



PIERGIORGIO BERUTO
ANTONIO ORZELLI

IEEE 802.3cg

Interworking between PLCA enabled nodes with non-PLCA enabled nodes

September 10th, 2018



- What happens when a non-PLCA enabled node is attached to a PLCA enabled mixing segment network?
 - non-PLCA enabled nodes (from now on called “rogue” from PLCA perspective) can trigger physical collisions on the line
 - When PLCA is enabled, the PHY still detects physical collisions (see fig. 148-7/2 - TRANSMIT) which are reported to the MAC via PLS_SIGNAL.indication primitive, triggering back-offs as normal
 - Network shall still be able to operate, eventually with degraded performance, i.e.
 - All nodes shall still be able to access the media
 - PLCA improvements waived, plain CSMA/CD behavior takes precedence
 - PLCA operations resume without external intervention as soon as there are no more rogues “jamming” the PLCA mechanism.
 - Need for a way to detect this condition and notify the management layers
- In other words:
 - PLCA is CSMA/CD with improvements so it shall interwork with non-PLCA enabled nodes natively.



- Assuming that a “rogue” node is transmitting, what can actually happen?
 - The rogue transmits during another PLCA-enabled node’s transmit opportunity. Two possibilities from here:
 - a) the “robbed” node was yielding the TO (no collision)
 - b) the “robbed” node was about to transmit a packet (collision)
 - rogue transmits over a BEACON (unlikely)
 - All PLCA enabled nodes wait for a new BEACON
 - rogue detects a physical collision, PLCA enabled nodes wait for a new BEACON (not a problem)



- Case (a): rogue “steals” the TO of a PLCA enabled node which had nothing to transmit
 - all nodes receive the packet as if it was sent by the node owning the TO (not a problem, node ID is not sent on the line. Identifying the sender is the MAC’s job)
 - “robbed” node receives the packet as well but does not increment the TO counter (because of YIELD)
 - not a problem for nodes with non-zero ID (would wait for a new BEACON before transmitting, anyway)
 - problem for the PLCA coordinator (ID = 0): BEACON would be sent earlier than normal.
 - Fix to FSM is required



- Case (b): both the rogue and the PLCA enabled node transmit a packet, resulting in a collision
 - MACs back-off with increased transmission attempt counter, as normal
 - At next transmit opportunity the “robbed” PHY would transmit with increased commit time (depending on random back-off time).
 - If physical collisions keep occurring (because of the rogue transmitting), at some point the **MAC gives up (discards the packet) leaving PLCA stuck in COMMIT state.**
 - **Fix requires adding a timer to get out of COMMIT state in this case.**
 - **Also useful in case the MAC resets without notice.**



