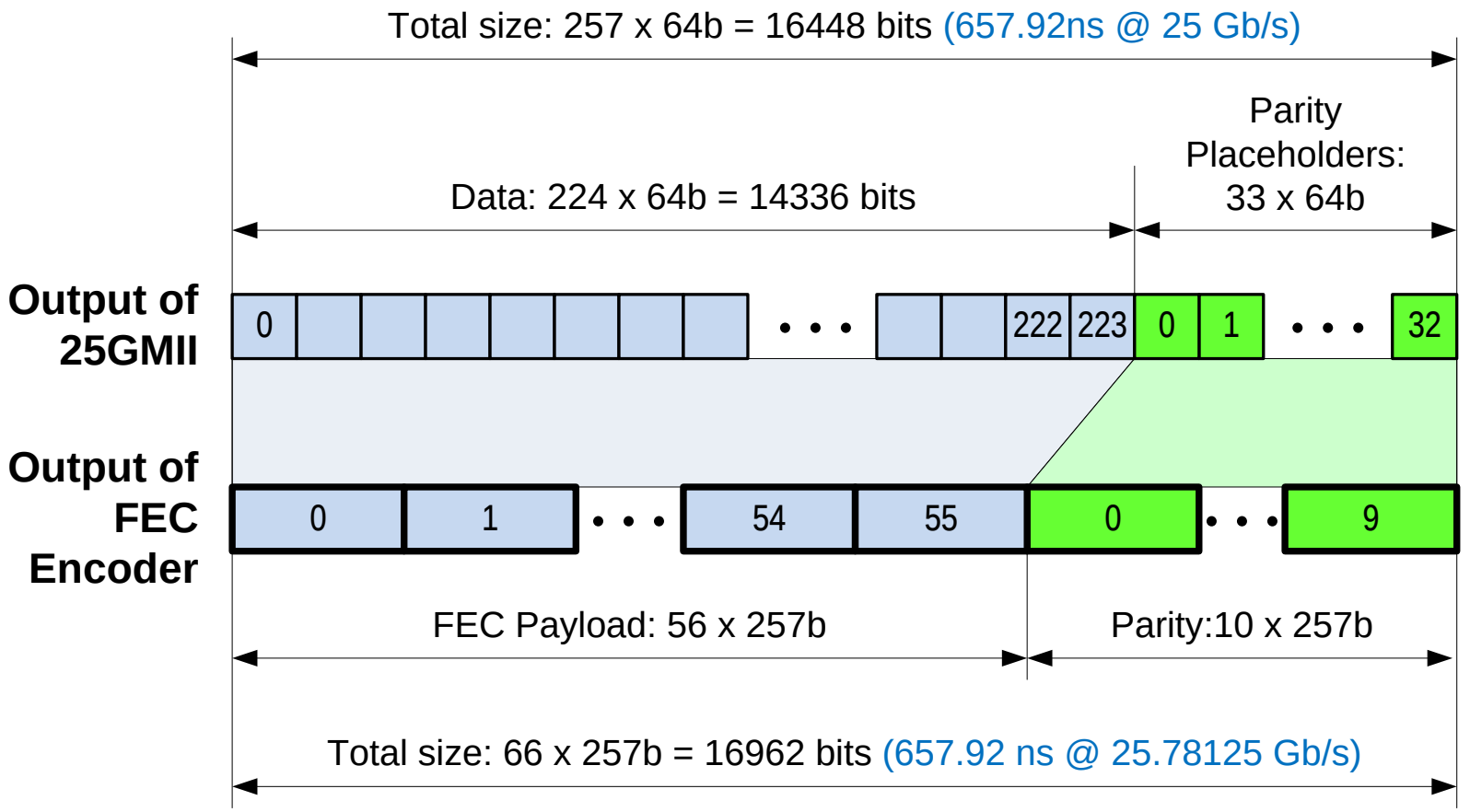
FEC Codeword Format and Alignment Mechanism

Glen Kramer

Broadcom

FEC Codeword Format

- In Chicago, we have adopted the following FEC codeword format (see motion #6)
- Adds ten 257b blocks of overhead per 56 blocks of payload
  - 2560 bits to carry LDPC parity code
  - 10 bits are left for FEC Codeword Delimiter (CD)



