This document was prepared by Yong Kim and Geoff Garner.

May 12, 2011

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.[2]~~

IEEE Std 802.11v™ – 2011, IEEE 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.

IEEE Std 802.1D™-2004, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Bridges.[3, 4]

IEEE Std 802.1Q™- 2005, IEEE Standard for Local and metropolitan area networks—Virtual Bridged Local Area Networks.

IEEE Std 802.1ag™-2007, IEEE Standard for Local and metropolitan area networks—Virtual Bridged Local Area Networks—Amendment 5: Connectivity Fault Management.

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.

IEEE Std 802.3av™-2009, IEEE Standard for Information technology—Part 3: Amendment 1: Physical Layer Specifications and Management Parameters for 10 Gb/s Passive Optical Networks.

~~[2]IEEE 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).~~

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

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

IEEE Std 802.11™-2007, IEEE 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.

IEEE Std 1588™-2008, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.

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 transceivers, June 2010.

ITU-T Recommendation G.984.3, Amendment 2 (2009-11) Gigabit-capable Passive Optical Networks (G-PON): Transmission convergence layer specification—Time-of-day distribution and maintenance updates and clarifications, November 2009.

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

10. Media-independent layer specification

10.3 Best master clock selection and announce interval setting state machines

10.3.14 AnnounceIntervalSetting state machine

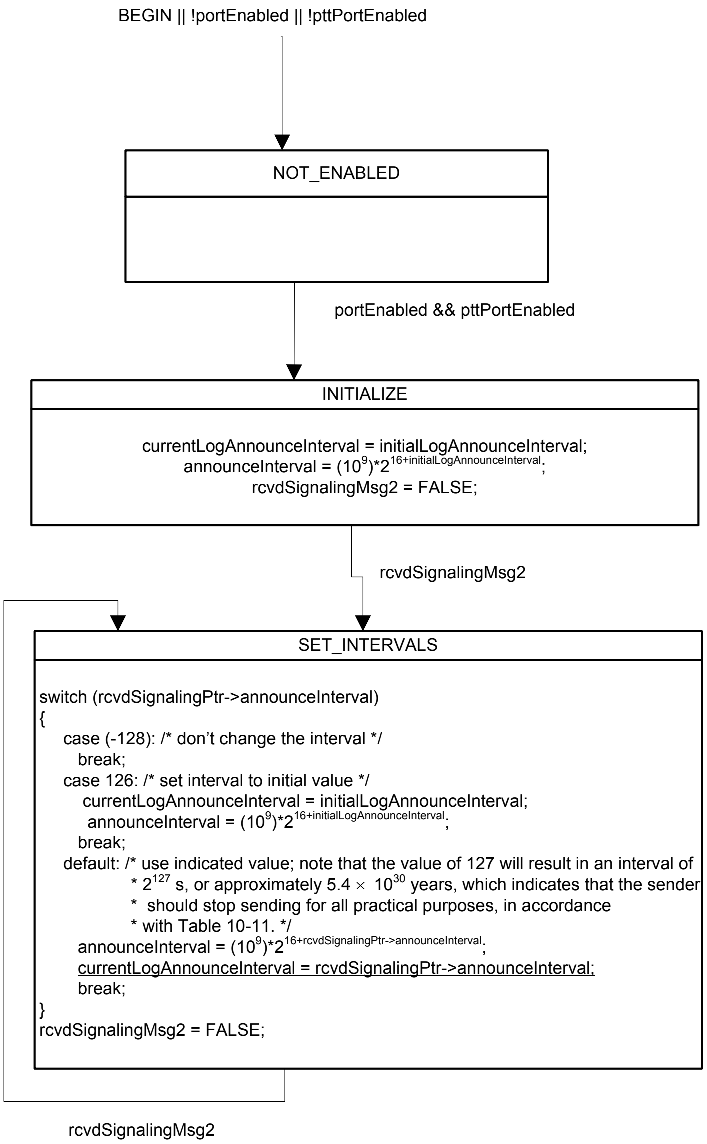
10.3.14.2 State diagram

Change the Figure of 10-16, as shown

[5]IETF RFCs are available from the Internet Engineering Task Force Web site at http://www.ietf.org/rfc.html.

