148. PLCA Reconciliation Sublayer (RS)

148.1 Introduction

This clause specifies a Reconciliation Sublayer to provide optional Physical Layer Collision Avoidance (PLCA) capabilities among participating stations. The PLCA RS is specified for operation with Clause 147 (10BASE-T1S) PHYs operating in half-duplex multidrop mode. PLCA can be dynamically enabled or disabled via management interface. When PLCA is disabled, the Reconciliation Sublayer mapping is identical to that specified in Clause 22.

When enabled, the PLCA RS aligns data from the MAC with transmission opportunities of the Physical Layer and maps the Physical Layer signals to PLS primitives towards the MAC. The use of PLCA-enabled Physical Layers in CSMA/CD half-duplex shared-medium networks can provide enhanced bandwidth and improved access latency under heavily loaded traffic conditions. PLCA-enabled nodes can coexist with nodes without PLCA enabled on the same mixing segment, all using IEEE 802.3 CSMA/CD.

148.1.1 Conventions in this clause

The body of this clause contains state diagrams, including definitions of variables, constants, and functions. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails.

148.1.1.1 State diagram notation

The conventions of 21.5 are adopted with the extension that some states in the state diagrams use an IF-THEN-ELSE-END construct to condition which actions are taken within the state. If the logical expression associated with the IF evaluates TRUE, all the actions listed between THEN and ELSE will be executed. In the case where ELSE is omitted, the actions listed between THEN and END will be executed. If the logical expression associated with the IF evaluates FALSE, the actions listed between ELSE and END will be executed. After executing the actions listed between THEN and ELSE, between THEN and END, or between ELSE and END, the actions following the END, if any, will be executed.

148.1.1.2 State diagram timer specifications

All timers operate in the manner described in 40.4.5.2.

148.1.1.3 Service specifications

The method and notation used in the service specification follows the conventions of 1.2.2.

148.2 Overview

The working principle of PLCA is that transmit opportunities on a mixing segment are granted in sequence based on a node ID unique to the local collision domain (set by the management entity). The method of determination of the node ID and to_timer by the management entity is beyond the scope of this standard. Proper operation of the Clause 148 functionality assumes that the assigned node ID is unique in the local collision domain.

The node ID assignment value does not appear externally or in the payload packet format. The node ID assignment value is fully contained within the local collision domain.

Transmit opportunities are generated in a round-robin fashion. The node with ID = 0 signals a BEACON on the medium. Reception of a BEACON indicates the start of a new cycle of transmit opportunities. If the node with ID = 0 fails, the network is still operational with the same performance level of a CSMA/CD network without PLCA.

Each node is allowed to transmit a single packet during its own transmit opportunity. Individual nodes can be enabled to transmit a number of additional packets, up to the configured limit, within the same transmit opportunity.

PLCA relies on the PLS_SIGNAL.indication and PLS_CARRIER.indication primitives to have the MAC delay transmission until a transmit opportunity is available.

PLCA-enabled nodes may be used in the same CSMA/CD collision domain as non-PLCA enabled nodes. As the percentage of non-PLCA enabled nodes increases, performance advantages also decrease. If the node with ID = 0 fails, the network is still operational with the same performance level of a CSMA/CD network without PLCA.

EDITORIAL NOTE: add overview text for Dynamic PLCA (D-PLCA)

148.3 Relationship with other IEEE standards

The relationship between the PLCA Reconciliation Sublayer, the ISO Open Systems Interconnection (OSI) Reference Model, and the IEEE 802.3 Ethernet model is shown in Figure 148–1. The Reconciliation Sublayer (shown shaded) in Figure 148–1 connects one Clause 4 Media Access Control (MAC) layer to the PHY. MII is defined in Clause 22.

148.4 PLCA Reconciliation Sublayer operation

148.4.1 General

This subclause specifies services provided by the PLCA RS as an extension to the RS specified in Clause 22. Figure 148–2 depicts the RS interlayer service interfaces. The PLCA RS contains the Control and Data state diagrams, the variable delay line, and command detect logic. When PLCA functions are not supported or are disabled by the management interface (plca_en = FALSE), RS operation shall conform to the RS definition in Clause 22.

148.4.2 Mapping of MII signals to PLS service primitives and PLCA functions

The RS maps the signals provided at the MII to the PLS service primitives defined in Clause 6 via the PLCA state diagrams, variables, and functions (see 148.4.4 and 148.4.5). The PLS service primitives provided by the RS behave in exactly the same manner as defined in Clause 6.

148.4.2.1 Mapping of PLS_DATA.request

When PLCA is disabled (plca_en = FALSE), the mapping of the PLS_DATA.request primitive shall be the one specified in 22.2.1.1. Otherwise, the following applies.

148.4.2.1.1 Function

Maps the primitive PLS_DATA.request to PLCA variables which in turn generate the MII signals TXD<3:0> and TX_EN.

148.4.2.1.2 Semantic of the service primitive

PLS_DATA.request (OUTPUT_UNIT)

The OUTPUT_UNIT parameter can take one of three values: ONE, ZERO, or DATA_COMPLETE. It represents a single data bit. The values ONE and ZERO are conveyed by the individual bits of the four-bit variable plca_txd<3:0>. Each bit of plca_txd<3:0> conveys one bit of data while plca_txen is set to TRUE. The value DATA_COMPLETE is conveyed by setting the variable plca_txen to FALSE. MII signals TXD<3:0> and TX_EN are generated by way of the PLCA Data state diagrams specified in 148.4.5. Synchronization between the RS and the PHY is achieved by way of the TX_CLK signal.

148.4.2.1.3 When generated

The plca_txd<3:0> and plca_txen variables are assigned after every group of four PLS_DATA.request transactions from the MAC sublayer to request the PLCA functions to transmit a nibble of data when the transmit opportunity is available, or to signal the end of the transmission. The TX_CLK signal is generated by the PHY. The TXD<3:0> and TX_EN signals are generated by the RS according to PLCA Data state diagrams (see 148.4.5).

148.4.2.2 Mapping of PLS_DATA.indication

Map of the primitive PLS_DATA.indication shall comply with 22.2.1.2.

148.4.2.3 Mapping of PLS_CARRIER.indication

When PLCA is disabled (plca_en = FALSE), the mapping of the PLS_CARRIER.indication primitive shall be the one specified in 22.2.1.3. Otherwise, the following applies

148.4.2.3.1 Function

Maps the primitive PLS_CARRIER.indication to the PLCA Data state diagram.

148.4.2.3.2 Semantic of the service primitive

PLS_CARRIER.indication (CARRIER_STATUS)

The CARRIER_STATUS parameter can take one of two values: CARRIER_ON or CARRIER_OFF.

148.4.2.3.3 When generated

The PLS_CARRIER.indication service primitive is generated by the RS according to the PLCA Data state diagram specified in 148.4.5.