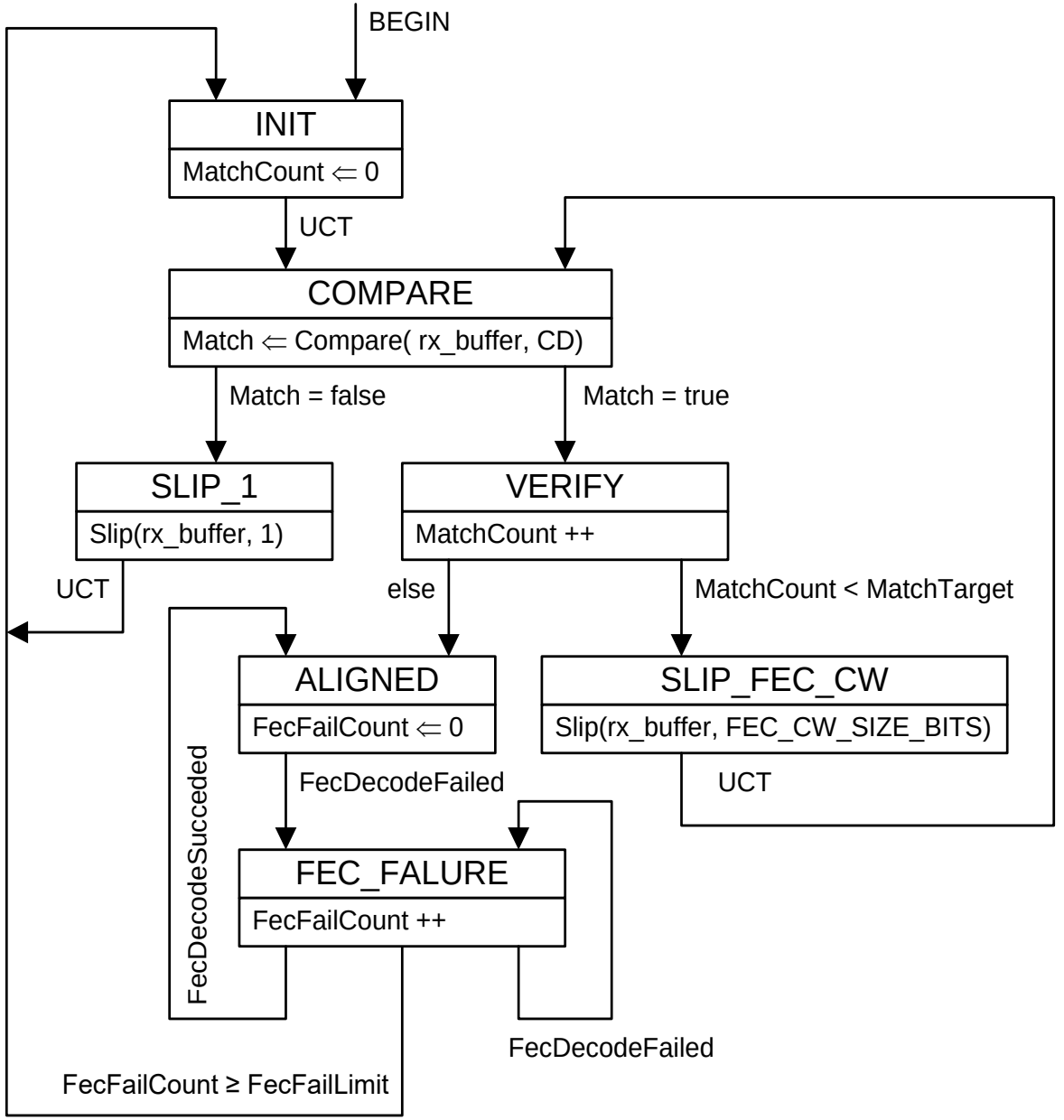
- An action item from the meeting was to verify that 10 bits are sufficient for an ONU to perform a a fast and reliable FEC codeword lock

ONU FEC alignment mechanism

- Downstream never needs to shorten the codewords. Every codeword has a fixed size of 66 257b blocks (payload + parity)
- After reboot, the ONU needs to align to FEC codewords
  - Direct step from bit alignment to FEC CW alignment
  - 257b blocks by themselves are not alignable; additional alignment markers would be required.

