Idle insertion using Carrier Sense

Eric Lynskey

IEEE 802.3av Task Force

July 17 – 19, 2007 San Francisco, CA

Problem statement

- IDLE needs to be inserted between frames to make room for parity.
  - Add idle using MPCP as is done in EPON
  - Add idle using MAC as is done in 10G WAN
  - Add idle using carrierSense as is done in EFM Cu

- This presentation proposes a method of using the carrierSense signal to force the MAC to defer transmission until enough idle has been added between frames.

How the MAC transmits a frame (1)

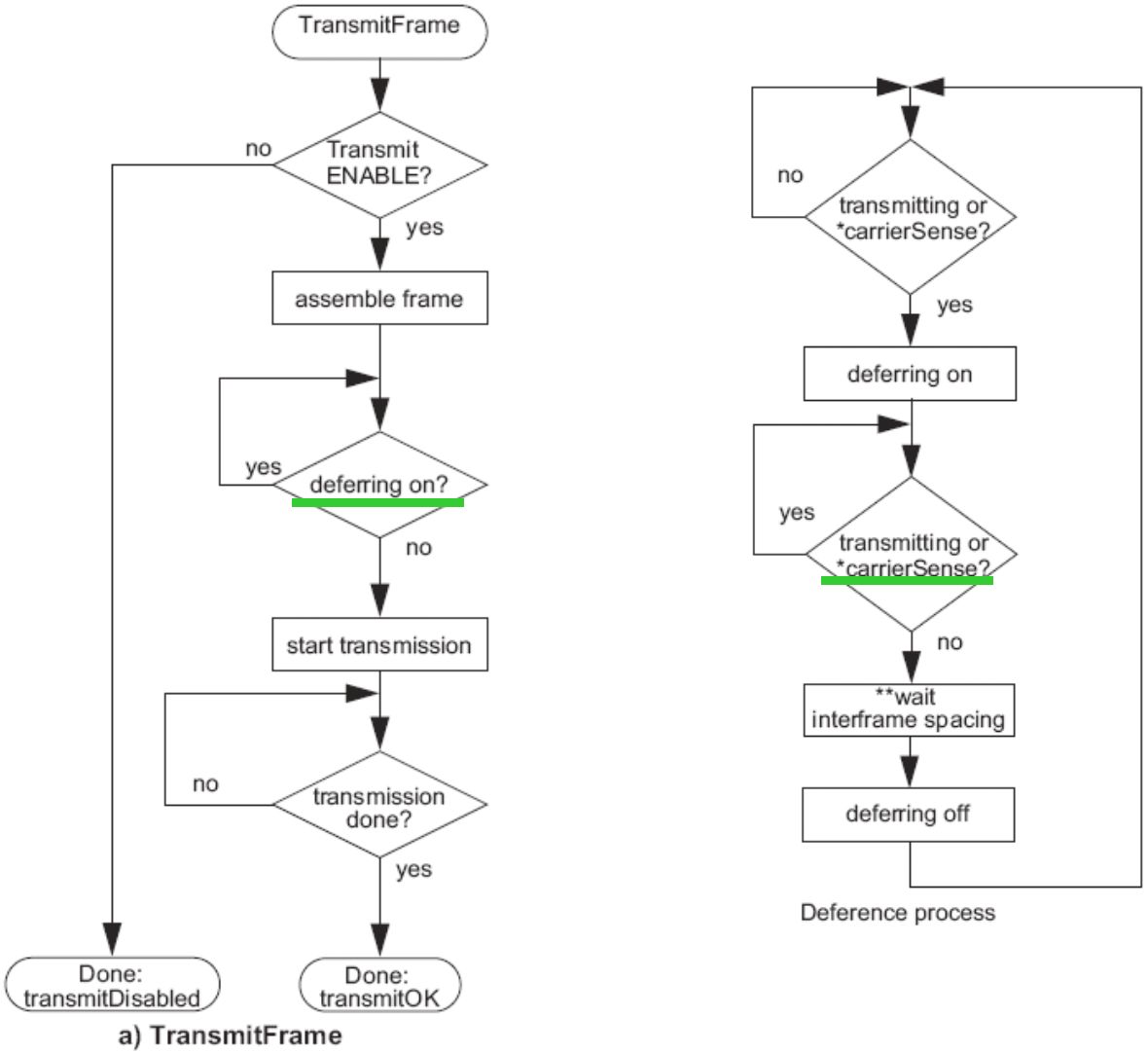


Figure 4A–2a—Control flow summary

How the MAC transmits a frame (2)

- TransmitFrame
  - Attempts to construct and transmit frame, and then returns with status. The MAC can accept another frame once this function returns.
- TransmitLinkMgmt
  - Called by TransmitFrame, this function monitors "deferring" variable and waits for deferring to be false before transmitting a new frame.
- Deference
  - This process runs asynchronously and controls the deferring variable. After frame transmission starts, deferring is true until carrierSense is false. MAC then enforces minimum IPG.
- carrierSense
  - Boolean that indicates the physical layer is not ready for the next packet and that transmission should defer.

