Progress Report on Envelope Header and Bonding

Duane Remein Huawei
Glen Kramer Broadcom Limited

From Vancouver meeting

- A compromise on GLID to PLID/ULID mapping was proposed in remein_3ca_4_0317
  - Idea #1
    - Avoid GLID to ULID mapping in OUTPUT process; included preamble replacement, Envelope Header identifies fragments, Envelope ID is always PLID/ULID
    - May restrict envelopes to single frame
  - Idea #2
    - /S/ always left justified in EQ (approved with motion #7 in New Orleans), allow for DPoE style encryption
  - Expected advantages
    - Simpler SDs
    - Reduced overhead
    - Higher usefulness for GLIDs

The State of the Compromise so far

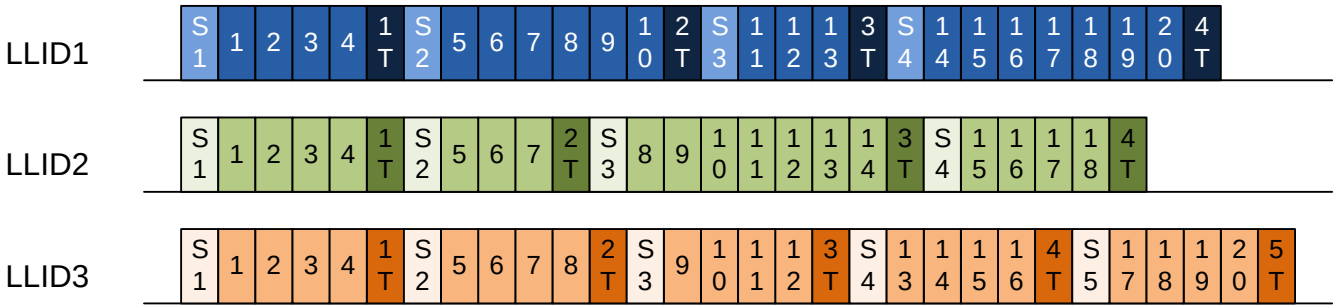
- General Description
  - Including two transmission examples
- Issues and current proposed fixes
  - Input Process SD
  - Output Process SD
  - Envelope Header structure

General Description

- Envelope
  - A transmission opportunity for a single PLID or ULID
  - always begins with an Envelope Header in the transmission channel
  - All preambles in an Envelope are replaced with an Envelope Header
    - MPRS requirement, see modified Input process SD
- Burst
  - Preceded by a fixed amount of overhead (details TBD)
    - Must be the same in each channel (equates to a fixed delay from MPCP perspective)
    - In 10G-EPON – Sync time, Burst Delimiter, 2 Idles
- MPCP/MPRS
  - Should be loosely coupled
    - MPCP delivers large envelopes decoupled from frames and frame sizes
    - MPRS handles header insertion, preamble replacement, FEC alignment, …

Example 1

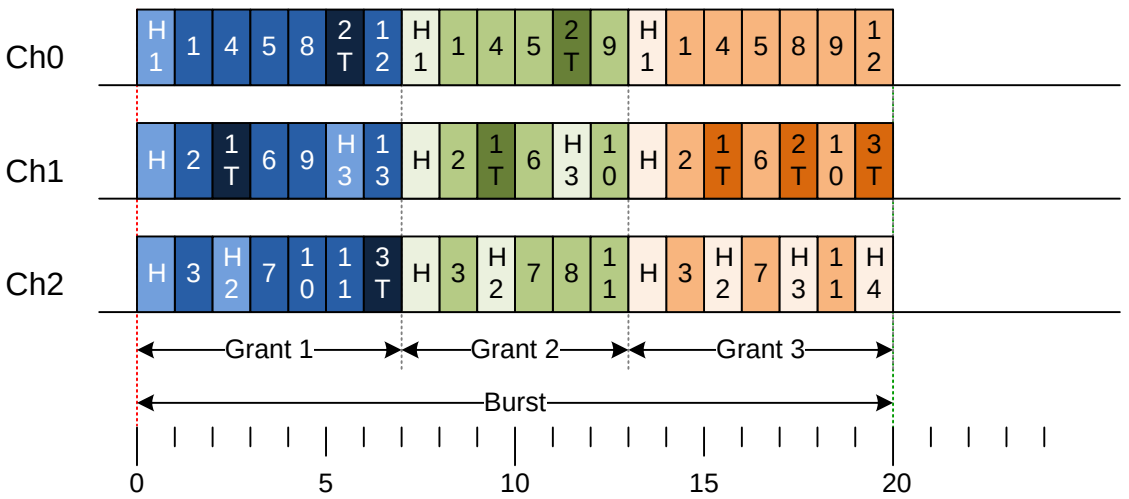
- Assume an ONU has the following queued frames for transport



- The OLT issues the following GATE:

Ch(0111); ST=T0; ULID(1), len=7; ULID(2), len=6; ULID(3), len=7;