148.4.2.4 Mapping of PLS_SIGNAL.indication

When PLCA is disabled (plca_en = FALSE) the mapping of the PLS_SIGNAL.indication primitive shall be the one specified in 22.2.1.4. Otherwise, the following applies.

148.4.2.4.1 Function

Map the primitive PLS_SIGNAL.indication to the PLCA Data state diagram.

148.4.2.4.2 Semantic of the service primitive

PLS_SIGNAL.indication (SIGNAL_STATUS)

The SIGNAL_STATUS parameter can take one of two values: SIGNAL_ERROR or NO_SIGNAL_ERROR.

148.4.2.4.3 When generated

SIGNAL_STATUS is generated by the PLCA Data state diagram specified in 148.4.5.

148.4.2.5 Mapping of PLS_DATA_VALID.indication

Map of the primitive PLS_DATA_VALID.indication shall comply with 22.2.1.7.

148.4.2.6 Generation of TX_ER

Generation of TX_ER shall comply with the PLCA Data state diagram specified in 148.4.5.1.

148.4.2.7 Response to RX_ER indication

Response to RX_ER indication from the MII shall comply with 22.2.1.5.

148.4.3 Requirements for the PHY

In order to support PLCA, the RS has to be connected to a 10BASE-T1S PHY.

148.4.3.1 PHY response to PLCA commands and notifications

148.4.3.1.1 BEACON request

The BEACON function is specified in 148.4.4.1.

The RS conveys the BEACON request via MII interface.

Upon the reception of this request, the PHY encodes and transmits a signal communicating the BEACON to other PHYs on the segment so that they generate a BEACON indication.

Upon the reception of this request, the RX_DV signal is not asserted.

PHYs may map the BEACON request to any suitable line coding as long as the requirements defined in this subclause are met.

148.4.3.1.2 COMMIT request

The COMMIT function is specified in 148.4.4.1.

The PLCA Control state diagram generates a COMMIT request by way of the tx_cmd variable as specified in 148.4.4.2. The RS conveys such request via MII interface as defined in 22.2.2.4.

Upon the reception of this request, the RX_DV signal is not asserted.

PHYs may map the COMMIT request to any suitable line coding as long as the requirement defined in this subclause are met.

148.4.3.2 Mapping of MII signals to PLCA variables

The PLCA RS is required to decode PLCA-specific signaling out of the MII.

148.4.3.2.1 BEACON indication

When the PHY receives a BEACON, it indicates this information to the RS by asserting MII signals.

The RS shall react to such indication by setting the PLCA variable rx_cmd to the value BEACON. The RS shall also reset the rx_cmd variable to NONE when the BEACON indication on the MII ceases, unless a COMMIT indication is signaled, in which case rx_cmd shall be set as specified in 148.4.3.2.2.

148.4.3.2.2 COMMIT indication

When the PHY receives a COMMIT from the line, it indicates this information to the RS by asserting MII signals. The PHY asserts CRS when a COMMIT indication is detected.

The RS shall react to such indication by setting the PLCA variable rx_cmd to the value COMMIT. The RS shall also reset the rx_cmd variable to NONE when the COMMIT indication on the MII ceases, unless a BEACON indication is signaled, in which case rx_cmd shall be set as specified in 148.4.3.2.1.

148.4.4 PLCA Control

148.4.4.1 PLCA Control state diagram

The PLCA Control function shall conform to the PLCA Control state diagram in Figure 148–3 and Figure 148–4 and associated state variables, functions, timers, and messages.

To achieve error free operation the PLCA node should be configured appropriately before transmit functions are enabled. Appropriate configuration includes the following:

a) Each local_nodeID is unique to the local collision domain.
b) There is one and only one node with local_nodeID = 0 on the local collision domain.
c) The transmit opportunity timer (to_timer) is set equal across all the nodes on the local collision domain.
d) plca_node_count is set on the node with local_nodeID = 0 to the maximum number of nodes supported on the local collision domain.

When PLCA functions are enabled and local_nodeID equals zero, PLCA switches to RECOVER state and waits one cycle of transmit opportunities. This prevents sending a BEACON at an inappropriate time (e.g., when the node with local_nodeID = 0 resets in the middle of a cycle of transmit opportunities and other nodes could still be sending valid data). A BEACON is then generated by such node by switching to SEND_BEACON state. On reception of a BEACON, all other nodes reset their own transmit opportunity counter and related timer.

When PLCA functions are enabled, nodes with nonzero local_nodeID wait in RESYNC state until a BEACON is received.

All nodes, including the one generating the BEACON, detect the end of the BEACON condition before proceeding to WAIT_TO state, in order to minimize latency differences across the network.

Entering WAIT_TO state, the node waits for one of these possible conditions:

1) CRS is asserted by the PHY through MII, indicating there is activity on the line.
2) curID becomes equal to local_nodeID while packetPending variable is TRUE, meaning that this node now owns a transmit opportunity and does have a packet to transmit.
3) curID becomes equal to local_nodeID while packetPending variable is FALSE, meaning that this node now owns a transmit opportunity but does not have a packet to transmit.
4) to_timer elapses, indicating the current transmit opportunity is yielded.

If condition (1) occurs, the node is about to receive either a valid packet, a COMMIT request, a BEACON request or it might be receiving a false carrier event.

In EARLY_RECEIVE state, the PLCA Control state diagram is waiting for the PHY to properly decode the incoming signal and to take the following actions:

- Switch to RECEIVE state if a COMMIT indication is reported or a valid packet is being decoded. The PLCA Control state diagram then remains in the RECEIVE state until the line is free (CRS deasserted).
- Switch to RESYNC state if a BEACON is received with local_nodeID ≠ 0, which starts a new cycle of transmit opportunities.
- Switch to RESYNC state if CRS is not followed by the reception of a packet and local_nodeID ≠ 0, meaning that a false carrier occurred and the curID variable might be out of synchronization. In this case, the node skips its transmit opportunity (TO) and waits for a new BEACON in order not to disrupt the current cycle of transmit opportunities.
- Switch to RECOVER state if local_nodeID is 0 and CRS is de-asserted but no packet is being received. In RECOVER state, since the curID variable might be out of synchronization, this node waits for the end of the current cycle of transmit opportunities before sending a new BEACON. This is required so as not to send a BEACON while other nodes might still be using their TO.

When condition (2) occurs, the node now gets a TO having at least one packet to be transmitted. COMMIT state is then entered to signal other nodes to stop their to_timer and wait for a packet by the means of a COMMIT request. COMMIT state is left once the data to be transmitted is available from the MAC or the PLCA delay line.

