2. Normative references

Change the Text of 2., as shown

The following referenced documents are indispensable for the application of this standard (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE P802.11v™ (D15.0, September 2010), Draft Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications—Amendment 8: IEEE 802.11™ Wireless Network Management.


<<Editor’s Note: IEEE 802.11v is referred to as P802.11v in the following 13 subclauses in the text:
5.5 (p. 12)
6.3 (p. 15)
7.3.2 (p. 22)
7.3.4 (p. 23)
7.4 (p. 24)
12.1 (p. 133)
12.1.1 (p. 133; 2 references, one in heading and one in text)
12.1.2 (p.134)
12.2 (p. 135)
12.3 (p. 135)
12.4.1.6.1 (p. 139)
12.4.1.6.2 (p. 140)

Each of these instances of a reference to IEEE P802.11v should be changed to IEEE 802.11v. Do we need to show each instance explicitly, with change bars, or is it sufficient to make a global statement that all references in the document to “IEEE P802.11v” should be changed to “IEEE 802.11v” (perhaps also stating where each instance is, as is done above)? >>

⁴IEEE P802.11v/D16 (November 2010) was approved by the IEEE-SA Standards Board on 2 February 2011. It was published as IEEE Std 802.11v-2011 on 9 February 2011 and is available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org).
IEEE Std 802.1D™-2004, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Bridges.¹, ⁴


IEEE Std 802.3™-2008, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area network—Specific requirements, Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.


IETF RFC 3410 (December 2002), Introduction and Applicability Statements for Internet Standard Management Framework, Case, J., Mundy, R., Partain, D, and Stewart, B.⁵

ITU-T Recommendation G.9960 (ex. G.hn), Unified high-speed wire-line based home networking tranceivers—System architecture and physical layer specification, June 2010.⁶

ITU-T Recommendation G.9961, Data link layer (DLL) for unified high-speed wire-line based home networking tranceivers, June 2010.


MoCA® MACPHY Specification v2.0, MoCA-M/P-SPEC-V2.0-20100507, Multimedia over Coax Alliance (MoCA).⁷

8. IEEE 802.1AS concepts and terminology

¹IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org).

⁴The IEEE standards or products referred to in Clause 2 are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.


⁶ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (http://www.itu.int/).

⁷MoCA specifications are available from the Multimedia over Coax Alliance at http://www.mocalliance.org/specs.
8.6 Time-aware system characterization

8.6.2 Time-aware system attributes

8.6.2.4 offsetScaledLogVariance

*Change the Text of 8.6.2.4, as shown*

The offsetScaledLogVariance is a scaled, offset representation of an estimate of the PTP variance. The PTP variance characterizes the precision and frequency stability of the ClockMaster. The PTP variance is the square of PTPDEV (see B.1.3.2).

The value shall be selected as follows:

a) offsetScaledLogVariance is set to the value that reflects the combination of the LocalClock and ClockSource entities; else
b) if the value that reflects these entities is not specified or not known, offsetScaledLogVariance is set to \(16640.17258\) \((4100\_{16}436A_{16})\). This value corresponds to the value of PTPDEV for observation interval equal to the default Sync message transmission interval (i.e., observation interval of 0.125 s, see 11.5.2.3 and B.1.3.2).

See 7.6.3 of IEEE Std 1588-2008 for more detailed description of PTP variance and offsetScaledLogVariance (7.6.3.3 of IEEE Std 1588-2008 provides a detailed description of the computation of offsetScaledLogVariance from PTP variance, along with an example).

10. Media-independent layer specification

10.2 Time-synchronization state machines

10.2.10 ClockMasterSyncReceive state machine

10.2.10.2 State machine functions

10.2.10.2.2 updateMasterTime()

*Change the Text of 10.2.10.2.2, as shown*

updateMasterTime(): updates the global variable masterTime (see 10.2.3.21), 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 received, and can be incremented by
localClockTickInterval (see 10.2.3.18) divided multiplied by gmRateRatio (see 10.2.3.14) when
rcvdLocalClockTick is TRUE.

10.2.12 ClockSlaveSync state machine

10.2.12.2 State machine functions

10.2.12.2.1 updateSlaveTime()

Change the Text of 10.2.12.2.1, as shown

updateSlaveTime(): updates the global variable clockSlaveTime (see 10.2.3.3), 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 incremented by localClockTickInterval (see
10.2.3.18) divided multiplied by rateRatio member of the received PortSyncSync structure. If no time-aware system
is grandmaster-capable, i.e., gmPresent is FALSE, then clockSlaveTime is set to the time provided by the
LocalClock. This function is invoked when rcvdLocalClockTick is TRUE.