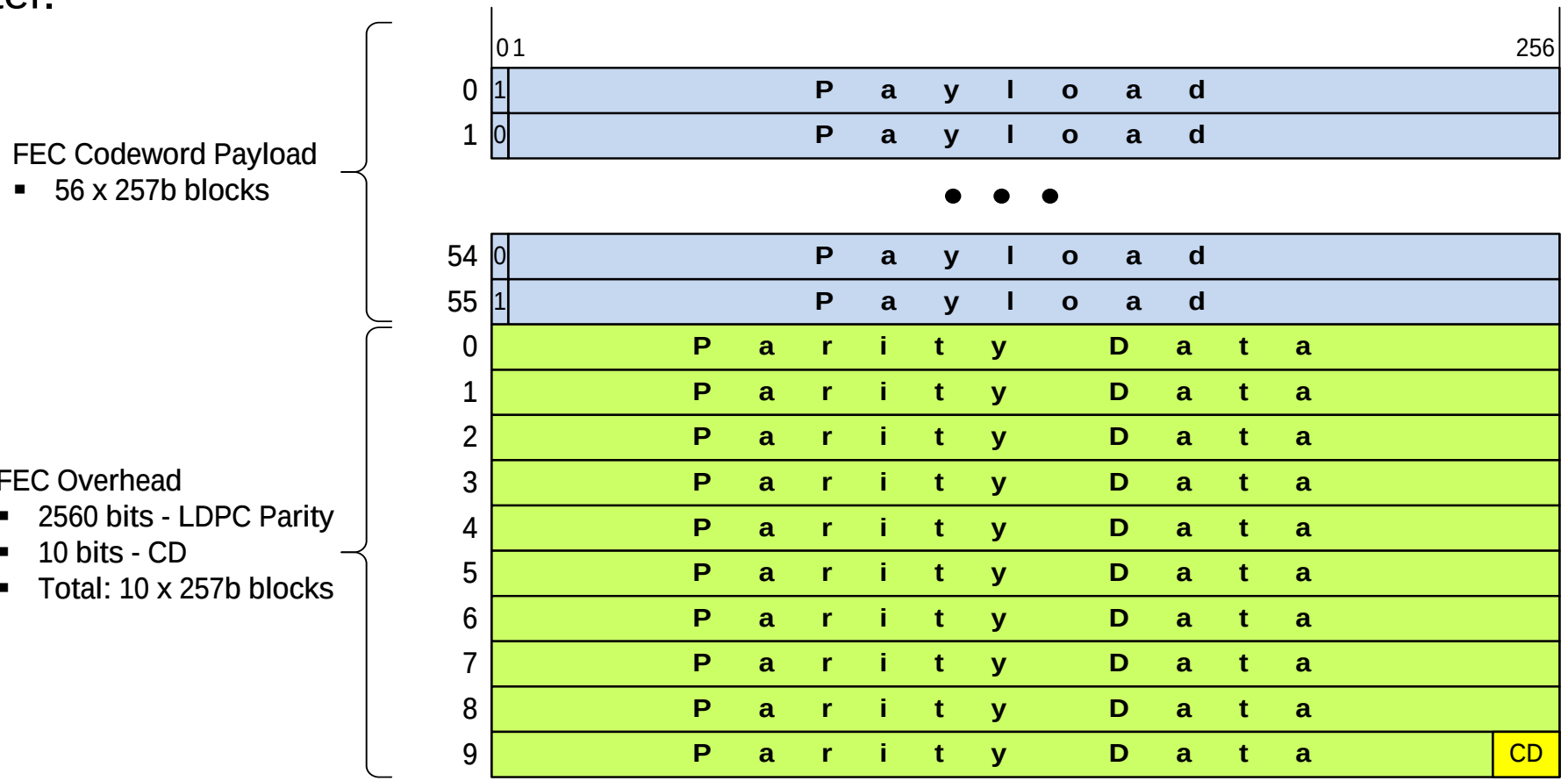
To do the alignment, the ONU does the following:

1. Hunt for CD (10-bits) in the received bit stream
   a) If CD is not matched, *MatchCount = 0*, slip incoming stream by 1 bit, go to (1).
   b) If CD is matched, *MatchCount* ++, slip incoming stream by one full FEC codeword (66 × 257 = 16962 bits), go to (1)
2. When *MatchCount* reaches a *MatchTarget* threshold (5?), declare the alignment
3. If FEC decoder consistently indicates decoding failure for *FecFailLimit* (3?) consecutive FEC CWs, set *MatchCount = 0*, go to 1 to realign.



