Stream-based FEC proposal

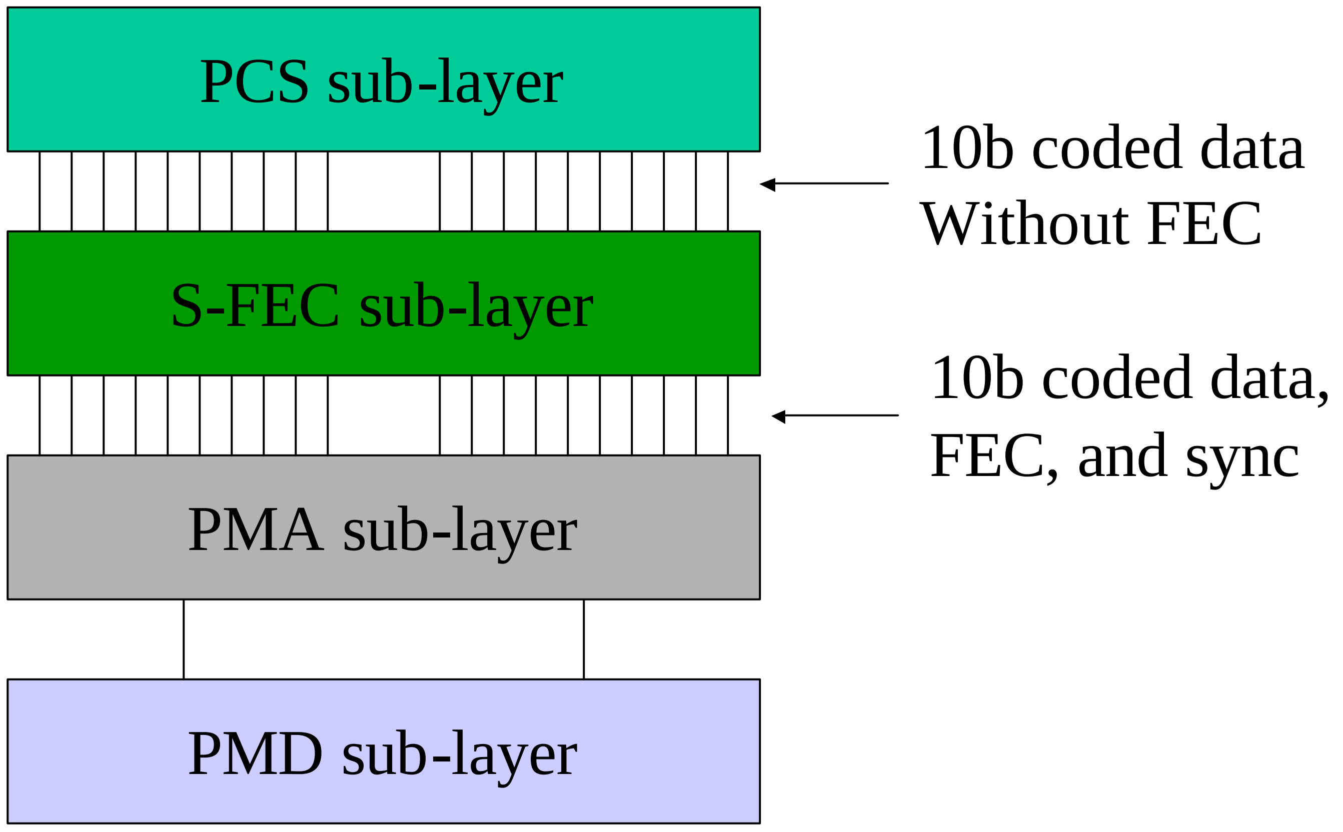
Supporters

Ajay Gummalla, John Limb	Broadcom
Ali Abaye, Masoud Khansari	Centillium
Frank Effenberger	Quantum Bridge
Mark Sankey	Calix
Oren Marmur, Eyal Shraga	Flexlight
Walt Soto	Agere

Stream-based FEC

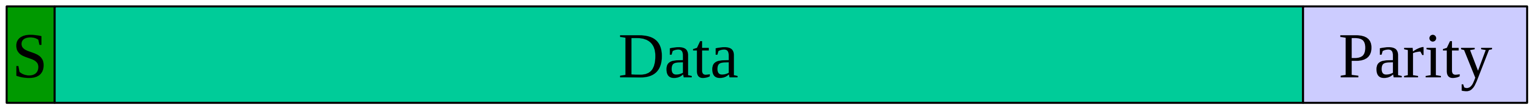
- The FEC process accepts a block of data
  - It doesn’t care about the actual content
  - The data is just ‘bits’ to the FEC
- The FEC computes parity information
- The parity info, plus a FEC synchronization symbol, is inserted into the bit stream
- The receiving side reconstructs the original bit stream, and upper layers are unaware

Basic Stream-FEC proposal



S-FEC encoding algorithm

- PCS data is accepted in 190 symbol blocks
- 1 special FEC sync symbol is added
  - These 191 bytes fit into 1912 bit payload of FEC
- Basic FEC code used is RS(255,239,16)
- 16 bytes of parity are encoded into 16-10b symbols using standard encoder
- Total efficiency is 190/207 = 91.8% fixed



What about running disparity?