10.3 Best master clock selection and announce interval setting state machines

10.3.8 Per-time-aware system global variables

10.3.8.1 reselect

Change the Text of 10.3.8.1, as shown

reselect: a Boolean array of length numberPorts (see 8.6.1). Setting reselect[j], where 0 £ j £ numberPorts, to TRUE
causes the ROLE_SELECTION block of the PortRoleSelection state machine (see 10.3.12) to be re-entered, which
in turn causes the port role of each port of the time-aware system to be updated (via the function
updtRolesTree(), see 10.3.12.1.4).

10.3.12 PortRoleSelection state machine

10.3.12.1 State machine functions

10.3.12.1.4 updtRolesTree()

Change the Text of 10.3.12.1.4, as shown

10.3.12.1.4 updtRolesTree(): performs the following operations (see 10.3.4 and 10.3.5 for details on the
priority vectors):
   a) Computes the gmPathPriorityVector for each 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 10.3.8.19) in lastGmPriority (see 10.3.8.20), computes the

gmPriorityVector for the time-aware system and saves it in gmPriority, chosen as the best of the
set consisting of the systemPriorityVector (for this time-aware system) and the

gmPathPriorityVector for each port for which the clockIdentity of the master port is not equal to
thisClock (see 10.2.3.22),

c) Sets the per-time-aware system global variables leap61, leap59, currentUtcOffsetValid,
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, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource are set to annLeap61, annLeap59,
annCurrentUtcOffsetValid, annTimeTraceable, annFrequencyTraceable, annCurrentUtcOffset, and annTimeSource, respectively, for that port.

2) If the gmPriorityVector was set to the systemPriorityVector, then leap61, leap59,
currentUtcOffsetValid, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource are set to sysLeap61, sysLeap59, sysCurrentUtcOffsetValid,
sysTimeTraceable, sysFrequencyTraceable, sysCurrentUtcOffset, and sysTimeSource, respectively.

d) Computes the **first three components and the clockIdentity of the fourth component of the**

masterPriorityVector for each port. (i.e., everything except the portNumber of the fourth

component)

e) Computes masterStepsRemoved, which is equal to:

1) messageStepsRemoved (see 10.3.9.7) for the 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 port role for port j and sets selectedRole[j] equal to this port role, as follows, for j = 1, 2, ..., numberPorts:

3) If the port is disabled (infoIs == Disabled), selectedRole[j] is set to DisabledPort.

4) If announcement timeout, or sync receipt timeout with gmPresent set to TRUE, have
occurred (infoIs == Aged), updtInfo is set to TRUE and selectedRole[j] is set to
MasterPort.

5) If the portPriorityVector was derived from another port on the time-aware system or from
the time-aware system itself as the root (infoIs == Mine), selectedRole[j] is set to
MasterPort. In addition, updtInfo is set to TRUE if the portPriorityVector differs from the
masterPriorityVector or portStepsRemoved differs from masterStepsRemoved.

6) If the portPriorityVector was received in an Announcement and announcement receipt
timeout, or sync receipt timeout with gmPresent TRUE, have not occurred (infoIs ==
Received), and the gmPriorityVector is now derived from the portPriorityVector,
selectedRole[j] is set to SlavePort and updtInfo is set to FALSE.

7) If the portPriorityVector was received in an Announcement and announce receipt
timeout, or sync receipt timeout with gmPresent TRUE, have not occurred (infoIs ==
Received), the gmPriorityVector is not now derived from the portPriorityVector, the
masterPriorityVector is not higher than the portPriorityVector, and the
sourcePortIdentity component of the portPriorityVector does not reflect another port on
the time-aware system, selectedRole[j] is set to PassivePort and updtInfo is set to FALSE.
8) If the portPriorityVector was received in an Announce message and announce receipt timeout, or sync receipt timeout with gmPresent TRUE, have not occurred (infoIs == Received), the gmPriorityVector is not now derived from the portPriorityVector, the masterPriorityVector is not higher than the portPriorityVector, and the sourcePortIdentity component of the portPriorityVector does reflect another port on the time-aware system, selectedRole[j] is set to PassivePort and updtInfo is set to FALSE.

<<Editor’s Note: A user has suggested combining (7) and (8) above. The only difference in the conditions is that in (7) the sourcePortIdentity component of the portPriorityVector does not reflect another port on the time-aware system, while in (8) the sourcePortIdentity component of the portPriorityVector does reflect another port on the time-aware system. However, in both cases (i.e., in both (7) and (8)), selectedRole[j] is set to PassivePort.