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

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



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 – Transport of time-synchronization information**. The mechanism is mathematically equivalent to the mechanism described in IEEE Std 1588-2008 for a two-step,**Error! Reference source not found.** 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.[86] 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 – Transport of time-synchronization information 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 send 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 $t_{r,i}$, 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:

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 neighborPropDelay (see 10.2.4.7). This must be ~~divided~~ multiplied by rateRatio$_{i-1}$ to express it in the grandmaster time base. The total propagation delay is equal to the mean propagation delay plus the quantity delayAsymmetry (see 8.3 and 10.2.4.8); delayAsymmetry is already expressed in the grandmaster time base. The residence time, $t_{s,i}-t_{r,i}$, must be multiplied by rateRatio$_i$ to express it in the grandmaster time base.

The preceding computation is organized slightly differently in the state machines of **Error! Reference source not found.** and **Error! Reference source not found.**. 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 10.2.2.2.6; recall that $t_{s,i-1}$ is relative to the LocalClock entity of time-aware system i–1). upstreamTxTime is equal to the quantity $t_{r,i}$ minus the link propagation

[16]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.

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 neighborPropDelay (see 10.2.4.7) divided by neighborRateRatio (see 10.2.4.6), and
d) The quantity delayAsymmetry (see 10.2.4.8) divided by rateRatio_i.

The division of delayAsymmetry by rateRatio_i is performed after rateRatio_i has been updated, as described shortly. The computation of upstreamTxTime is done by the MDSyncReceiveSM state machine in the function setMDSyncReceive() (see **Error! Reference source not found.**). 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 ($t_{s,i}$ – upstreamTxTime)(rateRatio_i).

As in item The quantity delayAsymmetry (see 10.) above, this computation is performed after rateRatio_i has been updated, as described shortly. The quantity of item The quantity () is added to $\text{correctionField}_{i-}1$ to obtain correctionField_i. The computation of item The quantity () and correctionField_i is done by the MDSyncSendSM state machine in the function setFollowUp() (see **Error! Reference source not found.**). The quantity correctionField_i 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.