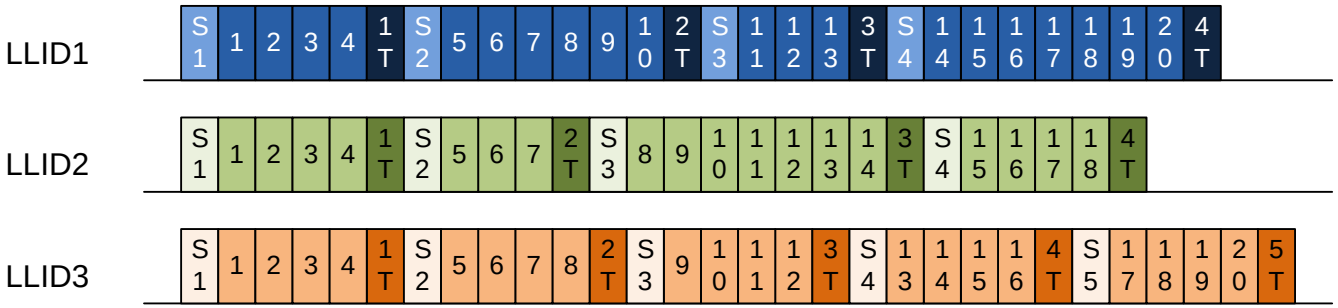
- The ONU would return the following:



S 1 Preamble frame #1 of "blue" LLID
H 1 EnvHdr (Pre replace) frm #1, "blue" LLID
3 EQ #3 of "blue" LLID
H
1 T Term of frame #1 "blue" LLID

Example 2a

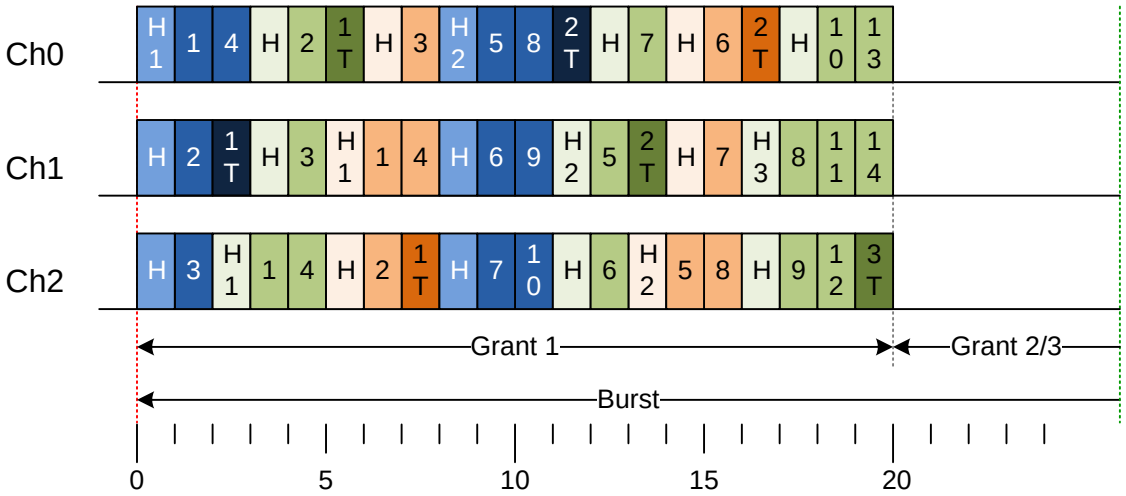
- Same queued frames



- What if the OLT issues the following GATE instead?

Ch(0111); ST=T0; GLID(1,2,3), len=20; ... ;

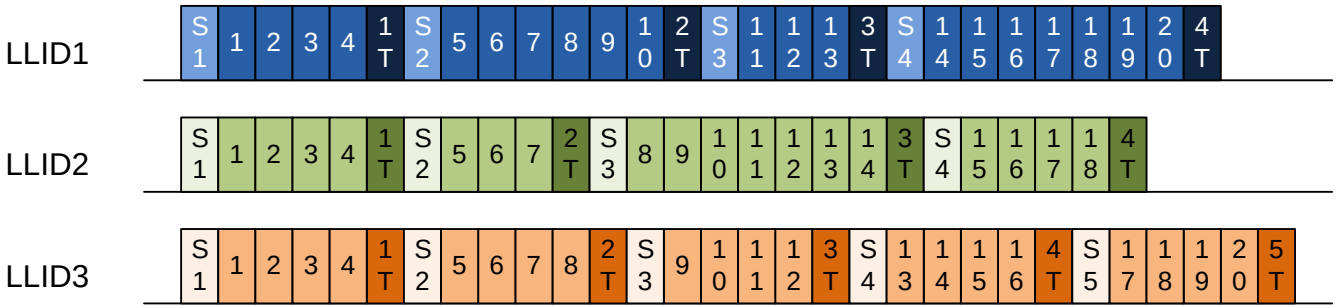
- The ONU could return the following:



S1 Preamble frame #1 of "blue" LLID
H1 EnvHdr (Pre replace) frm #1, "blue" LLID
3 EQ #3 of "blue" LLID
H
1T Term of frame #1 "blue" LLID