(7) and (8) could be combined to a single statement, which would read:
“If the portPriorityVector was received in an Announce message and announce receipt timeout, or sync receipt timeout with gmPresent TRUE, have not occurred (infoIs == Received), the gmPriorityVector is not now derived from the portPriorityVector, and the masterPriorityVector is not better than the portPriorityVector, selectedRole[j] is set to PassivePort and updtInfo is set to FALSE.”

The reason (7) and (8) are written as separate conditions is that they appeared as separate conditions in updtRolesTree() in IEEE 802.1D-2005 (p. 169, items (j) and (k)), and are separate conditions in IEEE 802.1Q-2011 (p. 378, items (l) and (m)). However, they are separate items in 802.1D-2005 and 802.1Q-2011 because in the formulation of RSTP in those documents a distinction is made between an AlternatePort (item (j) in 802.1D-2005 and item (l) in 802.1Q-2011) and a BackupPort (item (k) in 802.1D-2005 and item (m) in 802.1Q-2011). However, in the BMCA formulation in 802.1AS, the port role is set to PassivePort in both cases. Therefore, it is not strictly necessary to have items (7) and (8) as separate conditions; they could be combined as indicated above.

However, if items (7) and (8) are kept as separate conditions, the formulation of BMCA in 802.1AS is not incorrect; it is simply not as efficient as it might be. In this sense, combining these statements might not be appropriate for the corrigendum, since the purpose of the corrigendum is to fix bugs. However, they could be combined in the 802.1AS amendment, which is planned to be worked on concurrently with the corrigendum. Therefore, it needs to be decided whether to (A) combine items (7) and (8) in the 802.1AS corrigendum, (B) combine items (7) and (8) in the 802.1AS amendment, or (C) leave items (7) and (8) as is. >>

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

g) Updates gmPresent as follows:
   10) gmPresent is set to TRUE if the priority1 field of the rootSystemIdentity of the gmPriorityVector is less than 255.
   11) gmPresent is set to FALSE if the priority1 field of the rootSystemIdentity of the gmPriorityVector is equal to 255.

h) Assigns the port role for port 0, and sets selectedRole[0], as follows:
   12) if selectedRole[j] is set to SlavePort for any port with portNumber j, j = 1, 2, ..., numberPorts, selectedRole[0] is set to PassivePort.
   13) if selectedRole[j] is not set to SlavePort for any port with portNumber j, j = 1, 2, ..., numberPorts, selectedRole[0] is set to SlavePort.

i) If the clockIdentity member of the systemIdentity (see 10.3.2) member of gmPriority (see 10.3.8.19) is equal to thisClock (see 10.2.3.22), i.e., if the current time-aware system is the grandmaster, the pathTrace array is set to contain the single element thisClock (see 10.2.3.22).

10.3.14 AnnounceIntervalSetting state machine

10.3.14.2 State diagram

*Change the Figure of 10-16, as shown*
10.5 Message formats

10.5.2 Header

10.5.2.2 Header field specifications

10.5.2.6 Flags (Octet2)

Change the Text of 10.5.2.6, as shown
The value of the bits of the array are defined in **Table 10-6–Values of flag bits**. For message types where the bit is not defined in **Table 10-6–Values of flag bits**, the value of the bit is set to FALSE.

<table>
<thead>
<tr>
<th>Octet</th>
<th>Bit</th>
<th>Message Types</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01</td>
<td>Announce</td>
<td>leap61</td>
<td>The value of the global variable leap61 (see Error! Reference source not found.)</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>Announce</td>
<td>leap59</td>
<td>The value of the global variable leap59 (see Error! Reference source not found.)</td>
</tr>
<tr>
<td>1</td>
<td>23</td>
<td>Announce</td>
<td>currentUtcOffsetValid</td>
<td>The value of the global variable currentUtcOffsetValid (see Error! Reference source not found.)</td>
</tr>
<tr>
<td>1</td>
<td>34</td>
<td>All</td>
<td>ptpTimescale</td>
<td>Reserved as TRUE, ignored on reception</td>
</tr>
<tr>
<td>1</td>
<td>45</td>
<td>Announce</td>
<td>timeTraceable</td>
<td>The value of the global variable timeTraceable (see Error! Reference source not found.)</td>
</tr>
<tr>
<td>1</td>
<td>56</td>
<td>Announce</td>
<td>frequencyTraceable</td>
<td>The value of the global variable frequencyTraceable (see Error! Reference source not found.)</td>
</tr>
</tbody>
</table>

11. Media-dependent layer specification for full-duplex, point-to-point links

11.1 Overview

11.1.3 Transport of time-synchronization information

*Change the Text of 11.1.3, as shown*

The transport of time-synchronization information by a time-aware system, using Sync and Follow_Up messages, is illustrated in Figure 11-2. The mechanism is mathematically equivalent to the mechanism described in IEEE Std 1588-2008 for a two-step, peer-to-peer transparent clock that is syntonized (see 11.4.5.1, 11.5.1, and 11.5.2.2 of IEEE Std 1588-2008). However, as will be seen shortly, the processes of transporting synchronization by a two-step, peer-to-peer transparent clock that is syntonized and by a two-step boundary clock are mathematically and
functionally equivalent. The main functional difference between the two types of clocks is that the boundary clock participates in best master selection and invokes the BMCA, while the peer-to-peer transparent clock does not participate in best master selection and does not invoke the BMCA (and implementations of the two types of clocks can be different).