- The parity symbols can flip the disparity
- We use two parity special codes
  - P1 – parity correcting: K28.1
  - P2 – parity preserving: K28.7
- These codes aren’t found in operating links
- They can be generated by errors, and this will be accounted for in sync locking time

S-FEC decoding algorithm

- Synchronization keys on sync symbol and delimiter sequence (in the upstream)
  - Error tolerance afforded by implementing hysteresis over several frames (see G.975)
- Data+Parity block is then run through RS decoder
- Resulting corrected data is handed up to PCS layer as if nothing happened

Details of S-FEC data handling

- PMA issues: comma, and disabling it
- Downstream delineation
- Upstream delineation
- Layer model with signals
- Rate adaptation scheme

PMA issues

- Current PMA does two things:
  - Serialization / De-serialization
  - Comma Detection
- Presuming the clocking (jitter) still works, the SerDes will work ok
- In the face of 1e-4 BER, comma detection will not work

Why BERs break commas

- It only takes a few errors to make a comma
  - Certain data codes can be converted into a comma with as few as one bit error (P=1e-4)
  - Even the ordinary comma (K28.5) can be shifted by two bit errors (P=1e-8)
- If the PMA is operating in promiscuous mode (EN_CDET=True), these errors will cause PMA to slip (typically losing data)
  - PCS won’t be able to fix this fault
- Bottom line: PMA Comma detection must be turned off (at least during normal operation)

How we intend to use PMAs

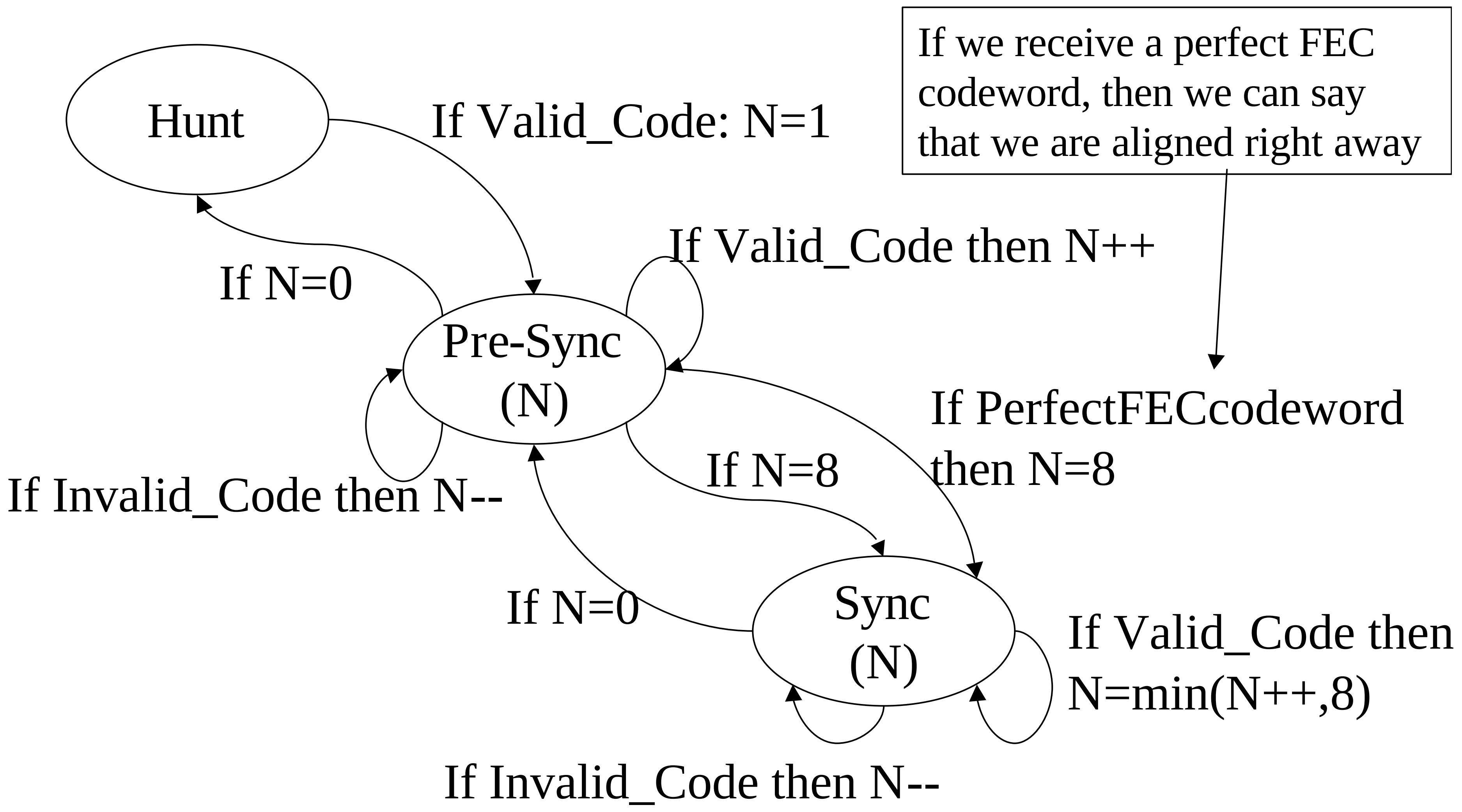
- Current definitions of PMA behavior are too loose to be useful
- As previously argued, comma detection is of dubious value with errors