When condition (3) occurs, the node now gets a TO without being ready to send any packet. In this case, the YIELD state is entered to skip the TO, allowing other nodes a chance to transmit. In some rare cases (e.g., a non-PLCA enabled node is connected to the network) it is possible to receive data in YIELD state. If this unlikely event happens, PLCA switches to RECEIVE state to wait until the end of the transmission and increment curID properly.

When condition (4) is met, another node has yielded its transmit opportunity, causing the curID variable to be incremented and to_timer to be reset.

148.4.4.2 PLCA Control variables

plca_reset

The plca_reset signal is used to reset the optional PLCA function in the RS. This signal maps to TRUE when acPLCAReset is in reset and to FALSE when acPLCAReset is normal, but is further qualified.
Values: TRUE or FALSE

plca_en

The plca_en signal controls the optional PLCA function in the RS. This signal maps to TRUE when aPLCAAdminState is enabled and to FALSE when aPLCAAdminState is disabled.
Values: TRUE or FALSE

CRS

The MII signal CRS.
Values: TRUE or FALSE

RX_DV

The MII signal RX_DV.
Values: TRUE or FALSE

receiving

Defined as: (RX_DV = TRUE) + (rx_cmd = COMMIT).
Values: TRUE or FALSE

tx_cmd

Command for the PLCA Data state diagram to convey to the PHY via the MII.
Values: NONE, BEACON or COMMIT

rx_cmd

Encoding present on RXD<3:0>, RX_ER, and RX_DV as defined in Table 22–2.
Values:
BEACON: PLCA BEACON indication encoding present on RXD<3:0>, RX_ER, and RX_DV
COMMIT: PLCA COMMIT indication encoding present on RXD<3:0>, RX_ER, and RX_DV
NONE: PLCA BEACON or COMMIT indication encoding not present on RXD<3:0>, RX_ER, and RX_DV

TX_EN

The MII signal TX_EN.
Values: TRUE or FALSE

local_nodeID

ID representing the PLCA transmit opportunity number assigned to the node. This signal maps to aPLCALocalNodeID.
Values: integer value from 0 to 255

plca_node_count

Maximum number of PLCA nodes on the mixing segment receiving transmit opportunities before the node with local_nodeID = 0 generates a new BEACON, reflecting the value of aPLCANodeCount. This parameter is meaningful only for the node with local_nodeID = 0; otherwise, it is ignored.
Values: integer number from 0 to 255

committed

Internal variable used to synchronize PLCA Control and Data functions.
It is set by PLCA Control state diagram to signal that the current transmit opportunity has been committed and the PLCA Data state diagram is now allowed to convey MII data to the PHY.
Values: TRUE or FALSE

packetPending

Internal variable used to synchronize PLCA Control and Data functions.
The PLCA Data state diagram sets this variable when it detects the MAC is ready to send a packet and have PLCA Control state diagram actually commit for the next available transmit opportunity.
Values: TRUE or FALSE

bc

Counts the number of additional packets currently sent in a burst after the first transmission.
Values: integer from 0 to 255

max_bc

Maximum number of additional packets the node is allowed to transmit in a single burst. This signal maps to aPLCAMaxBurstCount attribute.
Values: integer from 0 to 255

plca_active

Notifies the PLCA Status function whether the node is waiting for sending or receiving a BEACON or it already sent or received one.
Values: TRUE or FALSE

curID

Integer variable tracking the ID of the node that currently owns a transmit opportunity.
Values: integer from 0 to 255

PMCD

Prescient mii_clock_done. This variable is set false on entry to the RSYNC state and becomes TRUE 1

± ½ bit time prior to mii_clock_done becoming TRUE.
Values: TRUE or FALSE

dplca_en
The dplca_en signal controls the optional D-PLCA function. This signal maps to TRUE when aDPLCAAdminState is enabled and to FALSE when aDPLCAAdminState is disabled.
Values: TRUE or FALSE

dplca_txop_end
Notifies the D-PLCA state diagrams that the transmit opportunity indicated by the dplca_txop_id variable is over.
Values: TRUE or FALSE

dplca_txop_id
Copy of the curID variable, synchronized with dplca_txop_end.
Values: integer from 0 to 255

dplca_txop_claim
Notifies the D-PLCA state diagrams whether the transmit opportunity indicated by dplca_txop_id was claimed by a node. Additionally, it specifies the type of claim:
- SOFT, meaning that a packet not including a COMMIT indication was received. SOFT claims may be issued implicitly by nodes not supporting D-PLCA.
- HARD, meaning that a packet including a COMMIT indication was received. HARD claims may be issued by D-PLCA enabled nodes, and occasionally by statically configured PLCA enabled nodes.
Values: NONE, SOFT or HARD

148.4.4.3 Functions

No functions are defined for the PLCA Control state diagram.

148.4.4.4 Timers

beacon_timer
Times the duration of the BEACON signal.
Duration: 20 bit times.
Tolerance: ± ½ bit time.

beacon_det_timer
Timer for detecting received BEACONs.
Duration: 22 bit times.
Tolerance: ± 1 bit time.

invalid_beacon_timer
Timer used for BEACON validation. This timer is stopped any time rx_cmd = BEACON.
Duration: 4000 ns.
Tolerance: ± 400 ns.

burst_timer
This timer determines how long to wait for the MAC to send a new packet before yielding the transmit opportunity. For PLCA burst mode to work properly this timer should be set greater than one IPG. Timer duration maps to aPLCABurstTimer attribute.
Duration: integer number between 0 and 255, expressed in bit times.
Tolerance: ± ½ bit time.
The default value is specified in 30.16.1.1.7.

to_timer

The transmit opportunity timer maps to aPLCATransmitOpportunityTimer. The timer value needs to meet Equation (148–1). The to_timer should be set equal across the mixing segment for PLCA to work properly.

(148–1)

where $t_{propdelay}$ is the propagation delay between any two nodes on the mixing segment, and the delay specifications are the maxima and minima for the PHY type on the mixing segment (for 10BASE-T1S, see 147.11).

Duration: integer number between 1 and 255, expressed in bit times.
Tolerance: 100 ppm.
The default value is specified in 30.16.1.1.5.

append_commit_timer

Timer used by D-PLCA to append a COMMIT to each transmitted packet.
Duration: 22 bit times.
Tolerance: ± 1 bit time.