Figure 11-2 – Transport of time-synchronization information

Figure 11-2 shows three adjacent time-aware systems, indexed \( i-1 \), \( i \), and \( i+1 \). Synchronization is transported from time-aware system \( i-1 \) to time-aware system \( i \), and then to time-aware system \( i+1 \). Time-aware system \( i-1 \) sends a Sync message to time-aware system \( i \) at time \( t_{s,i-1} \), relative to the LocalClock entity of time-aware system \( i-1 \). At a later time, time-aware system \( i-1 \) sends an associated Follow_Up message to time-aware system \( i \), which contains a preciseOriginTimestamp, correctionField\(_{i-1} \), and rateRatio\(_{i-1} \). The preciseOriginTimestamp contains the time of the grandmaster when it originally sent this synchronization information. It is not indexed here because it normally does not change as the Sync and Follow_Up messages traverse the network. The quantity correctionField\(_{i-1} \) contains the difference between the synchronized time when the Sync message is sent (i.e., the synchronized time that corresponds to the local time \( t_{s,i-1} \)) and the preciseOriginTimestamp. The sum of preciseOriginTimestamp and correctionField\(_{i-1} \) gives the synchronized time that corresponds to \( t_{s,i-1} \). The quantity rateRatio\(_{i-1} \) is the ratio of the grandmaster frequency to the frequency of the LocalClock entity of time-aware system \( i-1 \).

Time-aware system \( i \) receives the Sync message from time-aware system \( i-1 \) at time \( t_{s,i} \), relative to its LocalClock entity. It timestamps the receipt of the Sync message, and the timestamp value is \( t_{r,i} \). It receives the associated Follow_Up message some time later.

Time-aware system \( i \) will eventually send a new Sync message at time \( t_{s,i} \), relative to its LocalClock entity. It will have to compute correctionField\(_{i} \), i.e., the difference between the synchronized time that corresponds to \( t_{s,i} \) and the preciseOriginTimestamp. To do this, it must compute the value of the time interval between \( t_{s,i-1} \) and \( t_{s,i} \), expressed in the grandmaster time base. This interval is equal to the sum of the following quantities:

---

8 The same mathematical and functional equivalence exists for one-step boundary and syntonized peer-to-peer transparent clocks. One-step clocks are not discussed here because time-aware systems described in this standard are two-step devices from the standpoint of IEEE Std 1588-2008.
a) The propagation delay on the link between time-aware systems \(i-1\) and \(i\), expressed in the grandmaster time base, and

b) The difference between \(t_{s,i}\) and \(t_{r,i}\) (i.e., the residence time), expressed in the grandmaster time base.

The mean propagation delay on the link between time-aware systems \(i-1\) and \(i\), relative to the LocalClock entity of time-aware system \(i-1\), is equal to \(\text{neighborPropDelay}\) (see 10.2.4.7). This must be divided by \(\text{neighborRateRatio}\) (see 10.2.4.6) to express it in the grandmaster time base. The total propagation delay is equal to the mean propagation delay plus the quantity \(\text{delayAsymmetry}\) (see 8.3 and 10.2.4.8); \(\text{delayAsymmetry}\) is already expressed in the grandmaster time base. The residence time, \(t_{s,i}-t_{r,i}\), must be multiplied by \(\text{rateRatio}\) to express it in the grandmaster time base.

The preceding computation is organized slightly differently in the state machines of 11.2.13 and 11.2.14. Rather than explicitly expressing the link propagation delay in the grandmaster time base, the local time at time-aware system \(i\) that corresponds to \(t_{s,i-1}\) is computed; this is the upstreamTxTime member of the MDSyncReceive structure (see 10.2.2.6; recall that \(t_{s,i-1}\) is relative to the LocalClock entity of time-aware system \(i-1\)). upstreamTxTime is equal to the quantity \(t_{s,i}\) minus the link propagation delay expressed relative to the LocalClock entity of time-aware system \(i\). The link propagation delay expressed relative to the LocalClock entity of time-aware system \(i\) is equal to the sum of the following:

c) The quantity \(\text{neighborPropDelay}\) divided by \(\text{neighborRateRatio}\), and
d) The quantity \(\text{delayAsymmetry}\) divided by \(\text{rateRatio}\).

