the logAnnounceInterval field set to 126 (see the AnnounceIntervalSetting state machine in 10.3.17). The currentLogAnnounceInterval specifies the current value of the announce interval. The default value of initialLogAnnounceInterval is 0. Every PTP Port supports the value 127; the PTP Port does not send Announce messages when currentLogAnnounceInterval has this value (see 10.3.17). A PTP Port may support other values, except for the reserved values indicated in Table 10-17. A PTP Port ignores requests for unsupported values (see 10.3.17). The initialLogAnnounceInterval and currentLogAnnounceInterval are per-PTP Port attributes.

When useMgtSettableLogAnnounceInterval is TRUE, currentLogAnnounceInterval is set equal to mgtSettableLogAnnounceInterval (see 14.8.15), and initialLogAnnounceInterval is ignored.

Announce messages shall be transmitted such that the value of the arithmetic mean of the intervals, in seconds, between message transmissions is within ± 30% of $2^{\text{currentLogAnnounceInterval}}$. In addition, a PTP Port shall transmit Announce messages such that at least 90% of the inter-message intervals are within ± 30% of the value of $2^{\text{currentLogAnnounceInterval}}$. The interval between successive Announce messages should not exceed twice the value of $2^{\text{portDS.logAnnounceInterval}}$ in order to prevent causing an announceReceiptTimeout event. The PortAnnounceTransmit state machine (see 10.3.16) is consistent with these requirements, i.e., the requirements here and the requirements of the PortAnnounceTransmit state machine can be met simultaneously.

NOTE 1—A minimum number of inter-message intervals is necessary in order to verify that a PTP Port meets these requirements. The arithmetic mean is the sum of the inter-message interval samples divided by the number of samples. For more detailed discussion of statistical analyses, see Papoulis [B25].

NOTE 2—If useMgtSettableLogAnnounceInterval is FALSE, the value of initialLogAnnounceInterval is the value of the mean time interval between successive Announce messages when the PTP Port is initialized. The value of the mean time interval between successive Announce messages can be changed, e.g., if the PTP Port receives a Signaling message that carries a message interval request TLV (see 10.6.4.3) and the current value is stored in currentLogAnnounceInterval. The value of the mean time interval between successive Announce messages can be reset to the initial value, e.g., by a message interval request TLV for which the value of the field logAnnounceInterval is 126 (see 10.6.4.3.8).

NOTE 3—A PTP Port that requests (using a Signaling message that contains a message interval request TLV; see 10.6.4 and 10.3.17) that the PTP Port at the other end of the attached link set its currentLogAnnounceInterval to a specific value can determine if the request was honored by examining the logMessageInterval field of subsequent received Announce messages.

10.7.2.3 Time-synchronization event message transmission interval

The logarithm to the base 2 of the sync interval (in seconds) is carried in the logMessageInterval field of the time-synchronization messages.

When useMgtSettableLogSyncInterval (see 14.8.19) is FALSE, the initialLogSyncInterval specifies the sync interval when the PTP Port is initialized and the value to which the sync interval is set when a message interval request TLV is received with the logTimeSyncInterval field set to 126 (see the SyncIntervalSetting state machine in 10.3.18). The default value is media-dependent; the value is specified in the respective media-dependent clauses. The initialLogSyncInterval is a per-PTP Port attribute.

The currentLogSyncInterval specifies the current value of the sync interval and is a per-PTP Port attribute.

When useMgtSettableLogSyncInterval is TRUE, currentLogSyncInterval is set equal to mgtSettableLogSyncInterval (see 14.8.20), and initialLogSyncInterval is ignored.

When the value of syncLocked is FALSE, time-synchronization messages shall be transmitted such that the value of the arithmetic mean of the intervals, in seconds, between message transmissions is within ± 30% of $2^{\text{currentLogSyncInterval}}$. In addition, a PTP Port shall transmit time-synchronization messages such that at least 90% of the inter-message intervals are within ± 30% of the value of $2^{\text{currentLogSyncInterval}}$. The interval between successive time-synchronization messages should not exceed twice the value of $2^{\text{portDS.logSyncInterval}}$ in order to prevent causing a syncReceiptTimeout event. The PortSyncSyncSend state machine (see 10.2.12) is consistent with these requirements, i.e., the requirements here and the requirements of the PortSyncSyncSend state machine can be met simultaneously.

NOTE 1—A minimum number of inter-message intervals is necessary in order to verify that a PTP Port meets these requirements. The arithmetic mean is the sum of the inter-message interval samples divided by the number of samples. For more detailed discussion of statistical analyses, see Papoulis [B25].

NOTE 2—If useMgtSettableLogSyncInterval is FALSE the value of initialLogSyncInterval is the value of the sync interval when the PTP Port is initialized. The value of the sync interval can be changed, e.g., if the PTP Port receives a Signaling message that carries a message interval request TLV (see 10.6.4.3) and the current value is stored in currentLogSyncInterval. The value of the sync interval can be reset to the initial value, e.g., by a message interval request TLV for which the value of the field logTimeSyncInterval is 126 (see 10.6.4.3.7).