148.4.4.5 Abbreviations

MCD See 148.4.5.5

148.4.4.6 State diagram

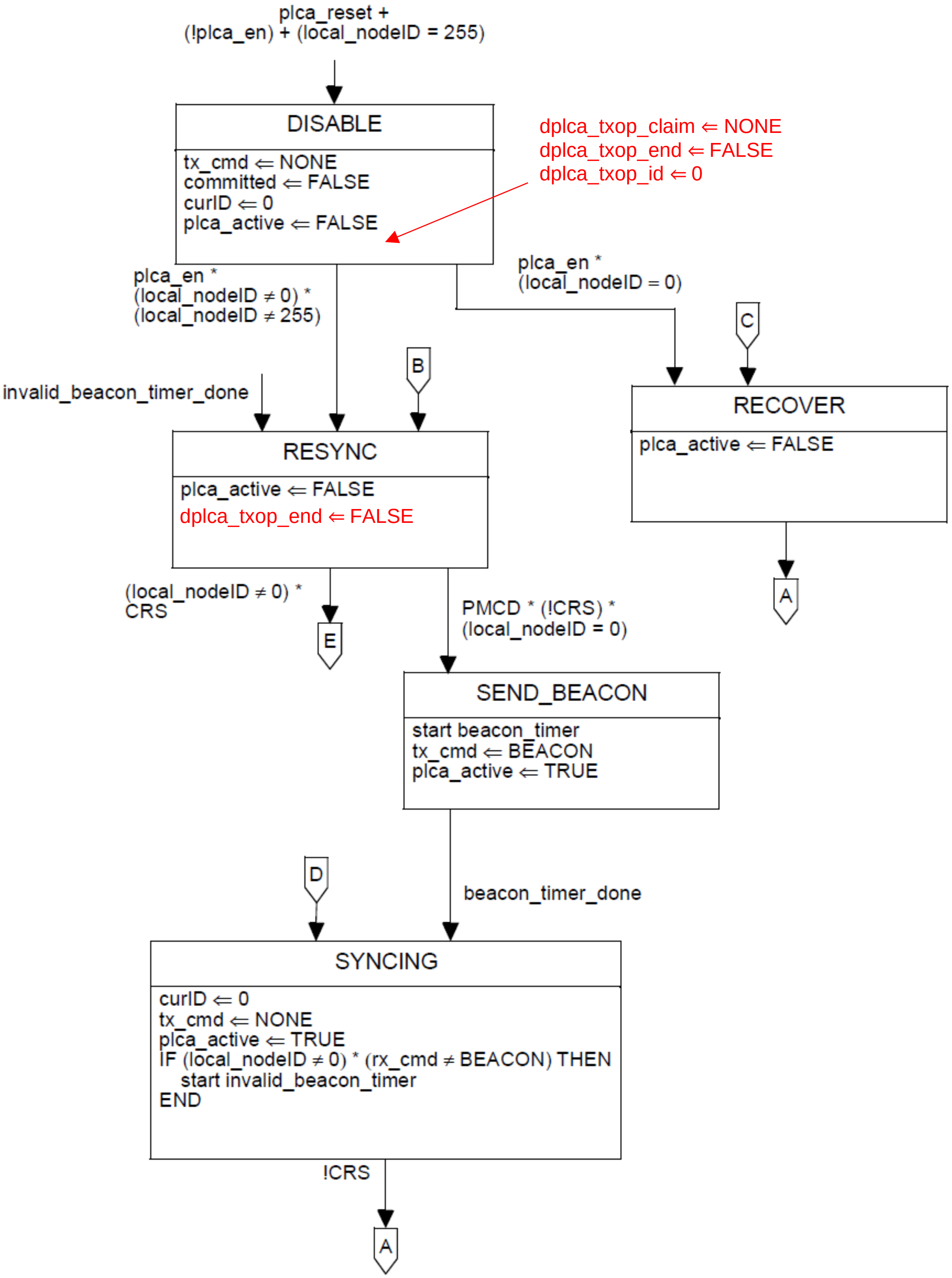


Figure 148–3—PLCA Control state diagram, part a

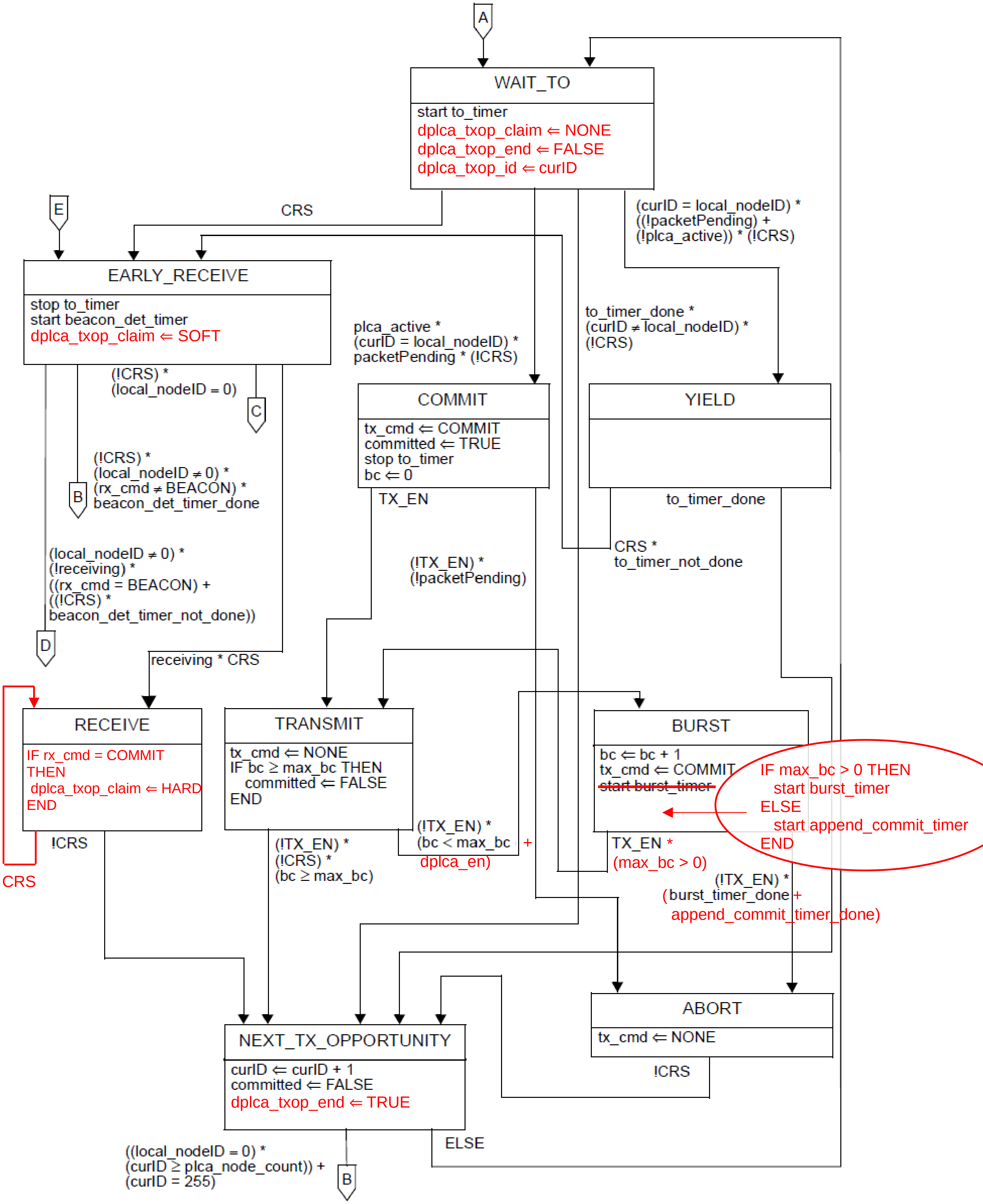


Figure 148–4—PLCA Control state diagram, part b

148.4.5 PLCA Data

148.4.5.1 PLCA Data state diagram

The PLCA Data state diagram is responsible for detecting when the MAC is ready to send a packet and delaying the transmission until a transmit opportunity is detected.

