D-PLCA Comment #48
Algorithm Optimization
(Removal of SOFT claims)

A Leading Provider of Smart, Connected and Secure Embedded Control Solutions

Tim Baggett

IEEE 802.3da Interim May 2025

Introduction & Background

- **PLCA – Physical Layer Collision Avoidance**
  - Defined in IEEE 802.3-2022 Clause 148
  - Intended to allow a mix of non-PLCA and PLCA enabled nodes *(accepting performance degradation)*

- **D-PLCA – Dynamic Physical Layer Collision Avoidance**
  - Optional extension to Clause 148, defined in IEEE 802.3da
  - Allows for plug-and-play networking
  - *Intended* to allow a mix of non-PLCA, PLCA, and D-PLCA enabled nodes

Introduction & Background

- **D-PLCA nodes monitor which TO are in use and move as needed**
  - Select an unused TO – one that has not heard another node transmit
  - If a D-PLCA node hears another node transmit using the same TO it has selected, it then switches to using a new unused TO
  - The system eventually converges with each node settling on its own TO
    - *This is the Goal, anyway…*
- **PLCA nodes without D-PLCA simply use their configured / assigned Transmit Opportunity**
- **Non-PLCA nodes jump in and transmit when they can (CSMA/CD)**

D-PLCA Transmit Opportunity Claims

- **Two types of claims: HARD and SOFT**
- **HARD claim**
  - PLCA and D-PLCA nodes are identified by their transmission of COMMITs
    - PLCA nodes **prepend** COMMITs to packets
      - But not always – only when PLCA RS had to assert a logical/emulated collision to the MAC
    - D-PLCA nodes always **append** COMMITs to packets
  - The detection of a COMMIT during a transmit opportunity means that a PLCA or D-PLCA node has made a HARD claim on that TO
- **SOFT claim**
  - Packets received during a TO **without** a COMMIT prepended or appended create a SOFT claim
  - Non-PLCA and nodes operating strictly in CSMA/CD generate SOFT claims
  - PLCA nodes will *occasionally* generate a SOFT claim when a logical/emulated collision is not asserted

D-PLCA TO Claim Aging

- **SOFT claims age and expire faster**
  - Non-PLCA nodes transmit when able according to CSMA/CD
    - Evenly affect all transmit opportunities
  - PLCA nodes transmitting a packet without prepended COMMIT
    - Affect only their own claimed TO; will soon enough generate a new HARD claim

- **HARD claims age and expire slower**
  - PLCA nodes transmitting a packet with a prepended COMMIT
  - D-PLCA nodes transmitting a packet with an appended COMMIT

D-PLCA TO Claim Aging

- **SOFT claims cover a PLCA node's TO when they do not append a COMMIT, covering the time until they do make a HARD claim.**

- **Expire HARD claims to allow nodes to drop off and eventually return their TO**
  - Part of the "un-plug and play"

- **Convergence time and stability of D-PLCA nodes claiming a unique TO / NodeID depends on the SOFT and HARD aging times.**

Issue – non-PLCA Nodes in D-PLCA

- **Non-PLCA nodes operating strictly in CSMA/CD randomly generate SOFT claims throughout the PLCA cycle**
- **When a D-PLCA node detects a SOFT (or HARD) claim in its TO, it will select a new, unclaimed TO**
  - The SOFT claim *may* be due to a statically assigned PLCA node
- **Randomness of CSMA/CD packets being transmitted within a PLCA network results in D-PLCA nodes being "kicked around"**
  - Negative impact on both convergence and stability
- **This is in addition to the negative effect non-PLCA nodes have in a PLCA network (collisions)**
  - Lower throughput, increased latency

Resolution – Disallow non-PLCA Nodes in D-PLCA enabled segments

- **Eliminate SOFT claims by eliminating non-PLCA nodes when D-PLCA is to be used**
  - HARD claims are determined *only* by reception of a packet in TO
    - No longer need to detect prepended or appended COMMITs
  - A TO can now simply be either CLAIMED or UNCLAIMED

- **Positive results:**
  - Simulations indicate convergence and stability!
  - PLCA / D-PLCA state diagrams are optimized / simplified