- Simplest thing is to TURN CDET OFF
  - We assume PMA must support EN_CDET signal, and that we turn CDET off **all the time**

S-FEC downstream delineation

- Assumptions: PMA CDET turned off.
- FEC layer performs both code and block delineation using same framing code
- State machine has three states
  - Hunt: No code alignment found
  - Pre-Sync: Have a suspect code alignment
  - Sync: Confirmed code alignment, transfer data

Downstream State Machine



If we receive a perfect FEC codeword, then we can say that we are aligned right away
Hunt
If Valid_Code: N=1
If Valid_Code then N++
If N=0
Pre-Sync (N)
If PerfectFECcodeword then N=8
If N=8
If Invalid_Code then N--
If N=0
Sync (N)
If Valid_Code then N=min(N++,8)
If Invalid_Code then N--

Implementation of Hunt State

- FEC runs at code word rate (125 MHz)
- Parallel implementation
  - Ten bit alignments are considered every clock
  - Matching is done on all ten candidates
  - A hit selects that phase
- Serial implementation
  - In hunt state, alignment is slipped one bit a time
  - Eventually, all alignments (bit and word) are tested Reason: The FEC block is 208*10=13*5*2^5 bits long. The slip cycle is 11 bits long. The block and the cycle do not share common factors; therefore, all alignments will occur, and within their LCM (11 FEC blocks)

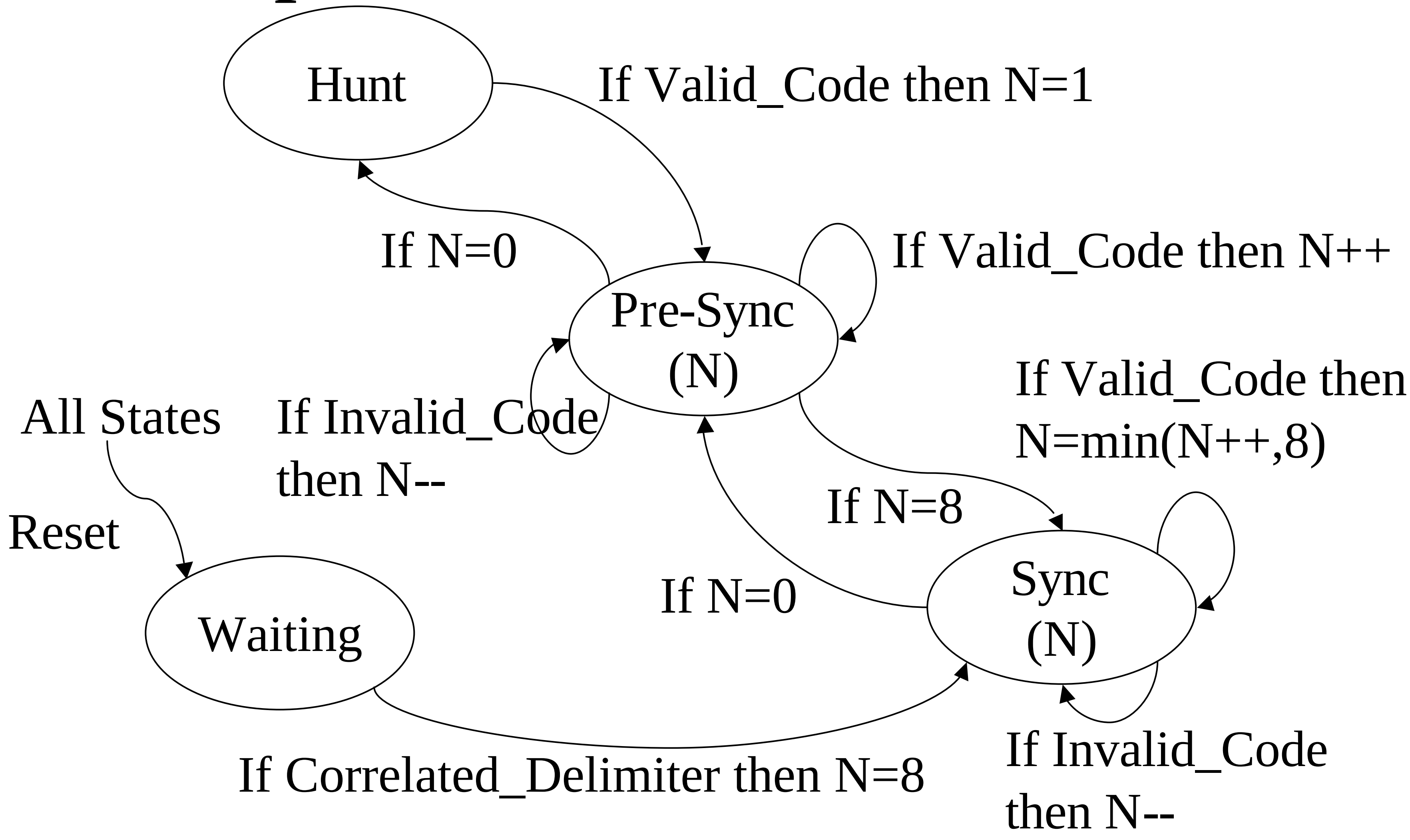
How long does sync take?