Example 2b

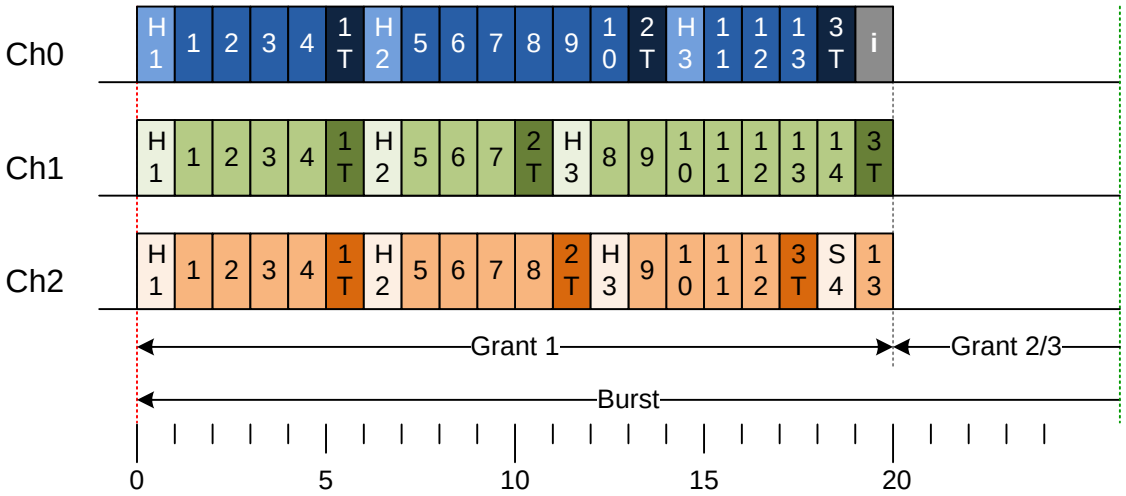
- Same



- Same GATE as in Ex 2a:

Ch(0111); ST=T0; GLID(1,2,3), len=20; ... ;

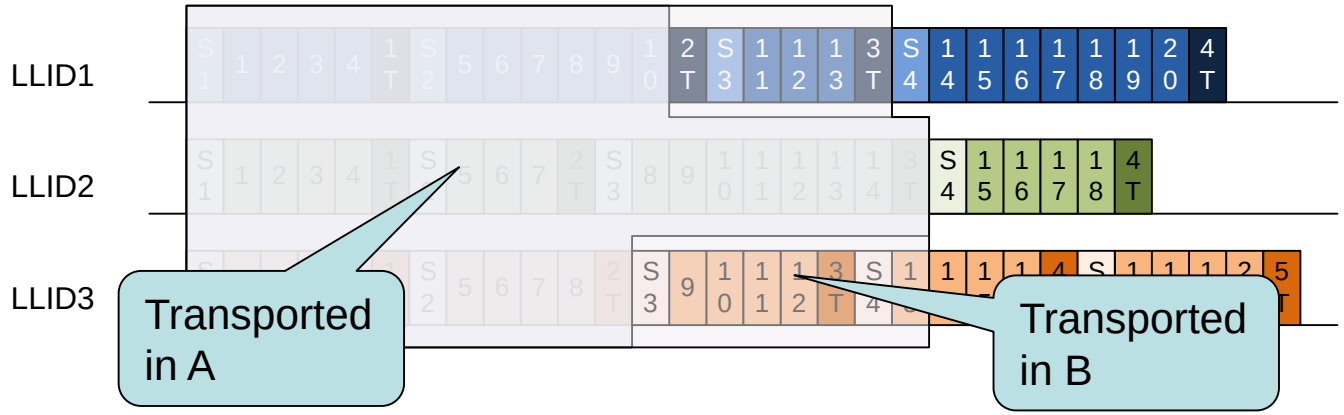
- OR the ONU could return this:



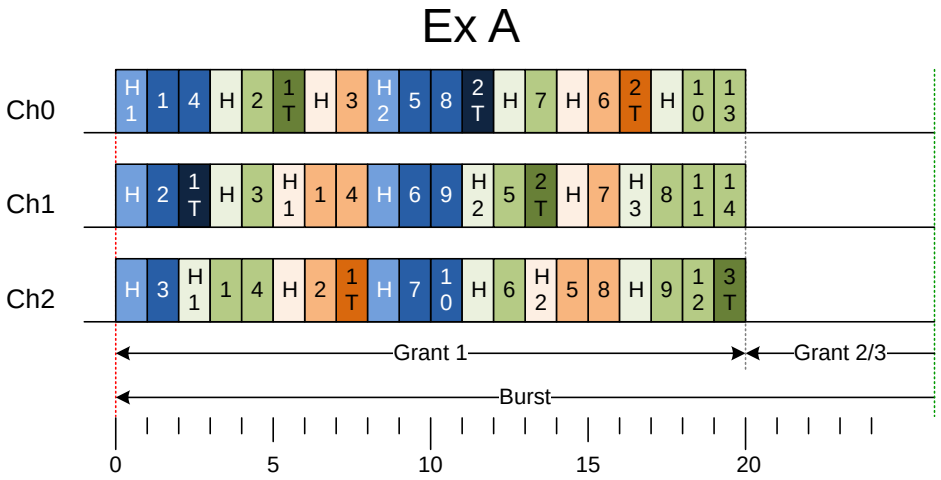
S 1 Preamble frame #1 of "blue" LLID
H 1 EnvHdr (Pre replace) frm #1, "blue" LLID
3 EQ #3 of "blue" LLID
H
1 T Term of frame #1 "blue" LLID