The division of \(\text{delayAsymmetry}\) by \(\text{rateRatio}\) is performed after \(\text{rateRatio}\) has been updated, as described shortly. The computation of upstreamTxTime is done by the MDSyncReceiveSM state machine in the function \(\text{setMDSyncReceive()}\) (see 11.2.13.2.1). When time-aware system \(i\) sends a Sync message to time-aware system \(i+1\), it computes the sum of the link propagation delay and residence time, expressed in the grandmaster time base, as:

e) The quantity \((t_{s,i} - \text{upstreamTxTime})(\text{rateRatio})\).

As in item The quantity \(\text{delayAsymmetry}\) above, this computation is performed after \(\text{rateRatio}\) has been updated, as described shortly. The quantity of item The quantity \((\) is added to correctionField\(_{i-1}\) to obtain correctionField\(_i\). The computation of item The quantity \((\) and correctionField\(_i\), is done by the MDSyncSendSM state machine in the function \(\text{setFollowUp()}\) (see 11.2.14.2.3). The quantity correctionField, is inserted in the Follow_Up message sent by time-aware system \(i\).

Note that the difference between mean propagation delay relative to the grandmaster time base and relative to the time bases of the time-aware system at the other end of the attached link or of the current time-aware system is usually negligible. To see this, note that the former can be obtained from the latter by multiplying the latter by the ratio of the grandmaster frequency to the frequency of the LocalClock entity of the time-aware system at the other end of the link attached to this port. This ratio differs from 1 by 200 ppm or less. For example, for a worst-case frequency offset of the LocalClock entity of the time-aware system at the other end of the link, relative to the grandmaster, of 200 ppm, and a measured propagation time of 100 ns, the difference in \(D\) relative to the two time bases is 20 ps. The corresponding difference for link delay asymmetry in this example is also negligible because the magnitude of the link delay asymmetry is of the same order of magnitude as the mean propagation time, or less. However, the difference is usually not negligible for residence time, because residence time can be much larger (see B.2.2).

It was previously indicated that the processes of transporting synchronization by a two-step, peer-to-peer transparent clock that is syntonized and by a two-step boundary clock are mathematically and functionally equivalent. This is because the computations described above compute the synchronized time when the Sync message is sent by the time-aware system. The same computations are done if time-aware system \(i\) sends a Sync message without having received a new Sync message, i.e., if Sync receipt timeout occurs (see 10.6.3.1). In this case, time-aware system \(i\) uses the most recently received time-synchronization information from time-aware system \(i-1\), which would be prior to time-aware system \(i\) having sent its most recent Sync message. The synchronized time corresponding to the sending of a Sync message is equal to the sum of the preciseOriginTimestamp and correctionField. Normally a
boundary clock places this entire value, except for any sub-nanosecond portion, in the preciseOriginTimestamp, while a transparent clock retains the preciseOriginTimestamp and updates the correction field. However, the sum of the two fields is equal to the synchronized time when the Sync message is sent in both cases.

The ratio of the grandmaster frequency to the frequency of the LocalClock entity at time-aware system $i$, rateRatio$_i$, is equal to the same quantity at time-aware system $i-1$, rateRatio$_{i-1}$, multiplied by the ratio of the frequency of the LocalClock entity at time-aware system $i-1$ to the frequency of the LocalClock entity at time-aware system $i$, neighborRateRatio (see 10.2.4.6). If neighborRateRatio is sufficiently small, this is approximately equal to the sum of rateRatio$_{i-1}$ and the quantity neighborRateRatio−1, which is the frequency offset of time-aware system $i-1$ relative to time-aware system $i$. This computation is done by the PortSyncSyncReceive state machine (see 10.2.7).
11.2 State machines for MD entity specific to full-duplex, point-to-point links

11.2.17 LinkDelaySyncIntervalSetting state machine

11.2.17.2 State Diagram

*Change the Figure of 11-10, as shown*
BEGIN || !portEnabled || !pttPortEnabled

NOT_ENABLED

portEnabled && pttPortEnabled

INITIALIZE

currentLogPdelayReqInterval = initialLogPdelayReqInterval;
currentLogSyncInterval = initialLogSyncInterval;
pdelayReqInterval = (109) * 2^currentLogPdelayReqInterval;
syncInterval = (109) * 2^currentLogSyncInterval;
computeNeighborRateRatio = TRUE;
computeNeighborPropDelay = TRUE;
rcvdSignalingMsg1 = FALSE;