FEC Codeword Format

- Performance of the alignment algorithm does not depend on whether the 10 bits comprising the CD are spread apart or grouped together.
- However, it does make a big difference for the implementation (i.e., 10-bit-wide shift register vs. 2560-bit-wide shift register.
- Proposed FEC structure with all CD bits grouped at the end.



Reliability of Alignment Mechanism

- The alignment SD will often pursue false leads before stumbling on the true delimiter. That is OK and expected.
  - $1/2^{10}$ chance of finding any pattern (including the CD) in the scrambled data
- Because the alignment mechanism relies on the feedback from the FEC Decoder, probability of a persistent wrong alignment is zero. A wrong alignment is detected quickly and it forces the ONU to try again. The ONU will always get proper alignment eventually. The question is how long that would take?
- The efficiency of FEC alignment depends on the choice of
  - *PD matching Hamming threshold*
  - *MatchTarget*
  - *FecFailLimit*
- We should find the best combination of these values to minimize the alignment time

Simulation Parameters

- ***10 million*** alignment attempts for each configuration
- ***Input BER* = 0.01**
- ***Hamming threshold* = varies {0, 1}**
  (in kramer_3ca_xxxx, we saw that H=2 results in increased alignment time due to large number of false matches, so it was removed from further consideration)
- ***MatchTarget* = varies {5, 4, 3}**
- ***Codeword Delimiter (CD)* = 0x3ca** (constant)
- ***FecFailLimit* = 3**
  - Assume the time to detect loss of alignment (persistent FEC decode failure) = 6.58µs (this includes Rx time for 3 failed FEC CWs and FEC decoder delay)

Test #1: Different Match Hamming Thresholds

Test configuration:

- Match Target = 5
- Hamming threshold = {1, 0}

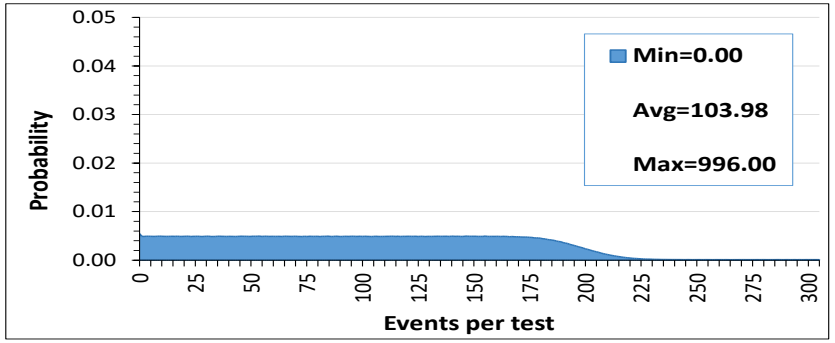
	Hamming Threshold = 1 (1-1-1-1-1)			Hamming Threshold = 0 (0-0-0-0-0)		
	Min	Avg	Max	Min	Avg	Max
False Leads	0	103.98	996	0	19.26	320
Missing True CD	0	0.02	4	0	0.65	18
Time to Alignment (μs)	2.63	71.41	665.37	2.63	16.84	248.06