Where does carrierSense come from?

- PLS_CARRIER.indication drives carrierSense
- Although defined, this primitive is not currently used for 10Gb/s.
- Propose that primitive is used for 10GEPON

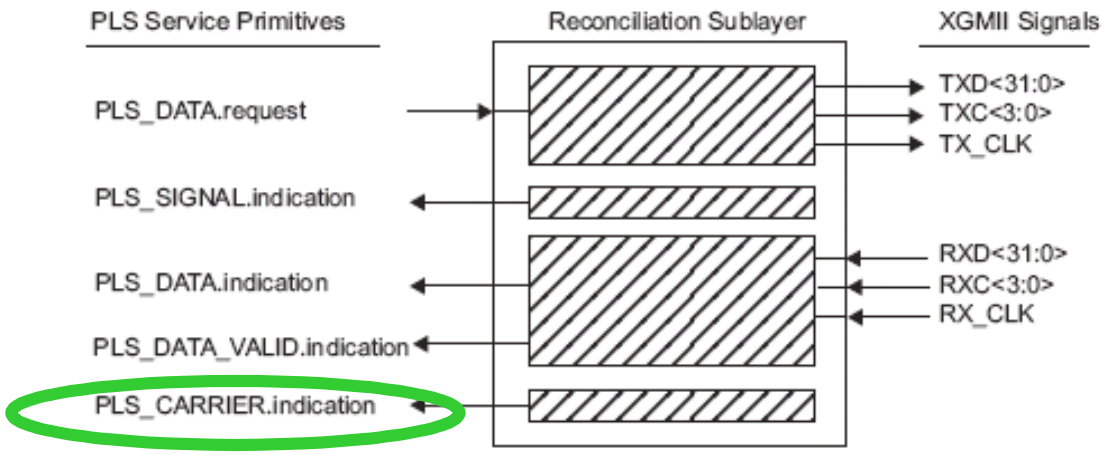
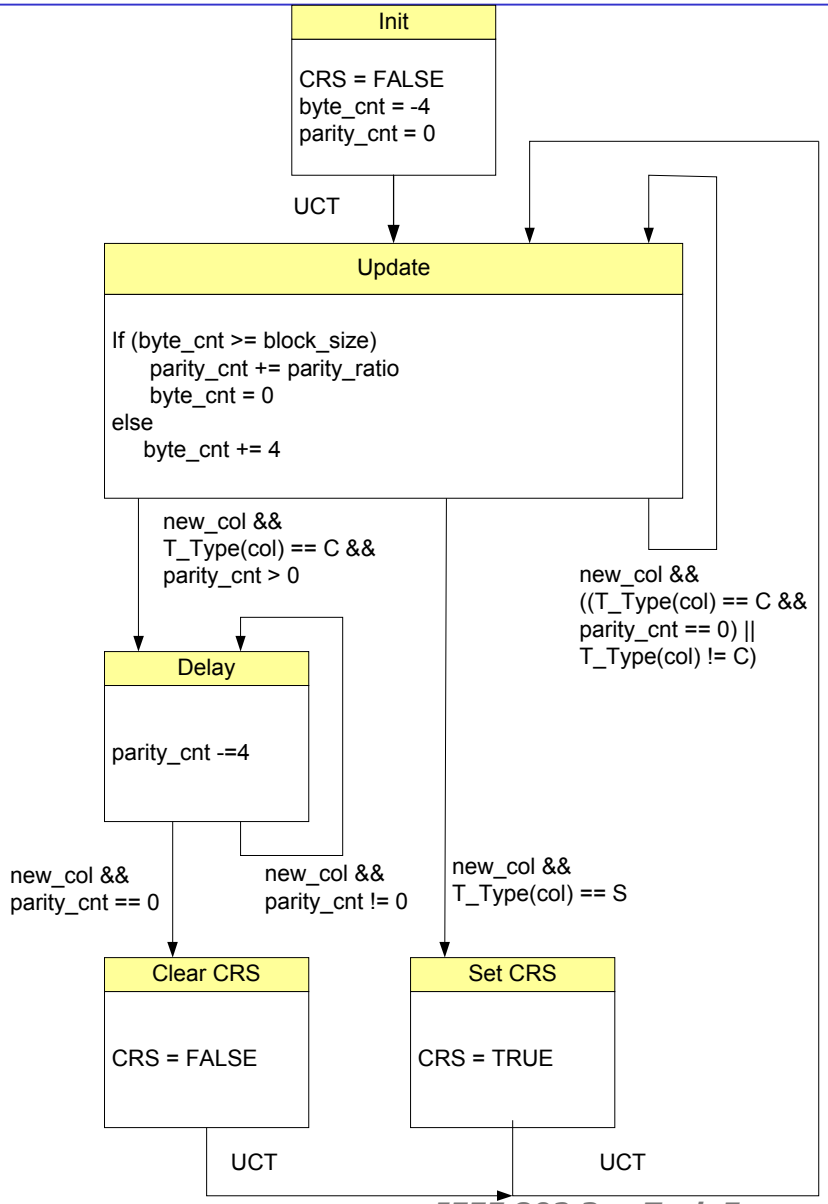