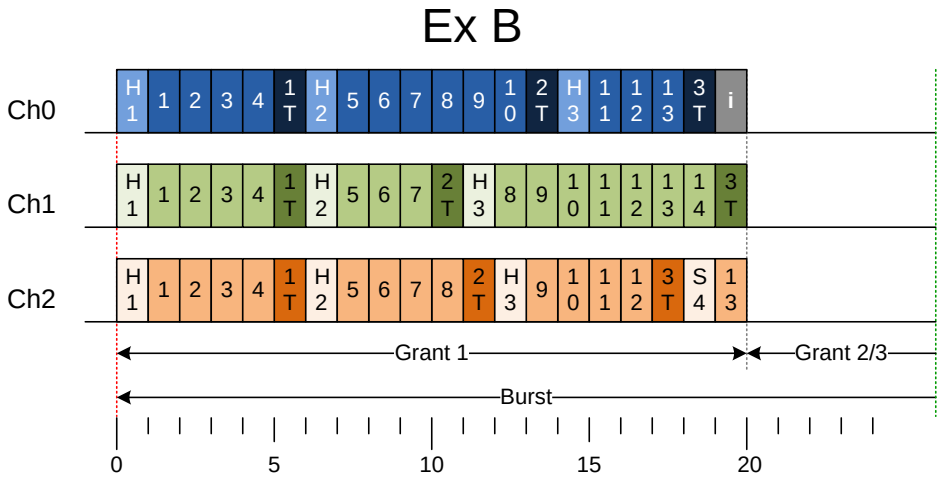
Ex 2

- Both responses are valid
- Both have some benefits
- But they are not equivalent



- However which response the ONU has is OOS for P802.3ca





Issues and fixes

Input SD (D0.4)

- MPCP must issue a MPRS_CTRL.request for each frame in order to achieve preamble replacement
  - Requires close coupling of MPRS and RS