- The parallel implementation, using no FEC side information
  - Taverage = 13.3 us
  - Time to 1e-12 probability = 29.8 us
- The parallel implementation, using FEC corrections on the sync symbol, and taking a perfect FEC codeword as indication of sync
  - Taverage = 3.7 us
  - Time to 1e-12 probability = 19.9 us

S-FEC Upstream delineation

- Assumptions: PMA CDET turned off.
- FEC accepts code groups as they come
- Beginning of burst: search for Delimiter
  - FEC enters “Waiting” state when Reset, or LOS signal is received, or inferred from PMA input
  - Search is bit by bit, using word-parallel logic
- Once Delimiter found, transition into SYNC state, continue as in downstream

Upstream State Machine



Choice of Delimiter

- If we choose a delimiter of length N (bits)
  - With 50% duty cycle
  - Maximal distance from all-zeroes and preamble pattern*
- The code will have a minimum Hamming distance, D
  - With a 101010 preamble, then the best D = int(N/2-1)
- If burst loss rate is R, then:

$$R = BER^{\text{int}(D/2+1/2)} * \binom{N}{\text{int}(D/2+1/2)}$$

Choice of Preamble

- Best pattern is 1010…
  - Maximum clock density
  - Minimum baseline wander and clock jitter
  - Allows best delimiter performance
- But, good old Ethernet uses I2…
  - This will work also
  - Clock recovery will pay a small price
  - Delimiter performance somewhat degraded

Preamble = 101010…
Delimiters and Error Rates

Delimiter Length (bits)	Error Rate (Errors/Burst)	MTBE (years)
20	1.5E-16	2E+02
40	8.5E-32	4E+17
60	5.3E-47	6E+32

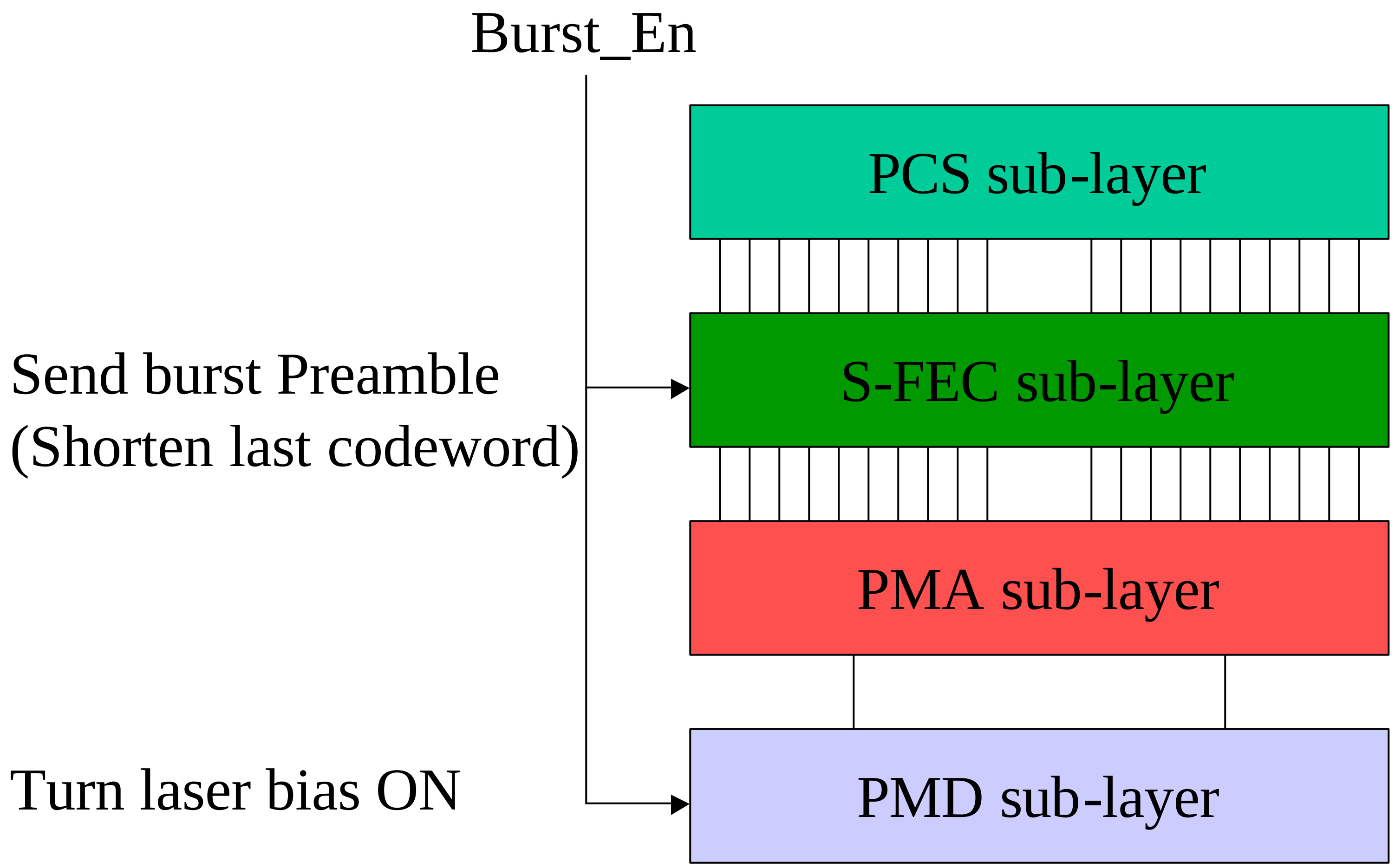
The above assumes:
Raw BER = 1e-4
Burst rate = 1e6 burst/second
Errors normally distributed
Preamble data pattern = 1010101010