SET_INTERVALS

switch (rcvdSignalingPtr->linkDelayInterval)
{
    case (-128): /* don't change the interval */
        break;
    case 126: /* set interval to initial value */
        currentLogPdelayReqInterval = initialLogPdelayReqInterval;
pdelayReqInterval = (109) * 2^currentLogPdelayReqInterval;
        break;
    default: /* use indicated value; note that the value of 127 will result in an interval of
              * 2^(127) s, or approximately 5.4 \times 10^{30} years, which indicates that the Pdelay
              * requester should stop sending for all practical purposes, in accordance
              * with Table 10-9. */
pdelayReqInterval = (109) * 2^rcvdSignalingPtr->linkDelayInterval;
currentLogPdelayReqInterval = rcvdSignalingPtr->linkDelayInterval;
        break;
}

switch (rcvdSignalingPtr->timeSyncInterval)
{
    case (-128): /* don't change the interval */
        break;
    case 126: /* set interval to initial value */
currentLogSyncInterval = initialLogSyncInterval;
        syncInterval = (109) * 2^currentLogSyncInterval;
        break;
    default: /* use indicated value; note that the value of 127 will result in an interval of
              * 2^(127) s, or approximately 5.4 \times 10^{30} years, which indicates that the sender
              * should stop sending for all practical purposes, in accordance
              * with Table 10-10. */
syncInterval = (109) * 2^rcvdSignalingPtr->timeSyncInterval;
currentLogSyncInterval = rcvdSignalingPtr->timeSyncInterval;
        break;
}

computeNeighborRateRatio = rcvdSignalingPtr->flags.computeNeighborRateRatio;
computeNeighborPropDelay = rcvdSignalingPtr->flags.computeNeighborPropDelay;
rcvdSignalingMsg1 = FALSE;
15. Managed object definitions

Change the Text of 15.5, ieee8021AsTimeSyncMib as shown

```snmp
ieee8021AsTimeSyncMib MODULE-IDENTITY
   LAST-UPDATED "201011100000Z201105100000Z" -- November 11, 2010May 10, 2011
   ORGANIZATION "IEEE 802.1 Working Group"
   CONTACT-INFO
      "WG-URL: http://www.ieee802.org/1/index.html
       WG-EMail: STDS-802-1@IEEE.ORG
       Contact: Geoffrey M. Garner
       Postal: 196 Ambassador Drive
              Red Bank, NJ 07701
              USA
       E-mail: gmgarner@alum.mit.edu"
   DESCRIPTION
      "The Management Information Base module for
       IEEE 802.1AS time synchronization protocol."
   REVISION "201011100000Z201105100000Z" -- November 11, 2010May 10, 2011
   DESCRIPTION
      "Published as part of IEEE Std 802.1AS
       Copyright (C) IEEE (2011)."
```

Change the Text of 15.5, ieee8021AsParentDSGrandmasterPriority1 as shown

```snmp
ieee8021AsParentDSGrandmasterPriority1 OBJECT-TYPE
   SYNTAX      Unsigned32(0..255)
   MAX-ACCESS  read-writeonly
   STATUS      current
   DESCRIPTION
      "Grandmaster’s most-significant priority declaration in
       the execution of the best master clock algorithm.
       Lower values take precedence. The default value is set
       to ieee8021AsDefaultDSPriority1."
   REFERENCE   "14.4.7"
   ::= { ieee8021AsParentDS 8 }
```

Change the Text of 15.5, ieee8021AsParentDSGrandmasterPriority2 as shown

```snmp
ieee8021AsParentDSGrandmasterPriority2 OBJECT-TYPE
```
SYNTAX      Unsigned32(0..255)
MAX-ACCESS  read-writeonly
STATUS      current
DESCRIPTION
"Grandmaster’s least-significant priority declaration in
the execution of the best master clock algorithm.
Lower values take precedence. The default value is set to
ieee8021AsDefaultDSDSPriority2."
REFERENCE   "14.4.8"
::= { ieee8021AsParentDS 9 }

Change the Text of 15.5, ieee8021AsTimePropertiesDSTimeSource as shown

ieee8021AsTimePropertiesDSTimeSource OBJECT-TYPE
SYNTAX      IEEE8021ASTimeSourceValue
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The value is timeSource for the current grandmaster
(see 14.2.15). It is equal to the global variable
timeTraceable (see 10.3.8.10).
Indicates the source of time used by the grandmaster
clock.
The default value is set to
ieee8021AsDefaultDSTimeSource."