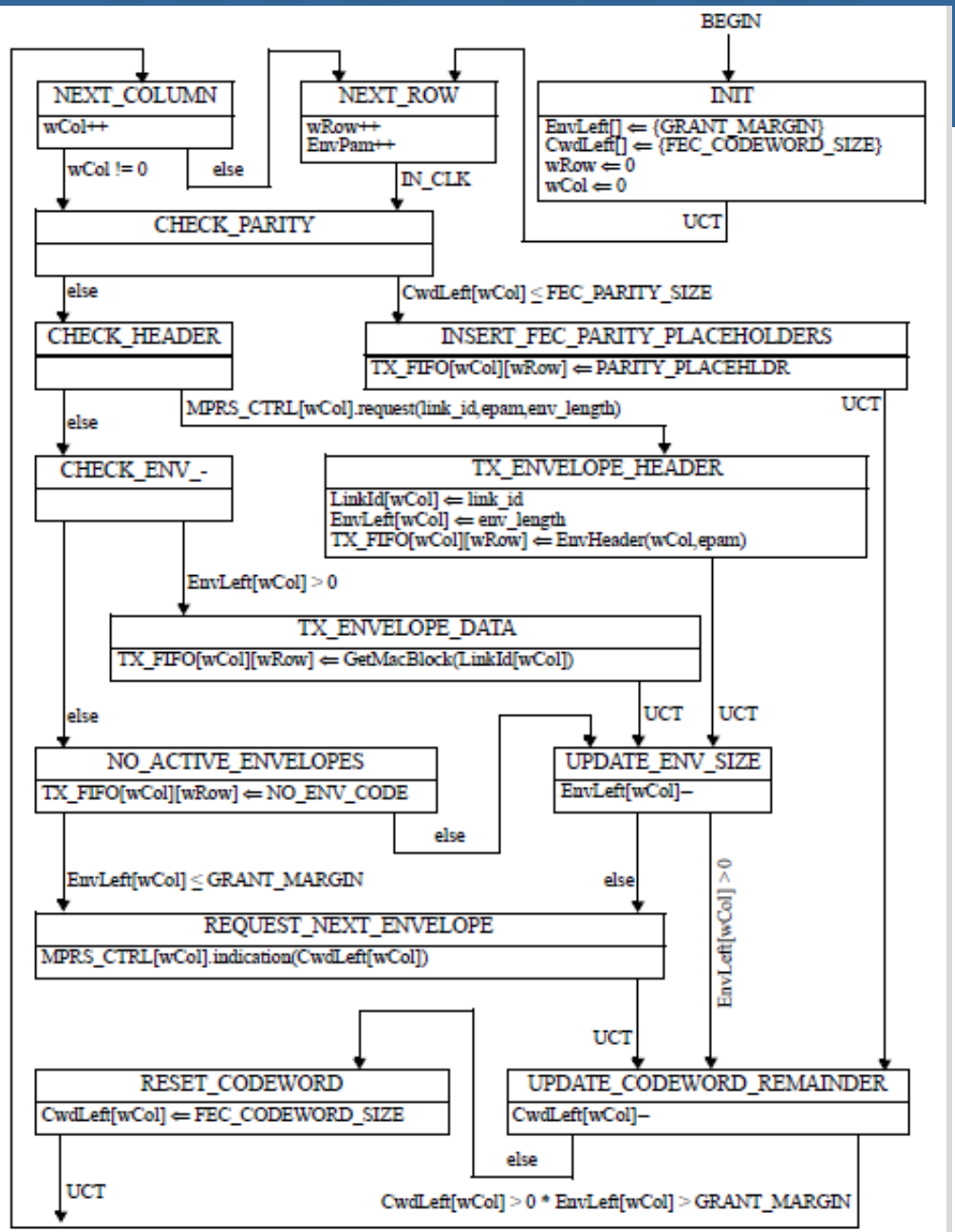
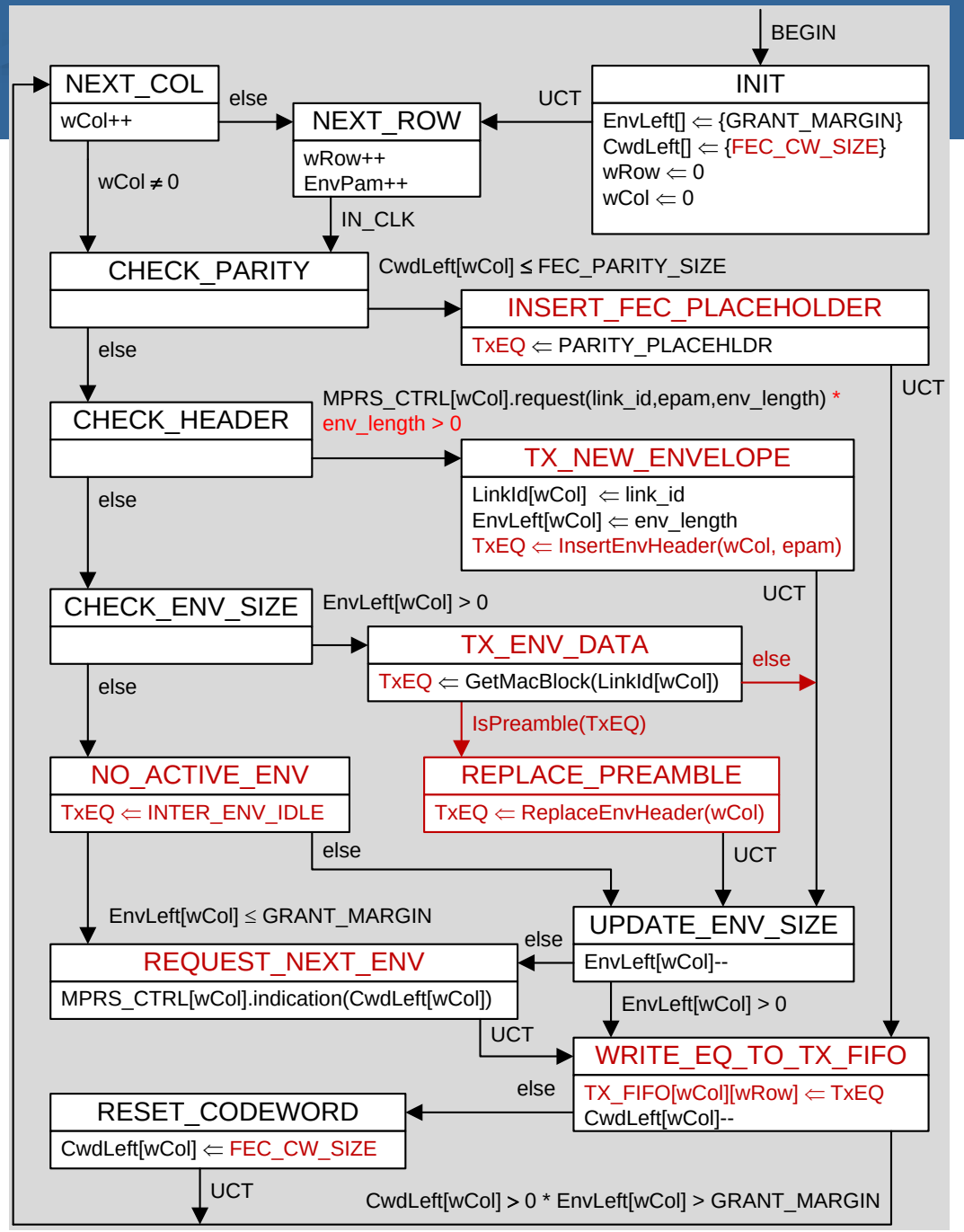


Figure 143–1—ONU MPRS Input Process state diagram

Revised Input Process SD

- Decouples MPCP and MPRS
  - MPCP can issue large Envelopes
  - MPRS handles Envelope Header insertion and Preamble Replacement



SD Stylistic changes (enter/exit from side of state) are at the Editors discretion