The PLCA Data function shall conform to the PLCA Data state diagram in Figure 148–5 and Figure 148–6 and associated state variables, functions, timers, and messages.

When PLCA functions are enabled, the PLCA Data state diagram transitions to the IDLE state and waits for the MAC to start a transmission or the PHY to assert carrier sense. In the former case, the data conveyed by the MAC through the PLS_DATA.request primitive is delayed by switching to HOLD state. In the latter case, CARRIER_ON is signaled through the PLS_CARRIER.indication to have the MAC defer any new transmission, then the RECEIVE state is entered.

The MAC however, might have started a transmission right before a carrier is detected. In this case, the Data state diagram switches to the COLLIDE state asserting SIGNAL_STATUS = SIGNAL_ERROR via PLS_SIGNAL.indication primitive to have the MAC perform a backoff and send the packet again later, without actually forwarding any data for the PHY to transmit on the medium.

During the HOLD state, the PLCA Control state diagram is notified via the packetPending variable that data is available to be transmitted and the beginning of the transmission is held in the variable delay line. At the next transmit opportunity, the PLCA Control state diagram allows transmitting the delayed data by setting the committed variable to TRUE. In such a case, the PLCA Data state diagram switches to TRANSMIT state to actually deliver the data for the PHY to encode and transmit on the medium.

The variable delay line is a small buffer that aligns a transmission with the transmit opportunity.

If plca_txer is asserted during the HOLD state, the PLCA Data state diagram switches to ABORT state to assert packetPending = FALSE and to wait until the MAC stops sending data. The aborted packet will not be transmitted on the medium.

If another node starts a transmission during the HOLD state, the delayed data is dropped and a collision is triggered by switching to COLLIDE state.

During the COLLIDE state, packetPending = FALSE and CARRIER_STATUS = CARRIER_ON are asserted via the PLS_CARRIER.indication primitive. When the MAC has completed sending the jam bits as described in Clause 4, the PLCA Data state diagram waits for the next transmit opportunity by switching to DELAY_PENDING state. The PLCA Data state diagram transitions to the PENDING state after waiting for the pending_timer. The pending_timer is used to prevent committing to a transmit opportunity before transmit data is available. This prevents conveying unwanted long COMMIT requests to the PHY.

During the PENDING state, the PLCA Data state diagram asserts packetPending = TRUE and keeps CARRIER_STATUS = CARRIER_ON via the PLS_CARRIER.indication primitive to prevent the MAC from making new transmit attempts until the PLCA Control state diagram signals that a new transmit opportunity is available. At that point, CARRIER_STATUS is set to CARRIER_OFF to have the MAC resend data after waiting one IPG period as described in Clause 4.

148.4.5.2 Variables

a

Current delay counter.

b

Flush counter.

CARRIER_STATUS

See 148.4.2.3.2.

COL

The MII signal COL specified in 22.2.2.12.

committed

See 148.4.4.2.

CRS

The MII signal CRS (see 22.2.2.11).

packetPending

See 148.4.4.2.

plca_en

See 148.4.4.2.

plca_reset

See 148.4.4.2.

plca_status

See 148.4.6.2.

plca_txd<3:0>

A four-bit data value conveying a nibble of data to transmit from four successive PLS_DATA.request(OUTPUT_UNIT) primitives where OUTPUT_UNIT has a value of ONE or ZERO. See 148.4.2.1.2. The addition of a subscript 'n-a', i.e., $\text{plca\_txd}_{n\text{-}a}$ indicates the plca_txd conveyed ‘a’ mii_clock_timer expirations before the most recent one.

plca_txen

See 148.4.2.1.2.

plca_txer

The conditions for generating plca_txer are the same as defined in 22.2.1.6 and 22.2.2.5 for the TX_ER MII signal.
Values: TRUE or FALSE

receiving

See 148.4.4.2.

rx_cmd

See 148.4.4.2.

SIGNAL_STATUS

See 148.4.2.4.2

tx_cmd

See 148.4.4.2.

tx_cmd_sync

The value of the tx_cmd variable sampled on the rising edge of the MII TX_CLK.
Values: see tx_cmd in 148.4.4.2

TXD

The MII signals TXD<3:0> specified in 22.2.2.4.

TX_EN

The MII signal TX_EN specified in 22.2.2.3.

TX_ER

The MII signal TX_ER specified in 22.2.2.5.

148.4.5.3 Functions

ENCODE_TXER

This function takes as its argument the tx_cmd_sync variable defined in 148.4.5.2. It returns TRUE if tx_cmd_sync is BEACON or COMMIT. Otherwise, it returns the value of the plca_txer variable, defined in 148.4.5.2.

ENCODE_TXD

This function takes as its argument the tx_cmd_sync variable defined in 148.4.5.2.
If tx_cmd_sync is BEACON, the return value is the TXD encoding defined in Table 22–1 for the BEACON request.
If tx_cmd_sync is COMMIT, the return value is the TXD encoding defined in Table 22–1 for the COMMIT request.
Otherwise, the return value is 0000.

148.4.5.4 Timers

commit_timer

Defines the maximum time the PLCA Data state machine is allowed to stay in WAIT_MAC state.
Duration: 288 bit times.
Tolerance: ± ½ bit time.

mii_clock_timer

A continuous free-running timer that shall expire synchronously with the rising edge of the MII TX_CLK.

Restart time: Immediately after expiration; restarting the timer resets the condition mii_clock_timer_done.
Duration: see 22.2.2.1.

pending_timer

Defines the time the PLCA Data state diagram waits in the DELAY_PENDING state before switching to PENDING state.
Duration: 512 bit times.
Tolerance: ± ½ bit time.

148.4.5.5 Abbreviations

MCD Alias for mii_clock_timer_done

148.4.5.6 Constants

delay_line_length

This constant is implementation dependent and specifies the maximum number of nibbles that the PLCA RS variable delay line can hold.
Value: up to 99

148.4.5.7 State diagram

148.4.6 PLCA Status

148.4.6.1 PLCA Status state diagram

The PLCA Status state diagram is responsible for reporting whether nodes are actively sending/receiving the BEACON. The PLCA Status function shall conform to the PLCA Status state diagram in Figure 148–7 and associated state variables, functions, timers, and messages.