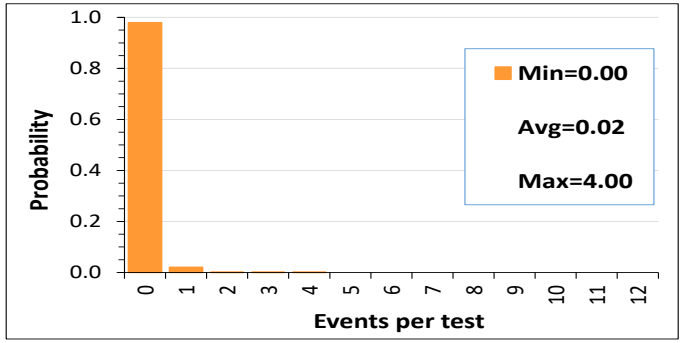
False Leads

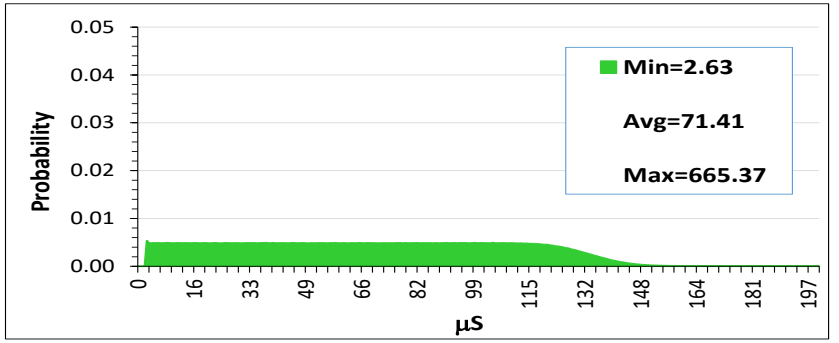
Missing True CD

Time to Alignment (μs)

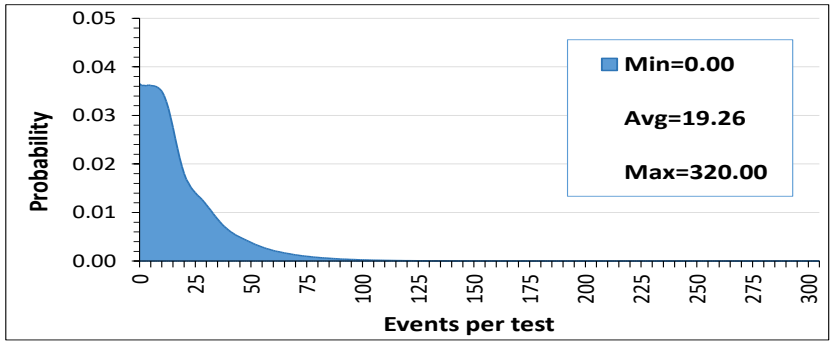
1-1-1-1-1

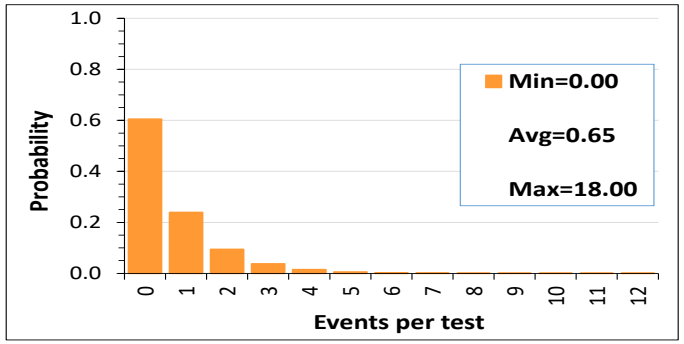


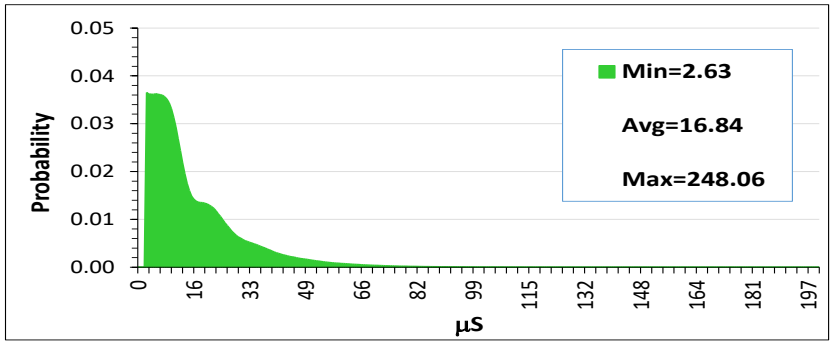




0-0-0-0-0







Test #2: Different Match Targets

Test configuration:

- Match Target = {4, 3}
- Hamming threshold = 0

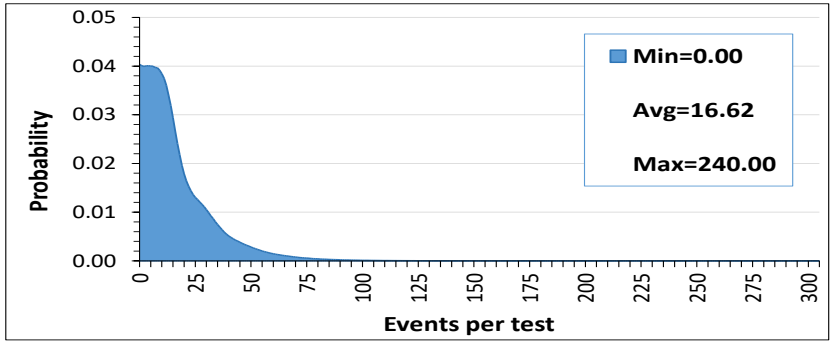
	Match Target = 4 (0-0-0-0)			Match Target = 3 (0-0-0)		
	Min	Avg	Max	Min	Avg	Max
False Leads	0	16.62	240	0	14.23	229
Missing True CD	0	0.49	14	0	0.35	12
Time to Alignment (μs)	1.97	14.01	186.66	1.32	11.46	168.37

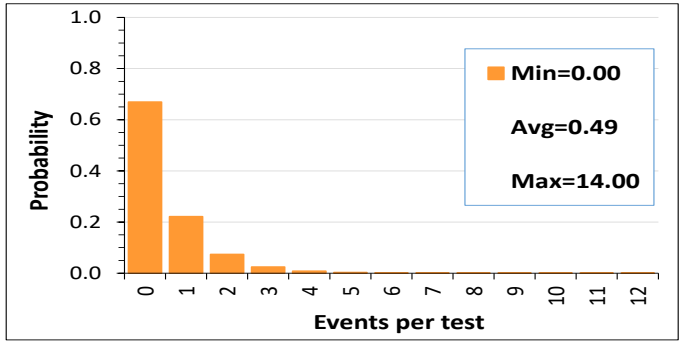
False Leads

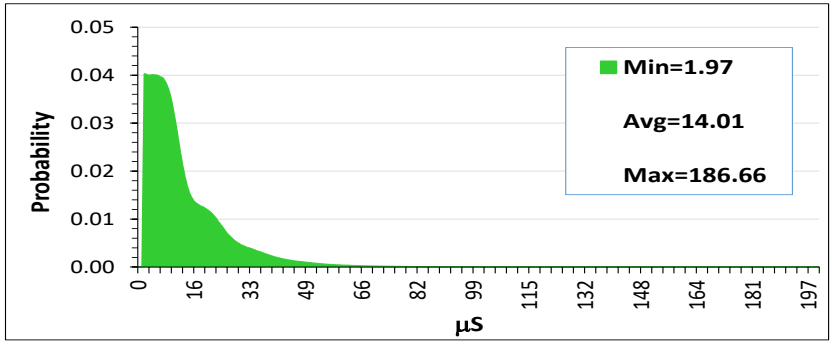
Missing True CD

Time to Alignment (μs)