Preamble = I2
Delimiters and Error Rates

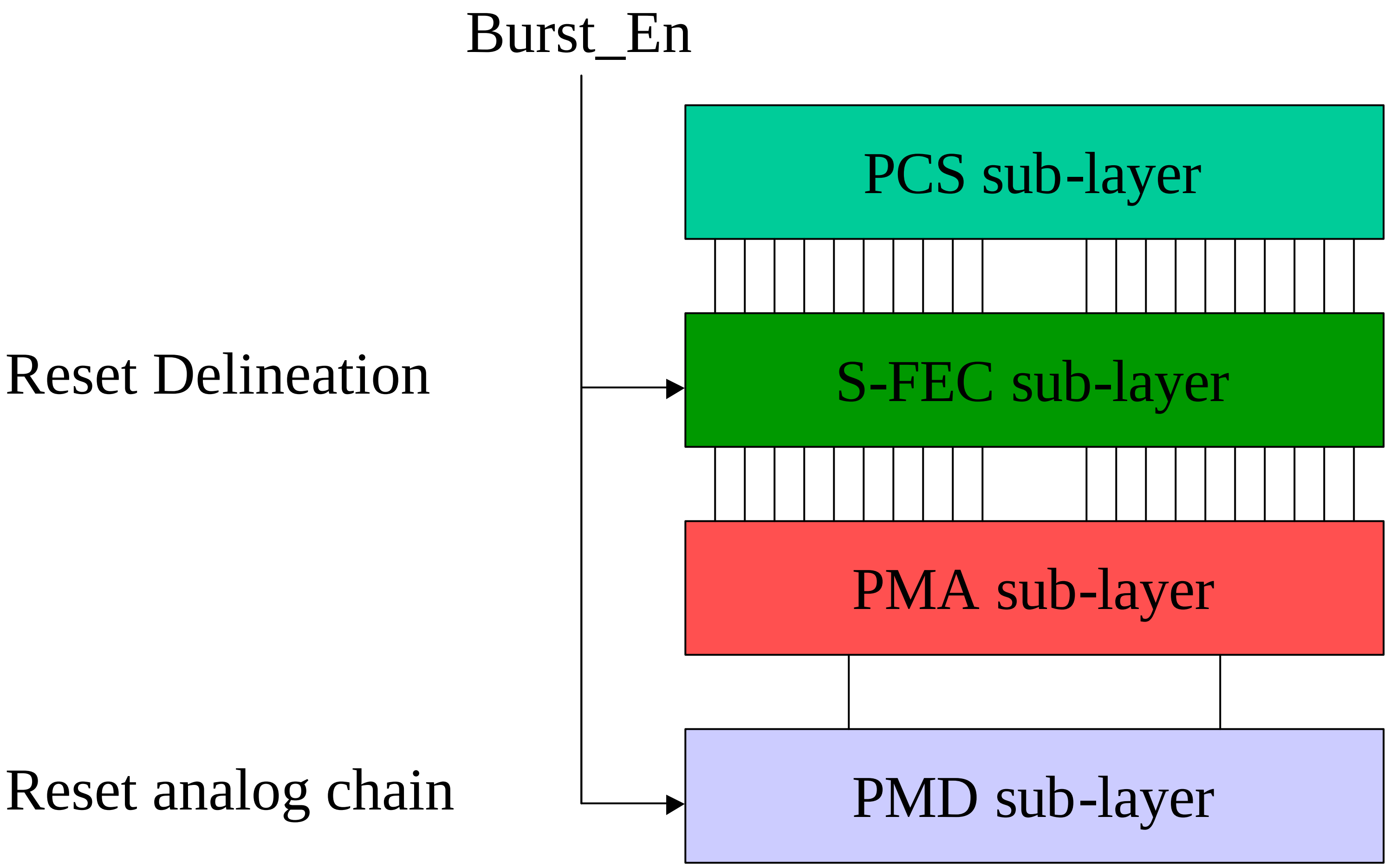
Delimiter Length (bits)	Error Rate (Errors/Burst)	MTBE (years)
20	4.8E-13	7E-02
40	7.7E-25	4E+10
60	1.4E-36	2E+22