Figure 46–2—Reconciliation Sublayer (RS) inputs and outputs

RS state diagram proposal



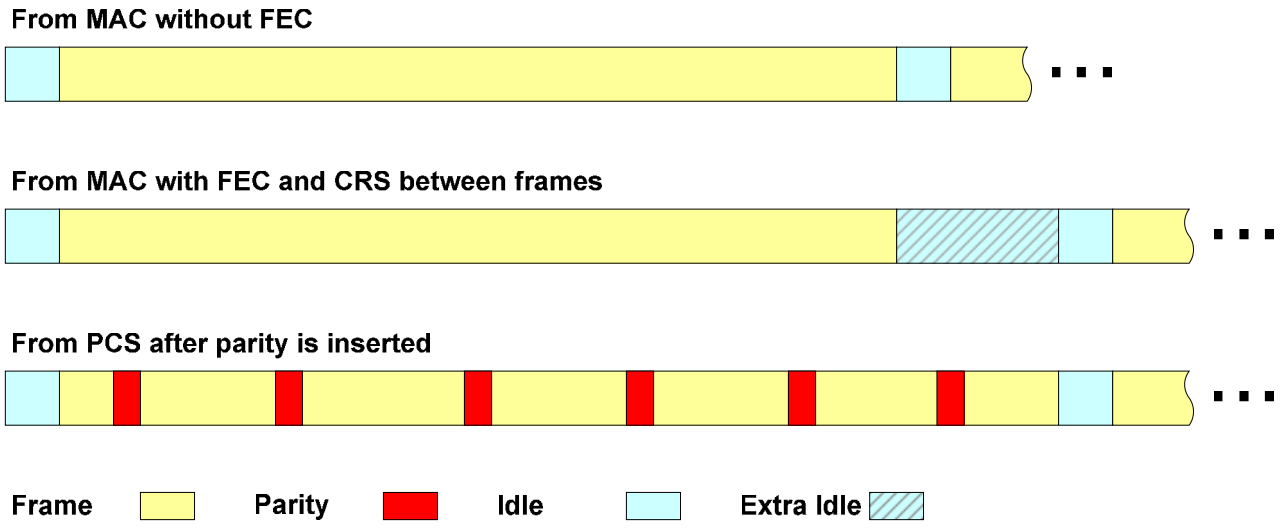
- **CRS** = carrier sense signal
- **byte_cnt** = number of bytes (idle + data) transmitted
- **parity_cnt** = number of parity bytes that need to be inserted by PCS
- **block_size** = size of FEC block
- **parity_ratio** = number of parity bytes to be inserted for every FEC block
- **new_col** = indicates that new column is available to transmit
- **T_Type()** = function to determine what type of column is to be transmitted
- **col** = contents of current column

State diagram walkthrough

- Init
  - Initialize variables and counters
- Update
  - Increment counters to keep track of how many parity bytes need to be inserted.
- Set CRS
  - Assert carrierSense signal once start of frame has been seen (MAC finishes current frame, cannot start another).
- Clear CRS
  - Deassert carrierSense signal once enough IDLE has been transmitted to account for all inserted parity.
- Delay
  - Once the current frame transmission is complete, keep track of how long the MAC needs to defer.

Result of adding IDLE

- MAC transmits frame.
- MAC defers between frames the correct number of bits.
- MAC enforces minimum IPG.
- PCS inserts parity within frame and removes extra idles.



Possible concerns

- Additional latency, timestamp drift
  - Frame following maximum sized frame will be marked with timestamp and then wait up to 6 time_quanta before being transmitted.
  - This is within the allowable drift for both OLT and ONU. At 10Gb/s speeds, this still allows for 2 time_quanta, or 40 bytes of delay.

Why use carrierSense?

- It is impossible for MPCP or the MAC to accurately predict how much parity will be inserted for any given frame.
- MPCP can only delay the MAC in units of time_quanta. At 10Gb/s speeds, a single time_quanta is 20 bytes.
- Only the PHY knows when it is inserting parity.
- The PHY should be in control of how much idle is needed between frames.
- An existing signal, carrierSense, exists that allows the PHY to defer future transmissions from the MAC in units of bits.

Propose that carrierSense is used to defer MAC