10.7.2.4 Interval for providing synchronization information by ClockMaster entity

The clockMasterLogSyncInterval specifies the mean time interval between successive instants at which the ClockMaster entity provides time-synchronization information to the SiteSync entity. The value is less than or equal to the smallest currentLogSyncInterval (see 10.7.2.3) value for all the ports of the PTP Instance. The clockMasterLogSyncInterval is an internal, per PTP Instance variable.

10.7.2.5 Interval for sending the gPTP-capable TLV Signaling message

The logarithm to the base 2 of the gPTP-capable message interval (in seconds) is carried in the logGptpCapableMessageInterval field of the gPTP-capable TLV. The default value shall be 0. The range shall be –24 through 24. Other values in the range [–128, 127] shall be reserved.

When useMgtSettableLogGptpCapableMessageInterval (see 14.8.14) is FALSE, the initialLogGptpCapableMessageInterval specifies the following:

a) The mean gPTP-capable message interval when the PTP Port is initialized, and
b) The value to which the gPTP-capable message interval is set when a gPTP-capable TLV is received with the logGptpCapableMessageInterval field set to 126 (see the GptpCapableMessageIntervalSetting state machine in 10.4.3).

The currentLogGptpCapableMessageInterval specifies the current value of the gPTP-capable message interval. Every PTP Port supports the value 127; the PTP Port does not send Signaling messages containing a gPTP-capable TLV when currentLogGptpCapableMessageInterval has this value (see 10.4.3). A PTP Port may support other values, except for the reserved values indicated in Table 10-21. A PTP Port shall ignore requests for unsupported values (see 10.4.3). The initialLogGptpCapableMessageInterval and currentLogGptpCapableMessageInterval are per-PTP Port attributes.

When useMgtSettableLogGptpCapableMessageInterval is TRUE, currentLogGptpCapableMessageInterval is set equal to mgtSettableLogGptpCapableMessageInterval (see 14.8.15), and initialLogGptpCapableMessageInterval is ignored.

NOTE 1—If useMgtSettableLogGptpCapableMessageInterval is FALSE, the value of initialLogGptpCapableMessageInterval is the value of the mean time interval between successive Signaling messages containing the gPTP-capable TLV when the PTP Port is initialized. The value of the mean time interval between successive Signaling messages containing the gPTP-capable TLV can be changed, e.g., if the PTP Port receives a Signaling message that carries a gPTP-capable TLV (see 10.6.4.4) and the current value is stored in currentLogGptpCapableMessageInterval. The value of the mean time interval between successive Signaling messages containing the gPTP-capable TLV can be reset to the initial value, e.g., by a Signaling message containing a gPTP-capable message interval request TLV for which the value of the field logGptpCapableMessageInterval is 126 (see 10.6.4.5.6).

NOTE 2—A PTP Port that requests (using a Signaling message that contains a gPTP-capable message interval request TLV; see 10.6.4 and 10.6.4.5) that the PTP Port at the other end of the attached link set its currentLogGptpCapableMessageInterval to a specific value can determine if the request was honored by examining the logGptpCapableMessageInterval field of subsequent received Signaling messages containing the gPTP-capable TLV.

NOTE 3—The GptpCapableTransmit state machine ensures that the times between transmission of successive Signaling messages, containing the gPTP-capable TLV, in seconds, are not smaller than $2^{\mathrm{currentLogGptpCapableMessaggeInterval}}$. This is consistent with the manner in which Pdelay_Req messages are transmitted (see NOTE 3 of 11.5.2.2).

10.7.3 Timeouts

10.7.3.1 syncReceiptTimeout

The value of this attribute tells a slave port the number of sync intervals to wait without receiving synchronization information, before assuming that the master is no longer transmitting synchronization information and that the BMCA needs to be run, if appropriate. The condition of the slave port not receiving synchronization information for syncReceiptTimeout sync intervals is known as *sync receipt timeout.*

The default value shall be 3. The syncReceiptTimeout is a per-PTP Port attribute.

10.7.3.2 announceReceiptTimeout

The value of this attribute tells a slave port the number of announce intervals to wait without receiving an Announce message, before assuming that the master is no longer transmitting Announce messages, and that the BMCA needs to be run, if appropriate. The condition of the slave port not receiving an Announce message for announceReceiptTimeout announce intervals is known as *announce receipt timeout.*

The default value shall be 3. The announceReceiptTimeout is a per-PTP Port attribute.

10.7.3.3 gPtpCapableReceiptTimeout

The value of this attribute tells a PTP Port the number of gPTP-capable message intervals to wait without receiving from its neighbor a Signaling message containing a gPTP-capable TLV, before determining that its neighbor is no longer invoking gPTP.

NOTE—A determination that its neighbor is no longer invoking gPTP will cause the PTP Port to set asCapable to FALSE.

The default value shall be 9. The range shall be 1 through 255.