Upon reset or when PLCA is disabled, the PLCA Status function enters the INACTIVE state and reports plca_status as FAIL. As soon as the PLCA Control function enters the SYNCING state (i.e., receiving or transmitting the BEACON), the plca_active variable is set to TRUE and PLCA Status switches to the ACTIVE state, reporting plca_status as OK.

From the ACTIVE state, whenever plca_active is set to FALSE by the PLCA Control function, the PLCA Status function enters the HYSTERESIS state, still reporting plca_status as OK and arming plca_status_timer.

If plca_active is reset to TRUE, then PLCA Status reverts to the ACTIVE state, effectively filtering the momentarily inactive state. Instead, if plca_status_timer expires while plca_active is still FALSE, the PLCA Status function reverts to the INACTIVE state, reporting plca_status as FAIL.

148.4.6.2 PLCA Status variables

plca_status

If plca_status is OK, BEACONs are being received or transmitted, and the PLCA Control state diagram is in normal operation. If plca_status is FAIL, the PLCA Control state diagram has been in the DISABLE, RESYNC, or RECOVER state for greater than the duration of the plca_status timer. This signal maps to aPLCAStatus attribute as specified in 30.16.1.1.2.
Values: OK or FAIL

plca_active

See 148.4.4.2.

plca_en

See 148.4.4.2.

plca_reset

See 148.4.4.2.

148.4.6.3 Functions

No functions are defined for PLCA Status state diagram.

148.4.6.4 Timers

plca_status_timer

Represents the time plca_status is maintained in OK state when plca_active is FALSE while in the HYSTERESIS state.
Duration: the duration of this timer is 130 090 bit times, which is 2 × (max to_timer × max plca_node_count + beacon_timer).
Tolerance: timer may expire up to 10 000 BT (nominally 1 ms at 10 Mb/s) greater than the specified duration

148.4.6.5 State diagram

148.4.7 D-PLCA

148.4.7.1 D-PLCA State diagram overview

EDITORIAL NOTE: descriptive text for D-PLCA is required here

148.4.7.2 Variables

plca_en
See 148.4.4.2.

dplca_en
See 148.4.4.2.

plca_reset
See 148.4.4.2

plca_status
See 148.4.6.2.

short_cnt
Counter of BEACON cycles for the short aging time (SOFT claims)
Values: positive integer number

long_cnt
Counter of BEACON cycles for the long aging time (HARD claims)
Values: positive integer number

dplca_aging
This variable controls the state of the D-PLCA aging state diagram.
Values: ON or OFF

coordinator_role_allowed
This variable controls whether the local node is allowed to take the coordinator role (local_nodeID = 0) during the D-PLCA node assignment procedure. This variable maps on the TBD attribute in Clause 30.x.
Values: TRUE or FALSE

txop_claim_table
This variable contains the claim state of the 256 transmit opportunities IDs. The claim state of each ID can be:

a. NONE, meaning that the transmit opportunity ID is available to be returned by the pick_free_txop function
b. SOFT, meaning the ID is currently claimed by a node transmission that did not include a COMMIT indication.
c. HARD, meaning the ID is currently claimed by a node transmission that included a COMMIT indication at the beginning or at the end of the carrier event.

The transmit opportunity table is maintained by the D-PLCA aging state diagram defined in TBD.
Values: Array of 256 elements, each having a value of NONE, SOFT or HARD.

txop_claim_table_new
Copy of txop_claim_table use by the D-PLCA Aging State Diagram to handle the expiration of HARD claims.
Values: same as txop_claim_table

local_nodeID

See 148.4.4.2.

plca_node_count

See 148.4.4.2.

rx_cmd

See 148.4.4.2.

dplca_txop_claim

See 148.4.4.2.

dplca_txop_end

See 148.4.4.2.

dplca_txop_id

See 148.4.4.2.

dplca_txop_table_upd

Synchronization variable set by the D-PLCA aging state diagram to notify the D-PLCA control state diagram that the table of transmit opportunity has been updated.
Values: TRUE or FALSE

HARD_AGING_CYCLES

Defines the number of BEACON cycles before the HARD claims over the transmit opportunities expire.
This variable maps to the TBD attribute define in Clause 30.x
Values: positive integer number

SOFT_AGING_CYCLES

Defines the number of BEACON cycles before the SOFT claims over the transmit opportunities expire.
This variable maps to the TBD attribute define in Clause 30.x
Values: positive integer number

148.4.7.3 Functions

max_hard_claim

This function takes as parameter the txop_claim_table defined in TBD.
It returns the highest ID in the table which is marked as HARD claimed. Note that the ID claimed by the local node does not count as claimed.

pick_free_txop

This function takes as parameter the txop_claim_table defined in TBD.
It returns any ID that is not marked as HARD or SOFT claimed in the table, with the following exceptions:
a. it shall not return zero, which is reserved for the PLCA coordinator
b. it shall not return an ID greater than the highest HARD claimed in the table, unless this is the only one available.

Note that it is allowed for this function to return the ID currently being claimed by the local node, unless it is claimed by another node.

The actual criteria for choosing among the available, allowed IDs is implementation defined.

hard_claiming

This function takes as parameter “ID”, a transmit opportunity integer number in the range of 0 to 255. It

returns the result of the following boolean expression:
dplca_txop_end * (dplca_txop_claim = HARD) * (dplca_txop_id = ID)

soft_claiming

This function takes as parameter "ID", a transmit opportunity integer number in the range of 0 to 255. It returns the result of the following boolean expression:
dplca_txop_end * (dplca_txop_claim = SOFT) * (dplca_txop_id = ID)

clear_txop_table

This function takes as an argument either the txop_claim_table or the txop_claim_table_new variable. When invoked, it sets all of the 256 elements of the specified table to the NONE claim state.

clear_soft_claims

This function takes as an argument either the txop_claim_table or the txop_claim_table_new variable. When invoked, it reverts all of the array elements that have been marked as SOFT claims to NONE.