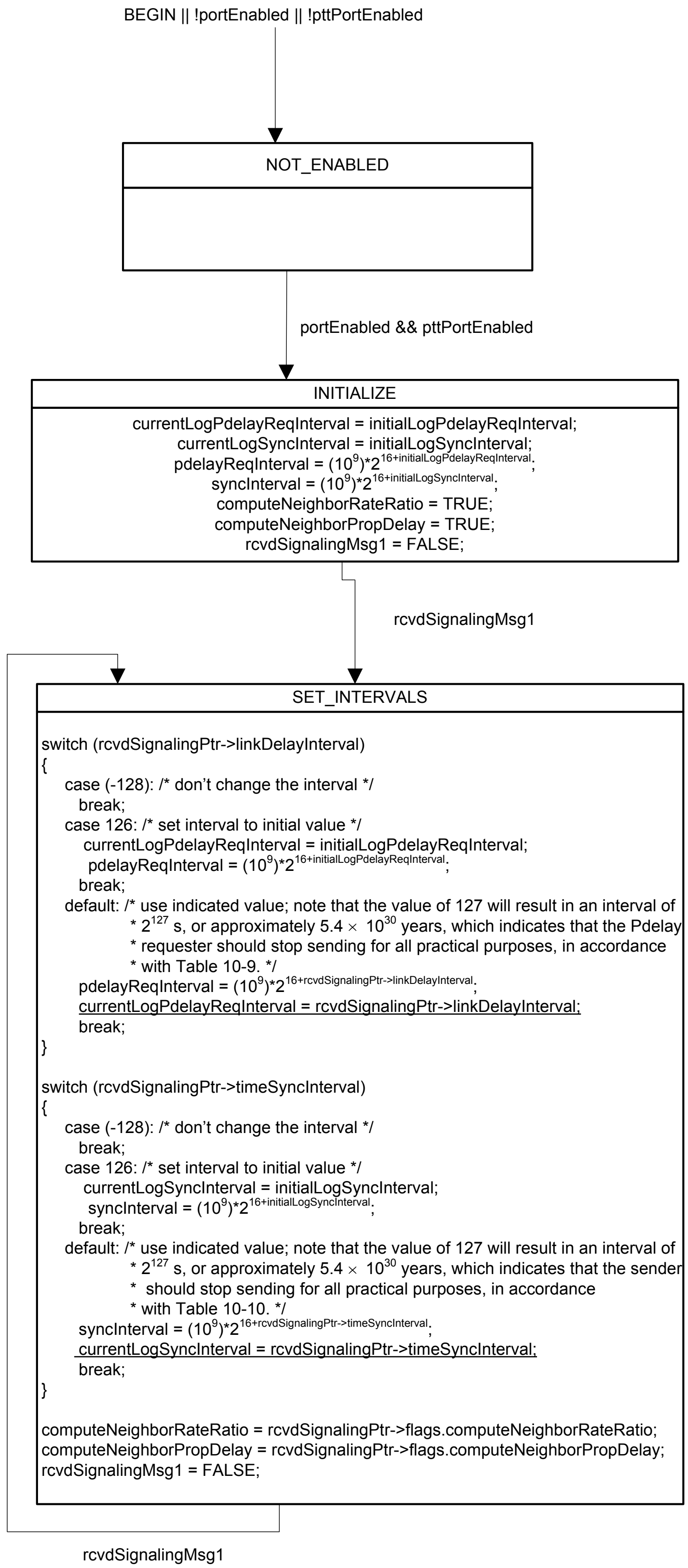
The ratio of the grandmaster frequency to the frequency of the LocalClock entity at time-aware system i, rateRatio$i$, is equal to the same quantity at time-aware system i–1, $\text{rateRatio}_{i-}1$, multiplied by the ratio of the frequency of the LocalClock entity at time-aware system i–1 to the frequency 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 $\text{rateRatio}_{i-}1$ 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 = $(10^9)*2^{16+\text{initialLogPdelayReqInterval}}$;
syncInterval = $(10^9)*2^{16+\text{initialLogSyncInterval}}$;
computeNeighborRateRatio = TRUE;
computeNeighborPropDelay = TRUE;
rcvdSignalingMsg1 = FALSE;
rcvdSignalingMsg1
SET_INTERVALS
switch (rcvdSignalingPtr->linkDelayInterval)
{
case (-128): /* don't change the interval */
break;
case 126: /* set interval to initial value */
currentLogPdelayReqInterval = initialLogPdelayReqInterval;
pdelayReqInterval = $(10^9)*2^{16+\text{initialLogPdelayReqInterval}}$;
break;
default: /* use indicated value; note that the value of 127 will result in an interval of
* 2^{127} s, or approximately 5.4×10^{30} years, which indicates that the Pdelay
* requester should stop sending for all practical purposes, in accordance
* with Table 10-9. */
pdelayReqInterval = $(10^9)*2^{16+\text{rcvdSignalingPtr->linkDelayInterval}}$;
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 = $(10^9)*2^{16+\text{initialLogSyncInterval}}$;
break;
default: /* use indicated value; note that the value of 127 will result in an interval of
* 2^{127} s, or approximately 5.4×10^{30} years, which indicates that the sender
* should stop sending for all practical purposes, in accordance
* with Table 10-10. */
syncInterval = $(10^9)*2^{16+\text{rcvdSignalingPtr->timeSyncInterval}}$;
currentLogSyncInterval = rcvdSignalingPtr->timeSyncInterval;
break;
}
computeNeighborRateRatio = rcvdSignalingPtr->flags.computeNeighborRateRatio;
computeNeighborPropDelay = rcvdSignalingPtr->flags.computeNeighborPropDelay;
rcvdSignalingMsg1 = FALSE;
rcvdSignalingMsg1

15. Managed object definitions

Change the Text of 15.5, ieee8021AsTimeSyncMib as shown

```
ieee8021AsTimeSyncMib MODULE-IDENTITY
    LAST-UPDATED "~~201011110000Z~~201105100000Z" -- ~~November 11, 2010~~May 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 "~~201011110000Z~~201105100000Z" -- ~~November 11, 2010~~May 10, 2011

    DESCRIPTION
            "Published as part of IEEE Std 802.1AS

             Copyright (C) IEEE (2011)."
```

Change the Text of 15.5, ieee8021AsParentDSGrandmasterPriority1 as shown

```
ieee8021AsParentDSGrandmasterPriority1 OBJECT-TYPE
    SYNTAX      Unsigned32(0..255)
    MAX-ACCESS  read-~~write~~only
    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

```
ieee8021AsParentDSGrandmasterPriority2 OBJECT-TYPE
```

```
    SYNTAX      Unsigned32(0..255)
    MAX-ACCESS  read-~~write~~only
    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~~2..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
                syncReceiptTimeout 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(02..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 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
                syncReceiptTimeout sync intervals is referred to as
                'sync receipt timeout'.

                The initialziationinitialization value is 3 (see 10.6.3.1).

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