Proposed text changes

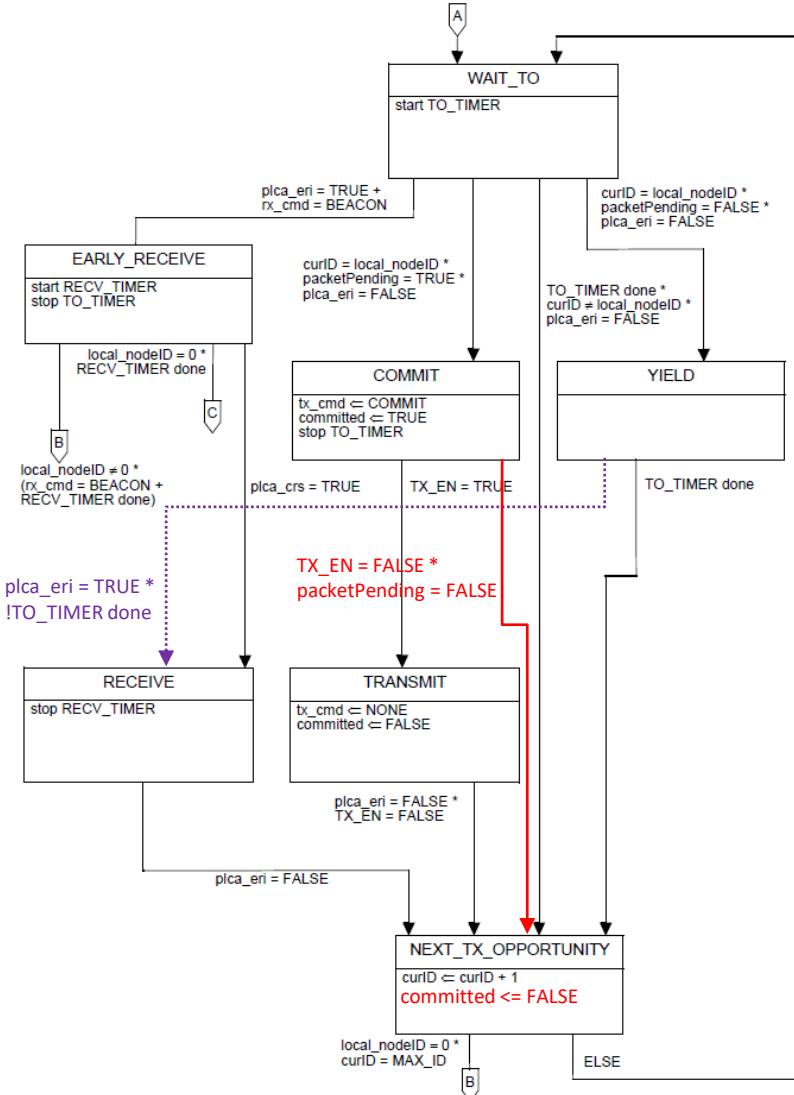


Figure 148-5—PLCA Control state diagram (2 of 2)

Add transition from YIELD to RECEIVE state with condition “plca_eri = TRUE * !TO_TIMER done”

Add transition from COMMIT to NEXT_TX_OPPORTUNITY state with condition “TX_EN = FALSE * packetPending = FALSE”