The above assumes:
Raw BER = 1e-4
Burst rate = 1e6 burst/second
Errors normally distributed
Preamble data pattern = 0011111010 1001000101

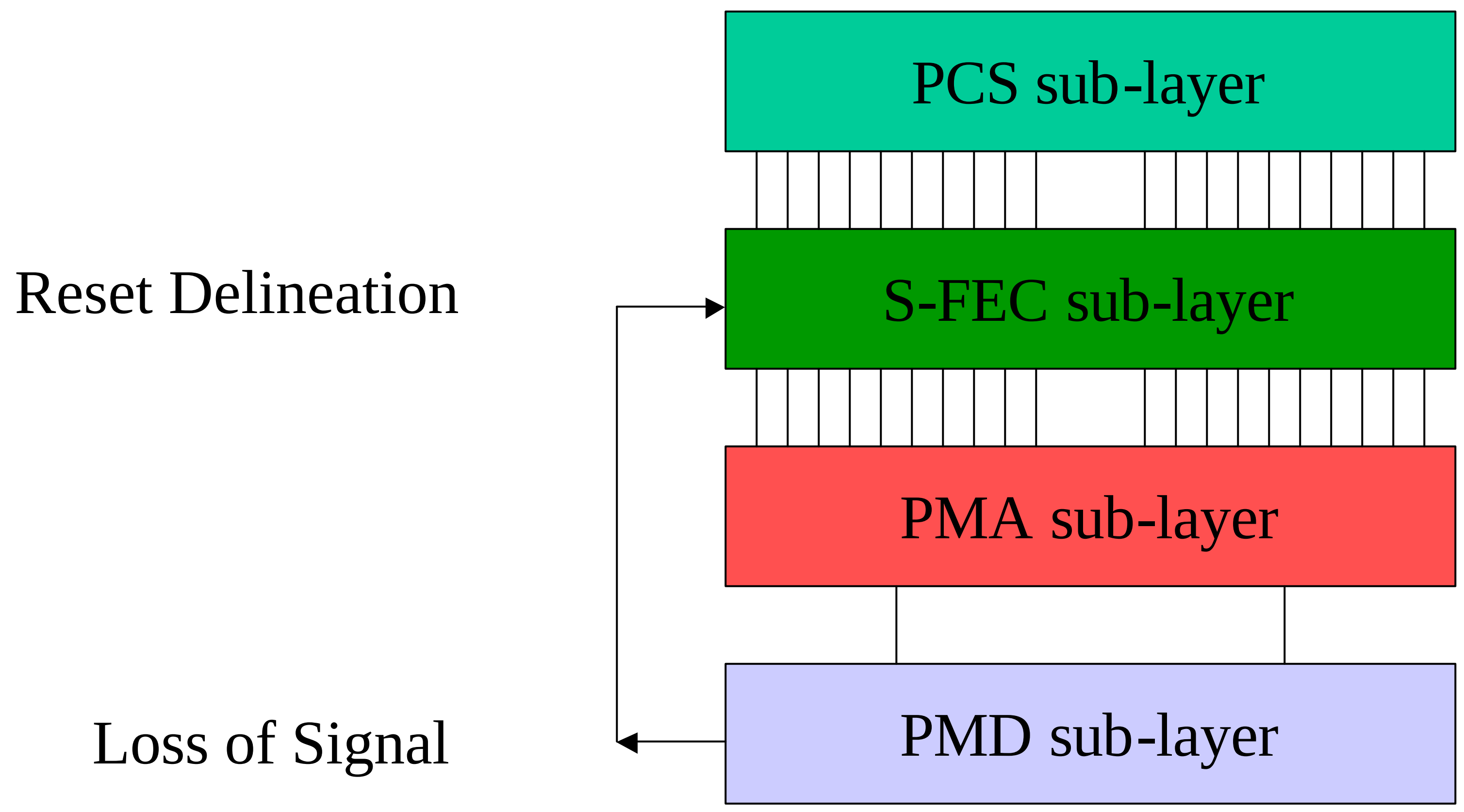
Layer Model (ONT)



Layer Model (OLT w/reset)