Revised Input Process change summary

- New State:
  - REPLACE_PREAMBLE
- New functions:
  - InsertEnvHeader(); inserts an envelope header with Preamble flag = 0
  - ReplaceEnvHeader(); replaces a Preamble with an envelope header with Preamble flag = 1
- New variable:
  - TxEQ; local variable that holds one EQ
- Renamed constants:
  - FEC_CW_SIZE (FEC_CODEWORD_SIZE)
  - INTER_ENV_IDLE (NO_ENV_CODE)
- Renamed States:
  - INSERT_FEC_PLACEHOLDER (INSERT_FEC_PARITY_PLACEHOLDERS)
  - TX_NEW_ENVELOPE (TX_ENVELOPE_HEADER)
  - TX_ENV_DATA (TX_ENVELOPE_DATA)
  - NO_ACTIVE_ENV (NO_ACTIVE_ENVELOPES)
  - REQUEST_NEXT_ENV (REQUEST_NEXT_ENVELOPE)
- Changed state actions:
  - TxEQ is set in various states rather than load TX_FIFO directly
  - TX_FIFO is set in WRITE_EQ_TO_TX_FIFO (UPDATE_CW_REMAINDER)

Revised Input Process functions (1 of 2)

InsertEnvHeader

```
InsertEnvHeader(int2 col, int5 epam)
{
    EQ hdr;

    if( EnvLeft[col+1] == GRANT_MARGIN &&
        EnvLeft[col+2] == GRANT_MARGIN &&
        EnvLeft[col+3] == GRANT_MARGIN ) EnvPam = epam;

    hdr<9>     = 0;                    //No preamble replacement
    hdr<10:31> = EnvLeft[col];         //EnvLength
    hdr<32:39> = EnvPam;               //EPAM
    hdr<40:55> = LinkId[col];          //LLID

    return hdr;
 }
```

Revised Input Process functions (2 of 2)

- ReplaceEnvHeader

```
ReplaceEnvHeader(int2 col)
{
    EQ hdr;

    hdr<9>     = 1;                    //Preamble replacement
    hdr<10:31> = EnvLeft[col];         //EnvLength
    hdr<32:39> = EnvPam;               //EPAM
    hdr<40:55> = LinkId[col];          //LLID

    return hdr;
 }
```

Output SD (D0.4)

- Does not account for INTER_ENV_IDLE (was NO_ENV_CODE)
- Does not account for Preamble replacement