Editing Instructions

- **For detailed editing instructions, please see:**
  - Baggett_3da_Cmt48_EditingInstructions_v03.pdf

Editing Instructions - Summary

- **30.16.1.1.8 (P32)**
  - L11 – Delete Section aDPLCASoftAgingCycles
- **30.16.1.1.9 (P32)**
  - L21 – Rename to aDPLCAHardAgingCycles to aDPLCAAgingCycles
  - L28 – Delete “HARD”
  - L29 – Rename hard_aging_cycles to aging_cycles
- **148.4.4.1 (P67)**
  - L35-49 – Update descriptive text, eliminating reference to HARD/SOFT claims
- **148.4.4.2 (P68)**
  - L42 – Update description for dplca_txop_claim relating to HARD/SOFT claims
- **148.4.4.4 (P69)**
  - L17-23 – Delete append_commit_timer
- **Figure 148-4 (P71)**
  - Delete SOFT claims, Delete SOFT claims on collision, Delete COMMIT/TRANSMIT state circulators, Change HARD claim to CLAIMED, Change NONE to UNCLAIMED, Revert burst COMMIT deleting append COMMIT

Editing Instructions - Summary

- **148.4.7.1 (P75-76)**
  - Replace descriptive text in third paragraph (L18-26)
  - Replace “hard claim” with “claim”
- **148.4.7.2 (P76)**
  - L47 – Rename hard_aging_cycles to aging_cycles
  - L52 – Rename aDPLCAHardAgingCycles to aDPLCAAgingCycles
  - L48 – Delete “HARD”
- **148.4.7.2 (P77)**
  - L4 – Rename long_cnt to aging_cnt
  - L5 – Delete “long”, delete “HARD claims”
  - L19-22 – Delete variable short_cnt
  - L23-28 – Delete variable soft_aging_cycles
  - L32 – Change descriptive text for txop_claim_table
    - Replacing NONE with UNCLAIMED (L35), HARD with CLAIMED deleting text relating to COMMIT indication (L41), Deleting the entry for SOFT (L38) , changing “NONE, SOFT, or HARD” to “UNCLAIMED or CLAIMED” (L45)

Editing Instructions - Summary

- **148.4.7.2 (P77)**
  - L48 – Delete “HARD”
- **148.4.7.3 (P78)**
  - L3-6 – Delete function CLEAR_SOFT_CLAIMS
  - L12 – Change function HARD_CLAIMING to CLAIMING
    - L15 – Change HARD to CLAIMED
  - L16 – Change function MAX_HARD_CLAIM to MAX_CLAIM
    - L18 – Change “HARD claimed” to “CLAIMED”
    - L19 – Change “claimed” to “CLAIMED”
  - L22 – Change function PICK_FREE_TXOP
    - L23 – Change “HARD or SOFT claimed” to “CLAIMED”
    - L27,28,29 – Change “HARD claimed” to “CLAIMED”
    - L31 – Change “HARD or SOFT” to “CLAIMED”
  - L36-40 – Delete function SOFT_CLAIMING

Editing Instructions - Summary

- **148.4.7.5 Figure 148-8 (P79)**
  - Change all instances of HARD_CLAIMING to CLAIMING
  - L49 Delete term “SOFT_CLAIMING(local_nodeID)+”
- **148.4.7.6 Figure 148-9 (P80)**
  - L7 – Delete short_cnt line
  - All – Rename:
    - “long_cnt” to “aging_cnt”,
    - “hard_aging_cycles”, “aging_cycles”,
    - “NONE” to “UNCLAIMED”, and “HARD” to “CLAIMED”
  - L19-23 – Delete IF/ELSE/END for short_cnt processing
  - L34 – Rename state “UPDATE_HARD” to “UPDATE_CLAIMED”
  - L34-42 – Delete state “UPDATE_SOFT”

Thank You

Questions?

Straw Poll

I support removing support for non-PLCA nodes operating in a D-PLCA segment and adopting the changes in

Baggett_3da_Cmt48_EditingInstructions_v03.pdf

Yes: 25

No: 0

Abstain: 10