Change the Text of 15.5, ieee8021AsPortDSAnnounceReceiptTimeout as shown

ieee8021AsPortDSAnnounceReceiptTimeout OBJECT-TYPE
SYNTAX      Unsigned32(0..255)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"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 BMC algorithm needs to be
run, if appropriate. The condition of the slave port
not receiving synchronization information for
csyncReceiptTimeout sync intervals is referred to as
‘sync receipt timeout’.

"The value is the number of Announce message
transmission intervals that a slave port waits without
receiving an Announce message, before assuming
that the master is no longer transmitting Announce messages, and that the BMC Algorithm needs to be run, if appropriate. The condition of the slave port not receiving an Announce message for announceReceiptTimeout announce intervals is referred to as ‘announce receipt timeout’.

The default value is 2–3 (see 10.6.3.2).

The contents of this variable SHALL be maintained across a restart of the system.

Change the Text of 15.5, ieee8021AsPortDSSyncReceiptTimeout as shown

ieee8021AsPortDSSyncReceiptTimeout OBJECT-TYPE
SYNTAX      Unsigned32(0..255)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION "The value is the number of time-synchronization transmission intervals that a slave port waits 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 referred to as ‘sync receipt timeout’.

The initialization value is 3 (see 10.6.3.1).

The contents of this variable SHALL be maintained across a restart of the system.

Annex B

(normative)

Performance requirements
B.1 LocalClock requirements

B.1.2 Noise generation

B.1.2.2 Wander generation

Change the Text of B.1.3.2, as shown

Wander generation is specified using the Time Deviation (TDEV) parameter. The corresponding values of the Allan Deviation (ADEV) and PTP Deviation (PTPDEV) are given for information; the former is also useful in describing the wander generation of clocks and oscillators, and the latter is related to the offsetScaledLogVariance attribute (see 8.6.2.4). Information on ADEV and TDEV are contained in ITU-T G.810 [B13] and IEEE Std 1139TM-1999 [B6]. Information on ADEV and PTPDEV are contained in 7.6.3 of IEEE Std 1588-2008.

TDEV, denoted \( \sigma_x(\tau) \), is estimated from a set of measurements, as shown in Equation (B.1):

\[
\sigma_x(\tau) = \sqrt{\frac{1}{6n^2(N-3n+1)} \sum_{j=1}^{N-3n+1} \left[ \sum_{i=j}^{n+j-1} (x_{i+2n} - 2x_{i+n} + x_i) \right]^2}, \quad n = 1, 2, \ldots \left\lceil \frac{N}{3} \right\rceil
\]

where

- \( \tau = n\tau_0 = \) observation interval
- \( \tau_0 = \) sampling interval
- \( N = \) total number of samples \([N-1]\tau_0 = \) measurement interval
- \( \lfloor y \rfloor \) denotes the floor function, i.e., the greatest integer less than or equal to \( y \)
- \( x_i = \) measured phase (time) error at the \( i^{th} \) sampling time [the units of \( x_i \) and \( \sigma_x(\tau) \) are the same]

ADEV, denoted \( \sigma_y(\tau) \), is estimated from a set of measurements, as shown in Equation (B.2):

\[
\sigma_y(\tau) = \sqrt{\frac{1}{2n^2\tau_0^2(N-2n)} \sum_{i=1}^{N-2n} (x_{i+2n} - 2x_{i+n} + x_i)^2}, \quad n = 1, 2, \ldots \left\lceil \frac{N-1}{2} \right\rceil
\]

where the notation is the same as defined above for TDEV.

PTPDEV, denoted \( \sigma_{\text{PTP}}(\tau) \), is estimated from a set of measurements, as shown in Equation (B.3):

\[
\sigma_{\text{PTP}}(\tau) = \sqrt{\frac{1}{6n^2(N-2n)} \sum_{i=1}^{N-2n} (x_{i+2n} - 2x_{i+n} + x_i)^2}, \quad n = 1, 2, \ldots \left\lceil \frac{N-1}{2} \right\rceil
\]
\[ \sigma_{\text{PTP}}(\tau) = \sqrt{\frac{1}{6(N-2n)} \sum_{i=1}^{N-2n} (x_{i+2n} - 2x_{i+n} + x_i)^2}, \quad n = 1, 2, \ldots, \left\lfloor \frac{N-1}{2} \right\rfloor } \]

where the notation is the same as defined above for TDEV.

TDEV, ADEV, and PTPDEV are second-order statistics on the phase error. All three statistics are functions of second differences of the phase error. This means that these statistics are not affected by a constant frequency offset. This behavior is desired, because these statistics are used here to constrain noise generation.