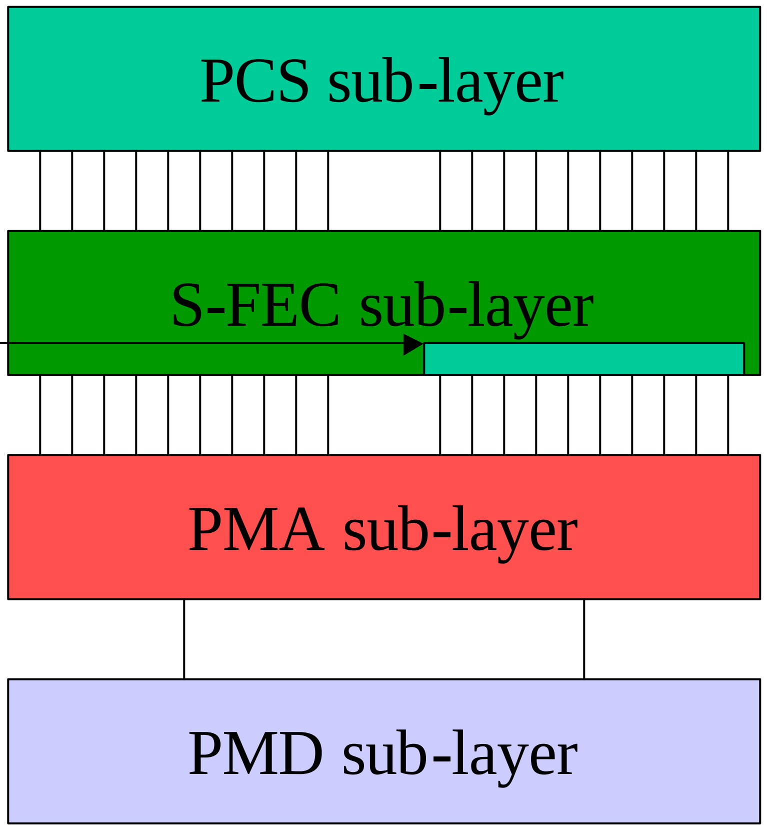


Layer Model (OLT w/fast LOS)



Layer Model (OLT w/inferred LOS)

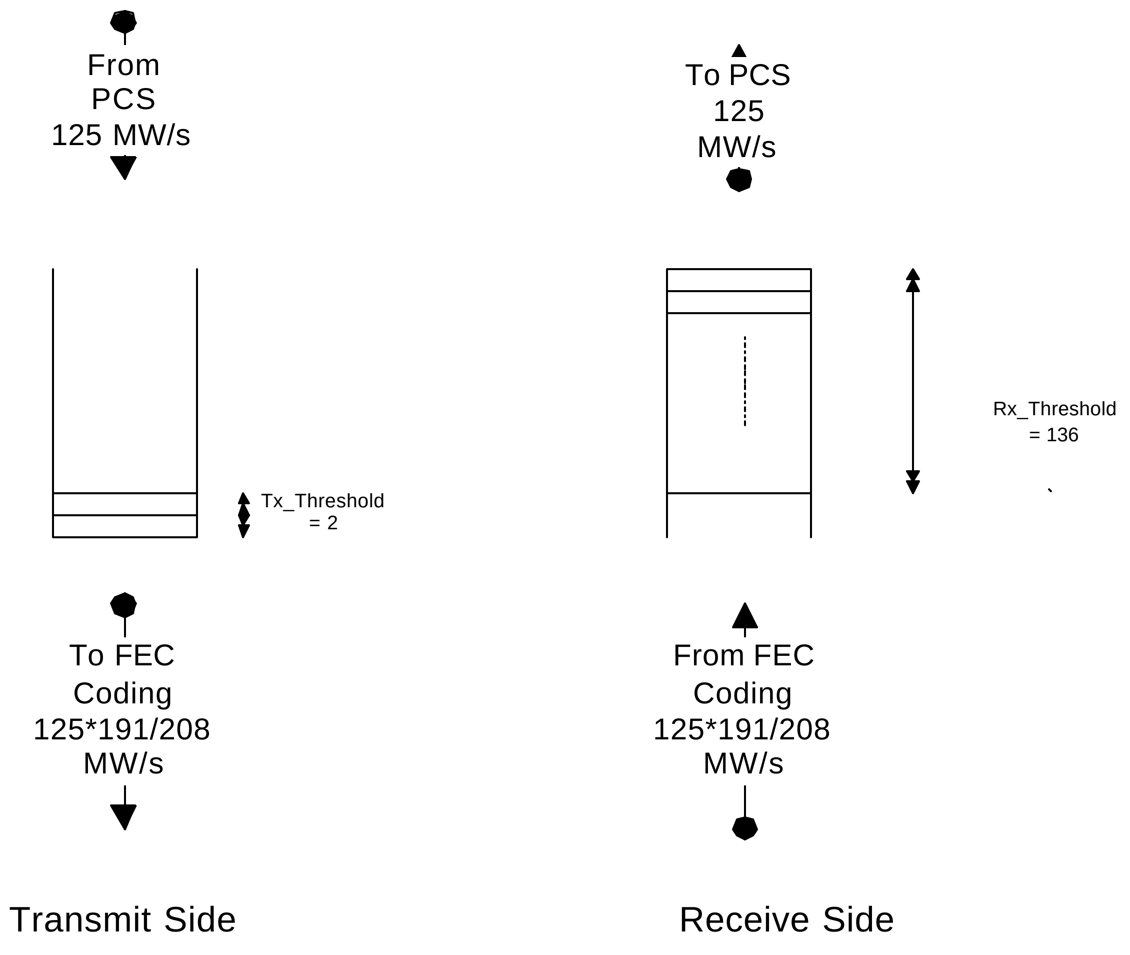
Loss of Signal inferred by lack of valid data at PMA input for