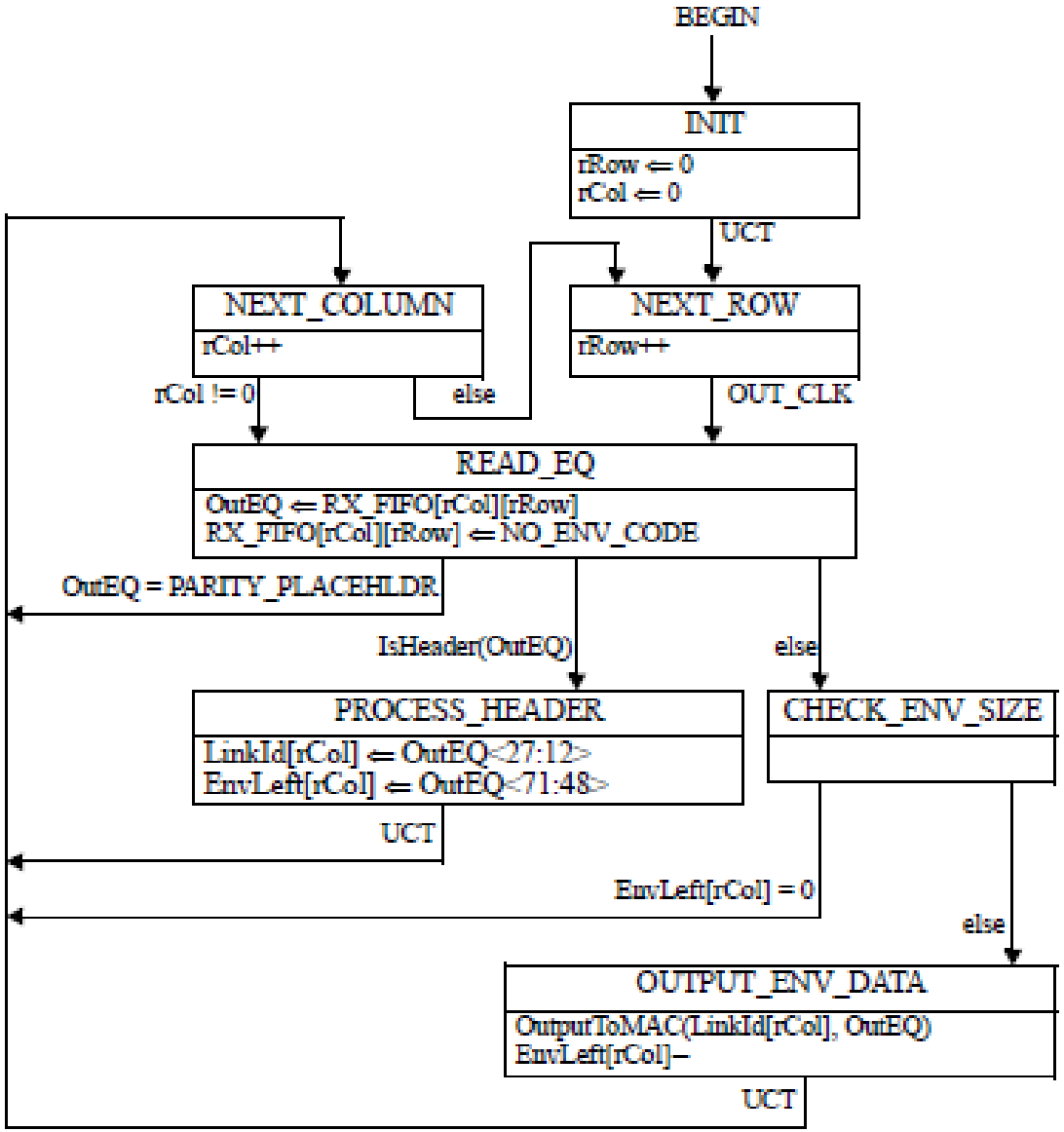
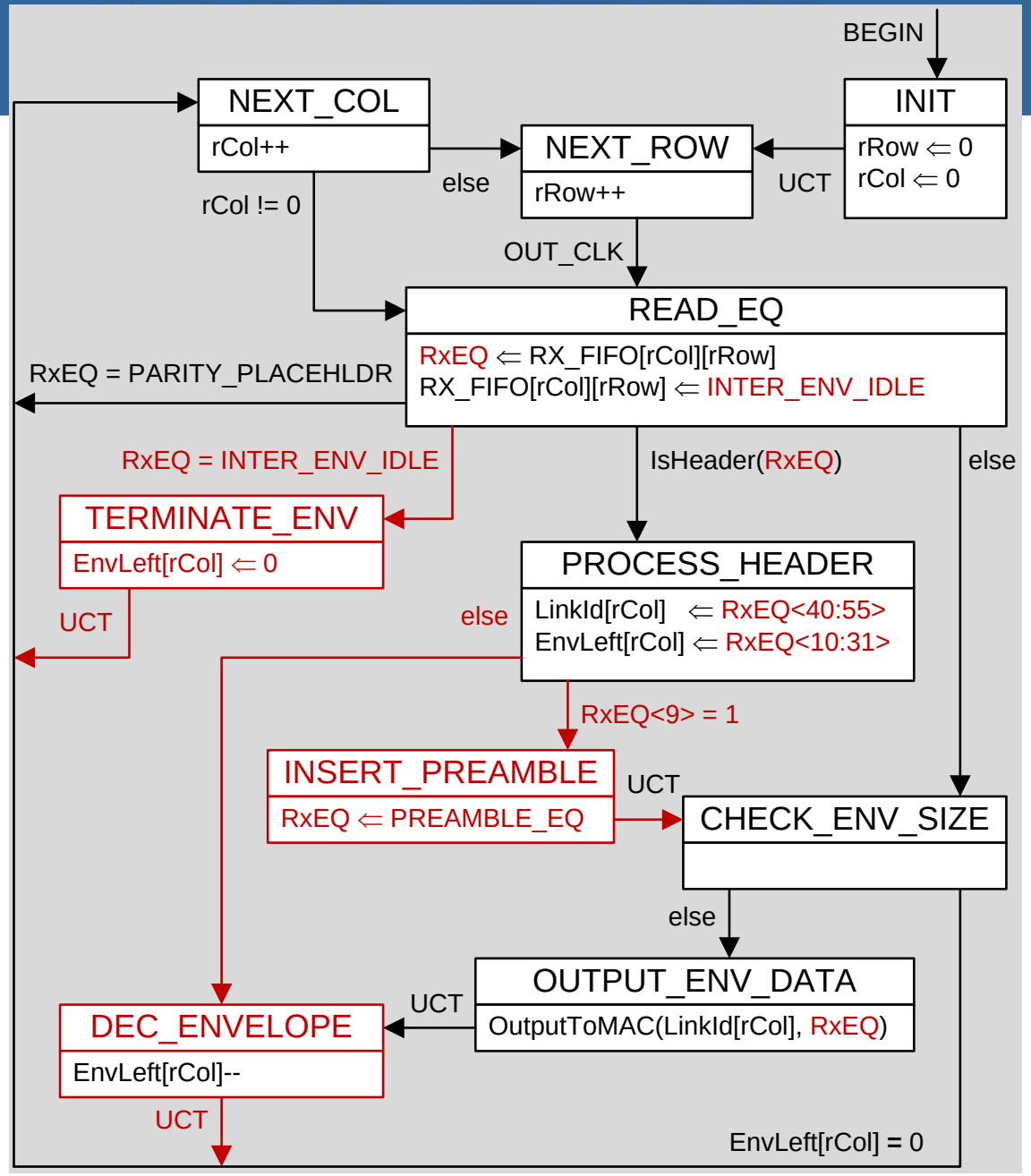


Figure 143–4—OLT MPRS Output Process state diagram

Revised Output Process SD

- Accommodates INTER_ENV_IDLE
- PROCESS_HEADER accommodates Preamble replacement
- EnvLeft decremented on reception of all Envelope Headers



SD Stylistic changes (enter/exit from side of state) are at the Editors discretion