148.4.7.4 Timers

wait_beacon_timer

Represents the time the D-PLCA state diagram waits for a BEACON indication.
Duration: the duration of this timer is defined by the TBD configuration parameter.
Tolerance: 1 BT

148.4.7.5 Control State diagram

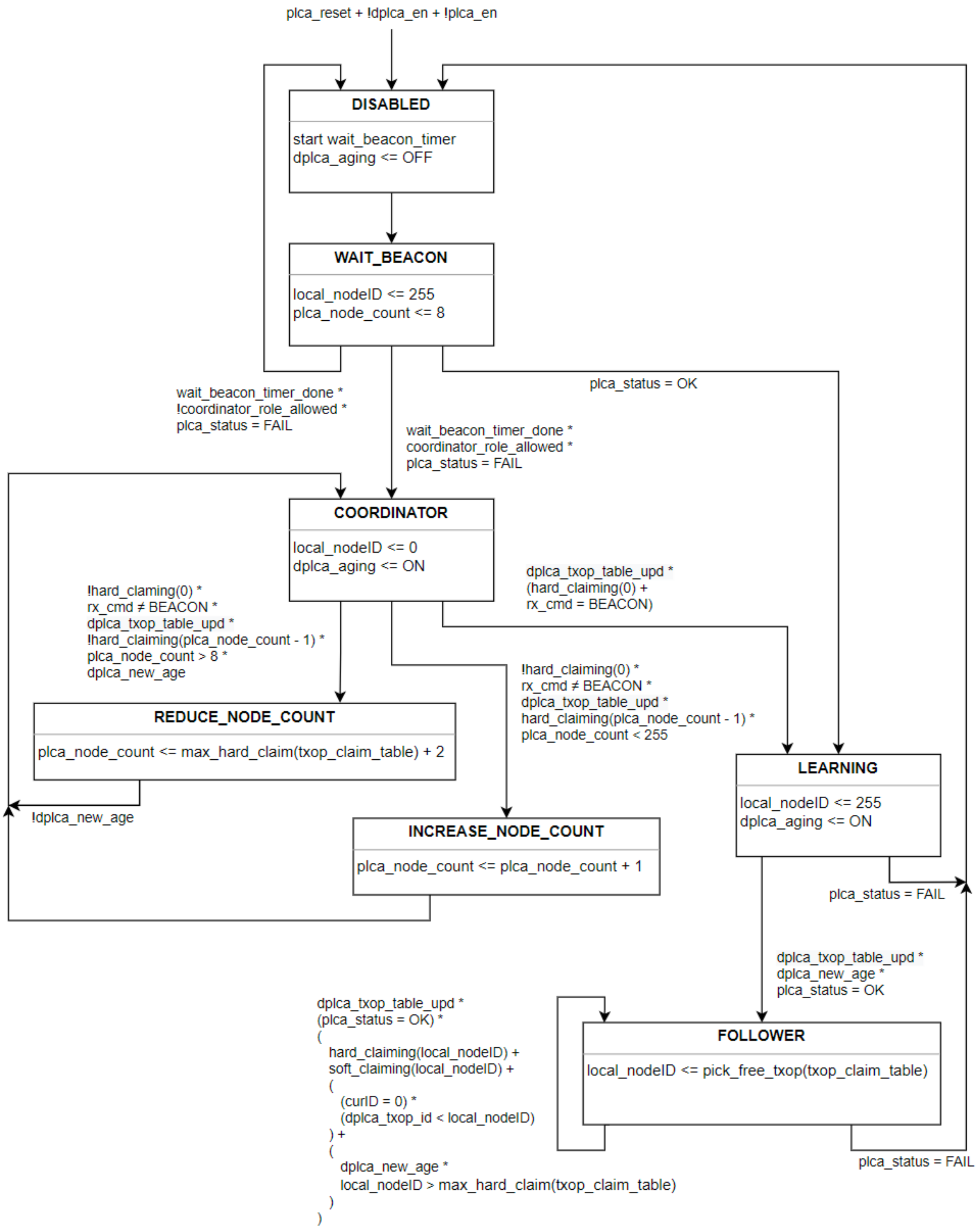
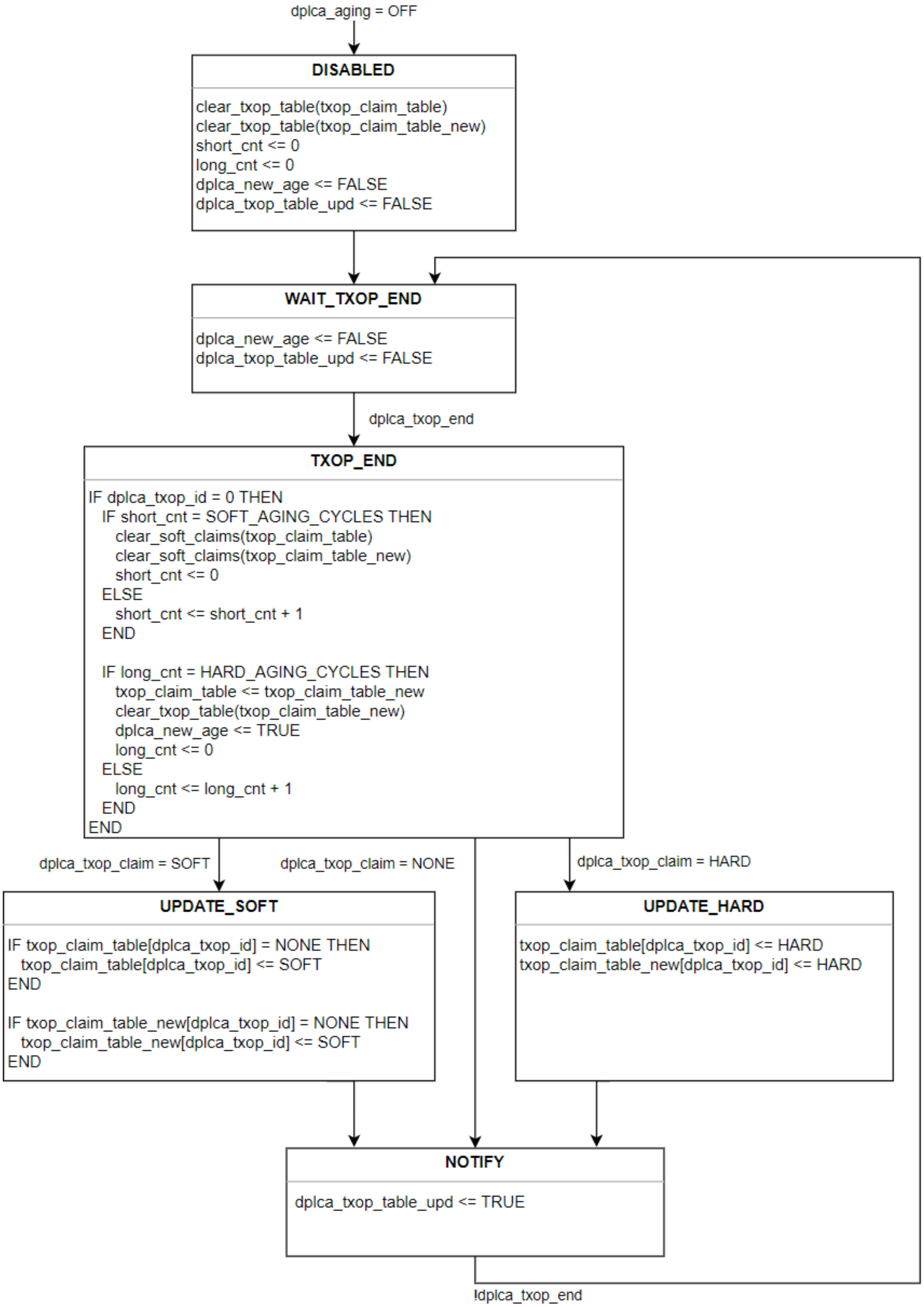