Add “committed <= FALSE” in NEXT_TX_OPPORTUNITY state box

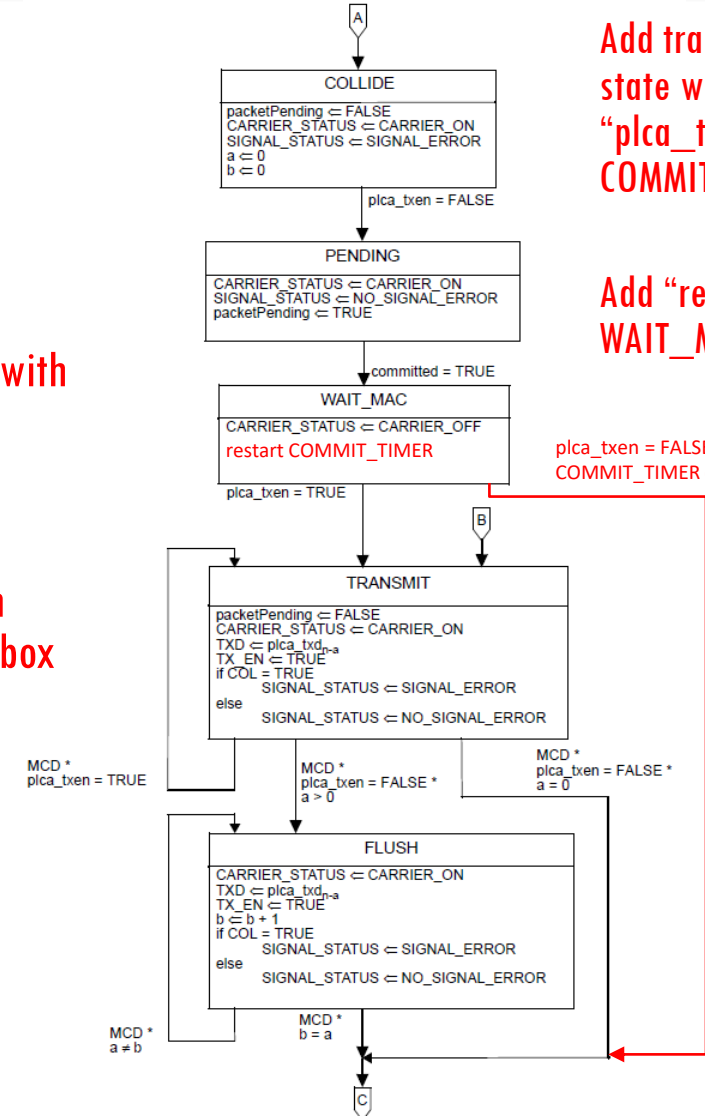


Figure 148-7—PLCA DATA state diagram (2 of 2)

Add transition from WAIT_MAC to COLLIDE state with condition “plca_txen = FALSE * COMMIT_TIMER done”

Add “restart COMMIT_TIMER” in WAIT_MAC state box

plca_txen = FALSE * COMMIT_TIMER done



- 148.4.6.4 Timers
 - COMMIT_TIMER
 - Defines the maximum time the PLCA Data state machine is allowed to stay in WAIT_MAC state.
Duration: 192 bit times
- 45.2.3
 - Add register 3.2294 “10BASE-T1S PCS Diagnostic 2”
 - Add register description

Bit(s)	Name	Description	R/W
3.2294.15:0	PhysicalColCnt	16 bit field counting the number of physical collisions occurred since last read of this register.	RO - SC

- PhysicalColCnt
 - Shall report the number of physical collisions (i.e. excluding the ones triggered by the optional PLCA RS) occurred since last time register 3.2294 was read.



- PLCA enabled nodes are expected to interwork with non PLCA enabled nodes on the same mixing segment network
 - When all nodes support and enable PLCA, performance is improved as expected
 - Otherwise the network behaves just as in normal CSMA/CD mode
- Bugs were found in current specifications that in some corner cases could lead to interworking problems in presence of physical collisions:
 - PLCA coordinator could send a BEACON earlier when yielding the transmit opportunity
 - PLCA could get stuck in COMMIT/WAIT_MAC states if MAC perform too many transmission attempts (because of physical collisions) and discards the packet
- Proposals to fix the bugs were shown
 - Addition of a diagnostic register to report physical collisions is proposed as well
- **NOTE: proposed fixes also cover the case of accidental assignment of duplicate node IDs, which triggers physical collisions. See also comment #286.**

THANK YOU!



- During IEEE Interim in Spokane it has been pointed out that:
 - Having PLCA-enabled nodes properly interworking with non PLCA-enabled nodes is a desirable enhancement from an architectural perspective but...
 - This situation occurs as a result of a mis-configured network
 - Simply allowing the network to keep working as-is in such a case is not right in most cases
 - Even thou the problem is properly detected and reported via management interface
 - A more desirable behavior is to have PLCA disabled and allow the management entity to take appropriate actions
- This addendum specifies an additional configuration option to implement such behavior



- The idea is to add a bit in a register to have PLCA automatically disabled when a certain number of physical collision is detected
- This feature is enabled by default
- More proposed text changes in Clause45_r2p0_resolution_v2.pdf
- Additionally, include text changes reported in next slide



- **30.3.9.2.1 acPLCAutoRevert**
ACTION

APPROPRIATE SYNTAX: Same as aPLCAAdminState

BEHAVIOUR DEFINED AS: This action provides a means to alter aPLCAAutoRevert. Setting acPLCAAutoRevert to the enabled state will result in aPLCAAdminState to be reset to disabled state as a result of detecting physical collisions as specified in 45.2.13.1.3;

**THANK YOU
AGAIN!**