TDEV for the LocalClock entity shall not exceed the mask of Table B.1 and Figure B.1, when measured using

a) A measurement interval that is at least 120 s (i.e., at least 12 times the longest observation interval),
b) A low-pass filter with 3 dB bandwidth of 10 Hz, first-order characteristic, and 20 dB/decade roll-off, and
c) A sampling interval \( t_0 \) that does not exceed 1/30 s.

### Table B.1 – Wander generation TDEV requirement for LocalClock entity

<table>
<thead>
<tr>
<th>TDEV limit</th>
<th>Observation interval ( \tau )</th>
</tr>
</thead>
<tbody>
<tr>
<td>No requirement</td>
<td>( \tau \leq 0.05 ) s</td>
</tr>
<tr>
<td>5.0( \tau ) ns</td>
<td>( 0.05 \leq \tau \leq 10 ) s</td>
</tr>
<tr>
<td>No requirement</td>
<td>( \tau &gt; 10 ) s</td>
</tr>
</tbody>
</table>
The ADEV limit that corresponds to the TDEV requirement of Table B.1 and Figure B.1 is shown in Table B.2 and Figure B.2, respectively.

Table B.2 – ADEV limit corresponding to wander generation requirement of Table B.1

<table>
<thead>
<tr>
<th>ADEV limit</th>
<th>Observation interval $\tau$</th>
</tr>
</thead>
<tbody>
<tr>
<td>No requirement</td>
<td>$\tau &lt; 0.05$ s</td>
</tr>
<tr>
<td>$1.054 \times 10^{-8}$</td>
<td>$0.05 \leq \tau \leq 10$ s</td>
</tr>
<tr>
<td>No requirement</td>
<td>$\tau &gt; 10$ s</td>
</tr>
</tbody>
</table>

Figure B.1 – Wander generation (TDEV) requirement for LocalClock entity
The PTPDEV limit that corresponds to the TDEV requirement of Table B.1 and Figure B.1 is shown in Table B.3 and Figure B.3, respect, respectively.

### Table B.3 – PTPDEV limit corresponding to wander generation requirement of Table B.1

<table>
<thead>
<tr>
<th>ADEV limit</th>
<th>Observation interval τ</th>
</tr>
</thead>
<tbody>
<tr>
<td>No requirement</td>
<td>τ &lt; 0.05 s</td>
</tr>
<tr>
<td>6.08τ ns</td>
<td>0.05 ≤ τ ≤ 10 s</td>
</tr>
<tr>
<td>No requirement</td>
<td>τ &gt; 10 s</td>
</tr>
</tbody>
</table>
Figure B.3 – PTPDEV limit corresponding to wander generation requirement of Figure B.1

Annex G

(informative)

Bibliography

*Change the Text of Annex G, as shown*


[B2] IEEE EUI-64, IEEE EUI-48, and IEEE MAC-48 assigned numbers may be obtained from the IEEE Registration Authority. ¹⁰

[B3] IEEE P802Std 802.1Qbb™-2011(D2.3, May 2010), Draft IEEE Standard for Local and Metropolitan Area Networks—Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks—Amendment 17: Priority-based Flow Control.11


[B5] IEEE Std 802.1X™-2010, IEEE Standard for Local and Metropolitan Area Networks—Port-Based Network Access Control.12, 13


10Available at http://standards.ieee.org/regauth/. Tutorials on these assigned numbers can be found on this Web site.

11Numbers preceded by P are IEEE authorized standards projects that were not approved by the IEEE-SA Standards Board at the time this publication went to press. For information about obtaining drafts, contact the IEEE.

12IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

13The IEEE standards or products referred to in Annex G are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.

14ISO publications are available from the ISO Central Secretariat, 1, ch. de la Voie-Creuse, Case Postale 56, CH-1211, Geneva 20, Switzerland (http://www.iso.org/). IEC publications are available from the Central Office of the International Electrotechnical Commission, 3, rue de Varembe, P.O. Box 131, CH-1211, Geneva 20, Switzerland (http://www.iec.ch/). ISO/IEC publications are also available in the United States from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

15ITU publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzer-land/Suisse (http://www.itu.int/).

16Available at https://kb.osu.edu/dspace/bitstream/1811/24301/1/Geom_Ref_Sys_Geodesy.pdf.
[B15] MoCA MACPHY Specification v1.0, MoCA-M-P-SPEC-V1.0-07122009, Multimedia over Coax Alliance (MoCA), July 12, 2009.17


[B17] Service de la rotation terrestre, Observatoire de Paris, 61, Av. de l’Observatoire 75014 Paris (France).


[B19] U.S. Naval Observatory.19

17MoCA specifications are available from the Multimedia over Coax Alliance at http://www.mocalliance.org/specs.

18Available at http://www.bipm.org/en/si/si_brochure/.