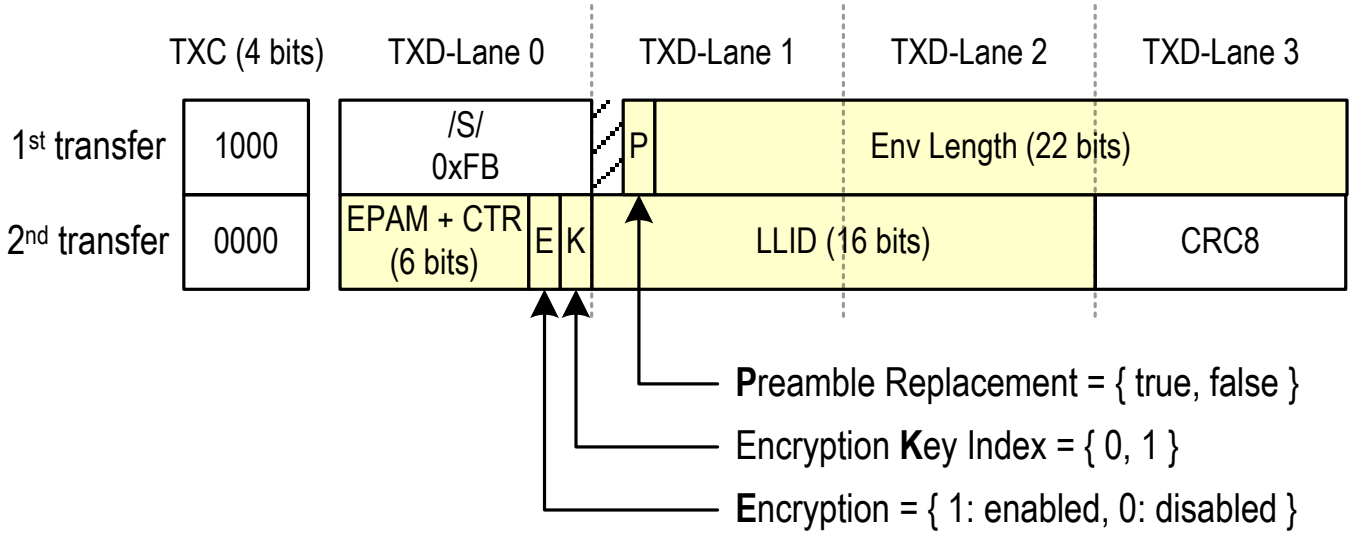
Revised Output Process change summary

- New States:
  - TERMINATE_ENV; closes the current envelope when INTER_ENV_IDLE received (indicates the end of an envelope)
  - INSERT_PREAMBLE; loads RxEQ with a preamble when Envelope Header is flagged as being a preamble replacement
  - DEC_ENVELOPE; split "EnvLeft[rCol] –" out of OUTPUT_ENV_DATA state
- Renamed variable:
  - RxEQ (OutEQ)
- New/Renamed constants:
  - PREAMBLE_EQ; one EQ that is equivalent to a preamble
  - INTER_ENV_IDLE (NO_ENV_CODE)

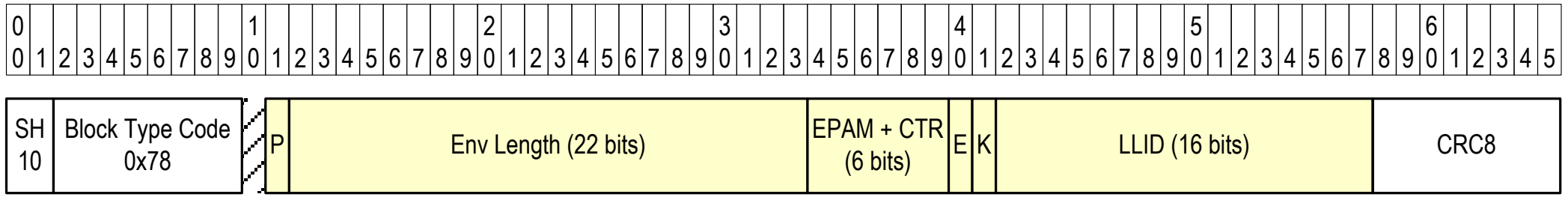
Revised Envelope Header structure

- /S/ - per Cl 49
- 1-bit reserved
- 1-bit Preamble replacement flag
- 22-bit Envelope Length (matches GATE/Grant capability)
- 2-bit reserved for DPoE style Encryption functions (reserved in 802.3ac)
- 6-bit EPAM (also functions as Encryption counter)
- 16-bit LLID
- 8-bit CRC

Envelope Header (two 25XGMII transfers)



66-bit encoding of Envelope Header



Request feedback

- The authors are continuing to work on this
  - Would like to eliminate "double header" (Envelope Header is immediately followed by Preamble Replacement when an Envelope begins with a Preamble)
  - We have some ideas on how to do this without overly complicating MPRS SDs but need to refine these ideas

- Additional input from TF participants is encouraged (currently 3-4 participants on this activity)
  - Send email to Duane and Glen if you would like to be included in the email discussion and occasional call

Thank You