Figure TBD: D-PLCA Control State Diagram

148.4.7.6 Aging State diagram



```
clear_txop_table(txop_claim_table)
clear_txop_table(txop_claim_table_new)
short_cnt <= 0
long_cnt <= 0
dplca_new_age <= FALSE
dplca_txop_table_upd <= FALSE
```

```
dplca_new_age <= FALSE
dplca_txop_table_upd <= FALSE
```

```
IF dplca_txop_id = 0 THEN
  IF short_cnt = SOFT_AGING_CYCLES THEN
    clear_soft_claims(txop_claim_table)
    clear_soft_claims(txop_claim_table_new)
    short_cnt <= 0
  ELSE
    short_cnt <= short_cnt + 1
  END

  IF long_cnt = HARD_AGING_CYCLES THEN
    txop_claim_table <= txop_claim_table_new
    clear_txop_table(txop_claim_table_new)
    dplca_new_age <= TRUE
    long_cnt <= 0
  ELSE
    long_cnt <= long_cnt + 1
  END
END
```

```
IF txop_claim_table[dplca_txop_id] = NONE THEN
  txop_claim_table[dplca_txop_id] <= SOFT
END

IF txop_claim_table_new[dplca_txop_id] = NONE THEN
  txop_claim_table_new[dplca_txop_id] <= SOFT
END
```

```
txop_claim_table[dplca_txop_id] <= HARD
txop_claim_table_new[dplca_txop_id] <= HARD
```

```
dplca_txop_table_upd <= TRUE
```

Figure TBD: D-PLCA Aging State Diagram

148.5 Protocol implementation conformance statement (PICS) proforma for Clause 148, PLCA Reconciliation Sublayer (RS)[1]

148.5.1 Introduction

The supplier of a protocol implementation that is claimed to conform to Clause 148, PLCA Reconciliation Sublayer (RS), shall complete the following protocol implementation conformance statement (PICS) proforma.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in Clause 21.

148.5.2 Identification

148.5.2.1 Implementation identification

Supplier[1]	
Contact point for inquiries about the PICS[1]	
Implementation Name(s) and Version(s)[1,3]	
Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Name(s)[2]	
NOTE 1—Required for all implementations. NOTE 2—May be completed as appropriate in meeting the requirements for the identification. NOTE 3—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).	

148.5.2.1

148.5.2.2 Protocol summary

Identification of protocol standard	IEEE Std 802.3cg-2019, Clause 148, PLCA Reconciliation Sublayer (RS)
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	
Have any Exception items been required? No [ ] Yes [ ] (See Clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3cg-2019.)	

[1] *Copyright release for PICS proformas:* Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

Date of Statement	

148.5.2.2

148.5.3 PICS proforma tables for PLCA Reconciliation Sublayer (RS)

148.5.3.1 Reconciliation Sublayer

Item	Feature	Subclause	Value/Comment	Status	Support
RS1	PLCA not supported or disabled by management interface	148.4.1	Conform to MII RS definition in Clause 22	M	Yes [ ]

148.5.3.1

148.5.3.2 Mapping of MII signals to PLS service primitives and PLCA functions

Item	Feature	Subclause	Value/Comment	Status	Support
MAP1	Mapping of PLS_DATA.request when PLCA is disabled	148.4.2.1	Specified in 22.2.1.1	M	Yes [ ]
MAP2	Mapping of PLS_DATA.indication	148.4.2.2	Specified in 22.2.1.2	M	Yes [ ]
MAP3	Mapping of PLS_CARRIER.indication when PLCA is disabled	148.4.2.3	Specified in 22.2.1.3	M	Yes [ ]
MAP4	Mapping of PLS_CARRIER indication when PLCA is enabled	148.4.2.3.1	Maps the primitive PLS_CARRIER.indication to the PLCA Data state diagram	M	Yes [ ]
MAP5	Mapping of PLS_SIGNAL.indication when PLCA is disabled	148.4.2.4	Specified in 22.2.1.4	M	Yes [ ]
MAP6	Mapping of PLS_SIGNAL.indication when PLCA is enabled	148.4.2.4.1	Map the primitive PLS_SIGNAL.indication to the PLCA Data state diagram	M	Yes [ ]

MAP7	Mapping of PLS_DATA_VALID.indication	148.4.2.5	Specified in 22.2.1.7	M	Yes [ ]
MAP8	Generation of TX_ER	148.4.2.6	Specified in 148.4.5.1	M	Yes [ ]
MAP9	Response to RX_ER indication	148.4.2.7	Specified in 22.2.1.5	M	Yes [ ]

148.5.3.2

148.5.3.3 Specific RS and PHY specification

Item	Feature	Subclause	Value/Comment	Status	Support
PLCA1	RS reaction to BEACON indication reception	148.4.3.2.1	PLCA variable rx_cmd is set to the value BEACON	M	Yes [ ]
PLCA	RS reaction when BEACON indication ceases	148.4.3.2.1	PLCA variable rx_cmd is reset to NONE unless a COMMIT indication is signaled, in which case rx_cmd shall be set as specified in 148.4.3.2.2	M	Yes [ ]
PLCA	RS reaction to COMMIT indication reception	148.4.3.2.2	PLCA variable rx_cmd is set to the value COMMIT	M	Yes [ ]
PLCA	RS reaction when COMMIT indication ceases	148.4.3.2.2	PLCA variable rx_cmd is reset to NONE unless a BEACON indication is signaled, in which case rx_cmd shall be set as specified in 148.4.3.2.1	M	Yes [ ]

148.5.3.3

148.5.3.4 PLCA Control

Item	Feature	Subclause	Value/Comment	Status	Support
CON1	PLCA Control function	148.4.4.1	Conform to Figure 148–3 and Figure 148–4	M	Yes [ ]

148.5.3.4

148.5.3.5 PLCA Data

Item	Feature	Subclause	Value/Comment	Status	Support
DAT1	PLCA Data function	148.4.5.1	Conforms to Figure 148–5 and Figure 148–6	M	Yes [ ]

148.5.3.5

148.5.3.6 PLCA Status

Item	Feature	Subclause	Value/Comment	Status	Support
STS1	PLCA Status function	148.4.6.1	Conforms to Figure 148–7	M	Yes [ ]

148.5.3.6