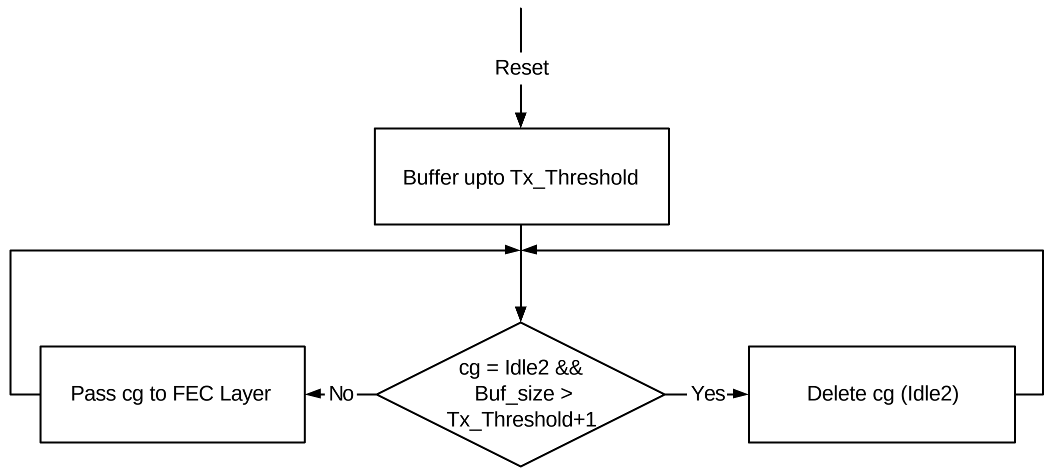
Rate Adaptation

- Adjust the MAC rate to the FEC rate by adding and deleting Idle2
- Self pacing algorithm
- No additional jitter added

Basic Concept



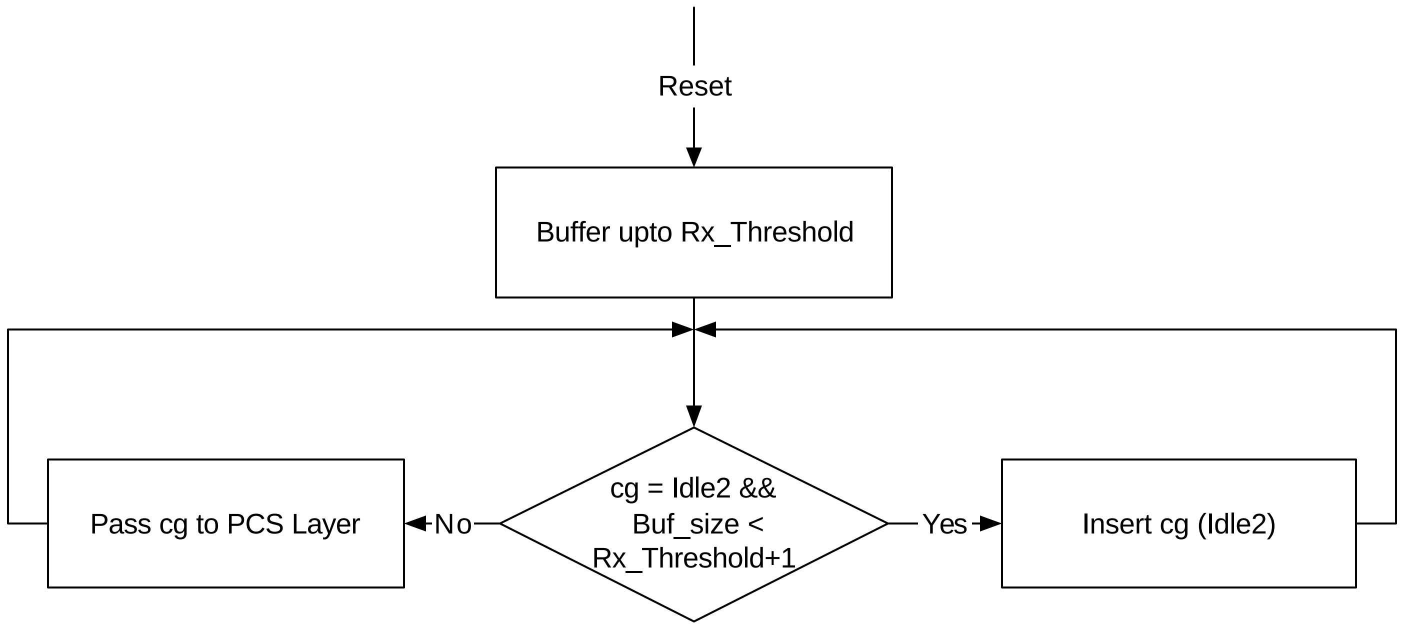
Transmit State Machine



Transmit Rate Adaptation State Machine

Operates at 125*191/208 MHz clock

Receive State Machine



Receive Rate Adaptation State Machine

Operates at 125 MHz clock

Comments

- This does not mean two clock domains
- This algorithm can be modified to operate the Tx State machine at 125 MHz clock