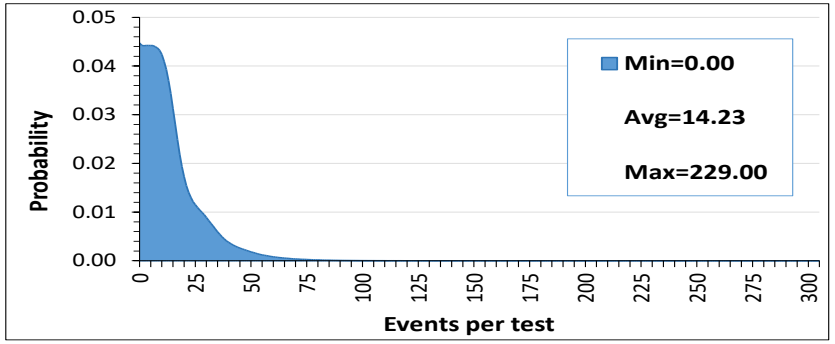
0-0-0-0

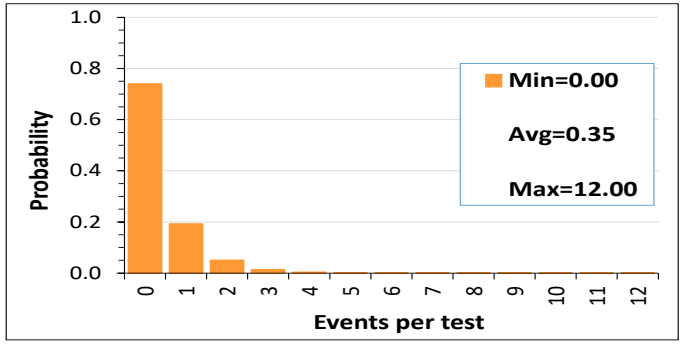


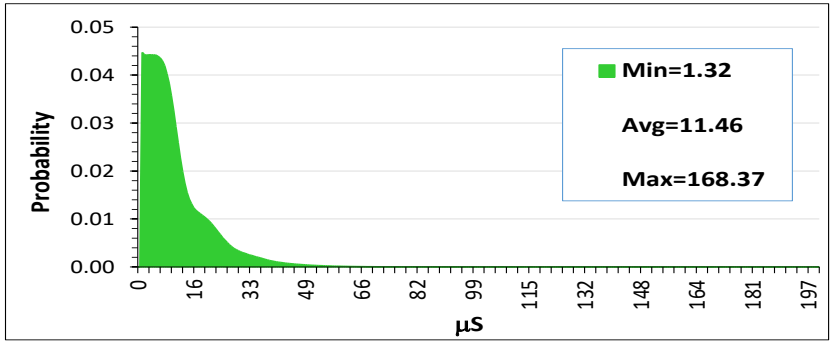




0-0-0







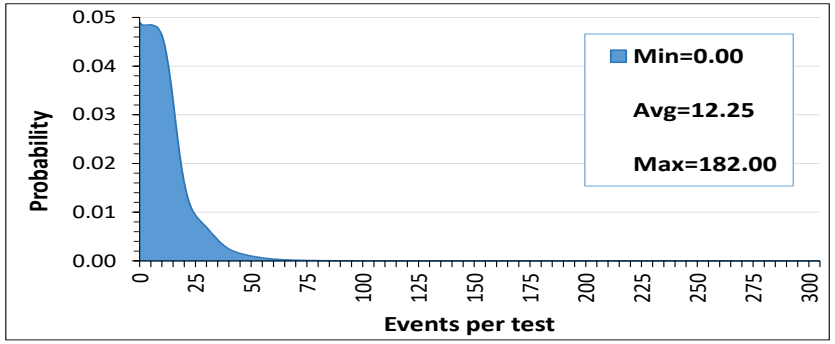
Test #3: Progressive Hamming Threshold (0, 0, 1, [1])

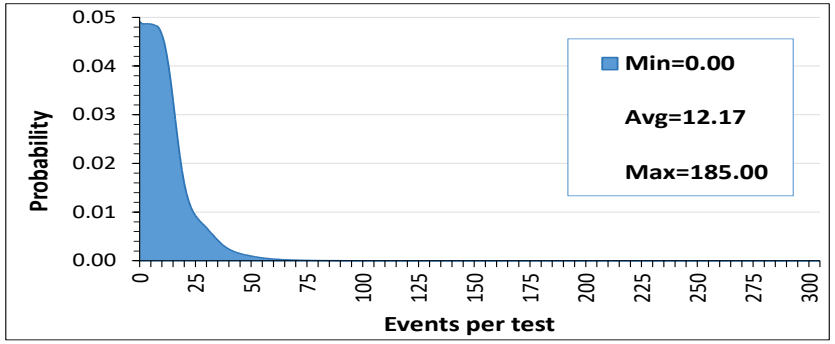
Test configuration:

- Match Target = {4, 3}
- Hamming threshold = 0, 0, 1, …

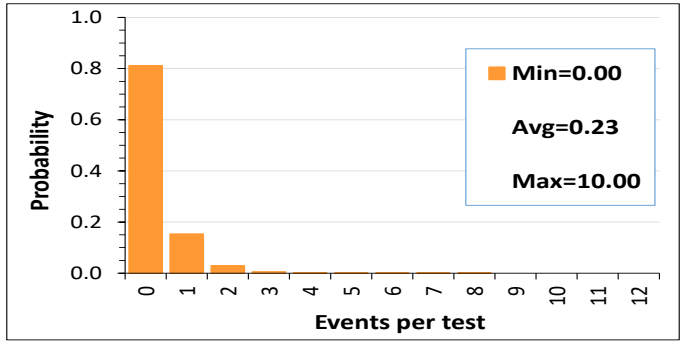
	Match Target = 4 (0-0-1-1)			Match Target = 3 (0-0-1)		
	Min	Avg	Max	Min	Avg	Max
False Leads	0	12.25	182	0	12.17	185
Missing True CD	0	0.23	10	0	0.23	11
Time to Alignment (µs)	1.97	10.60	132.88	1.32	9.87	134.58

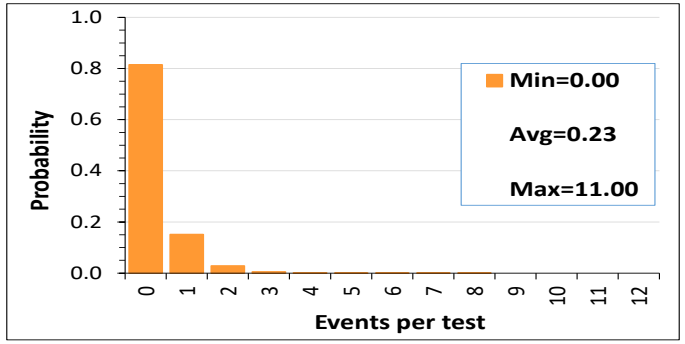
False Leads



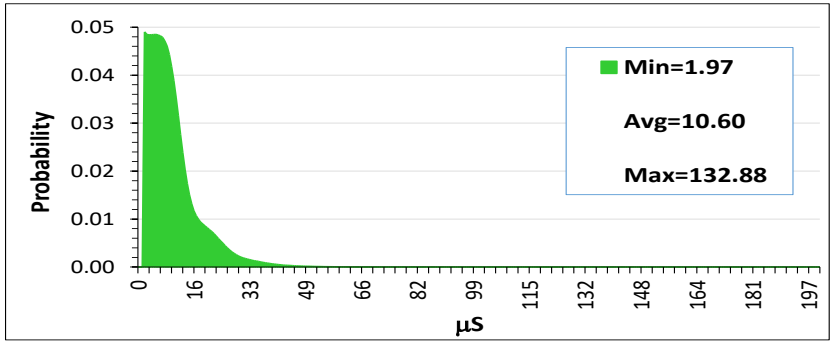


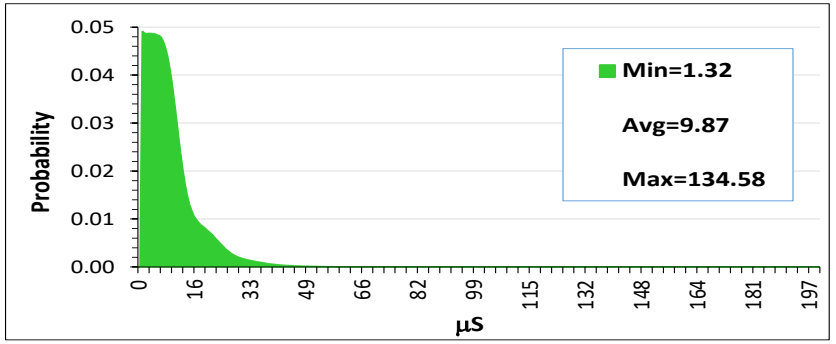
Missing True CD





Time to Alignment (µs)

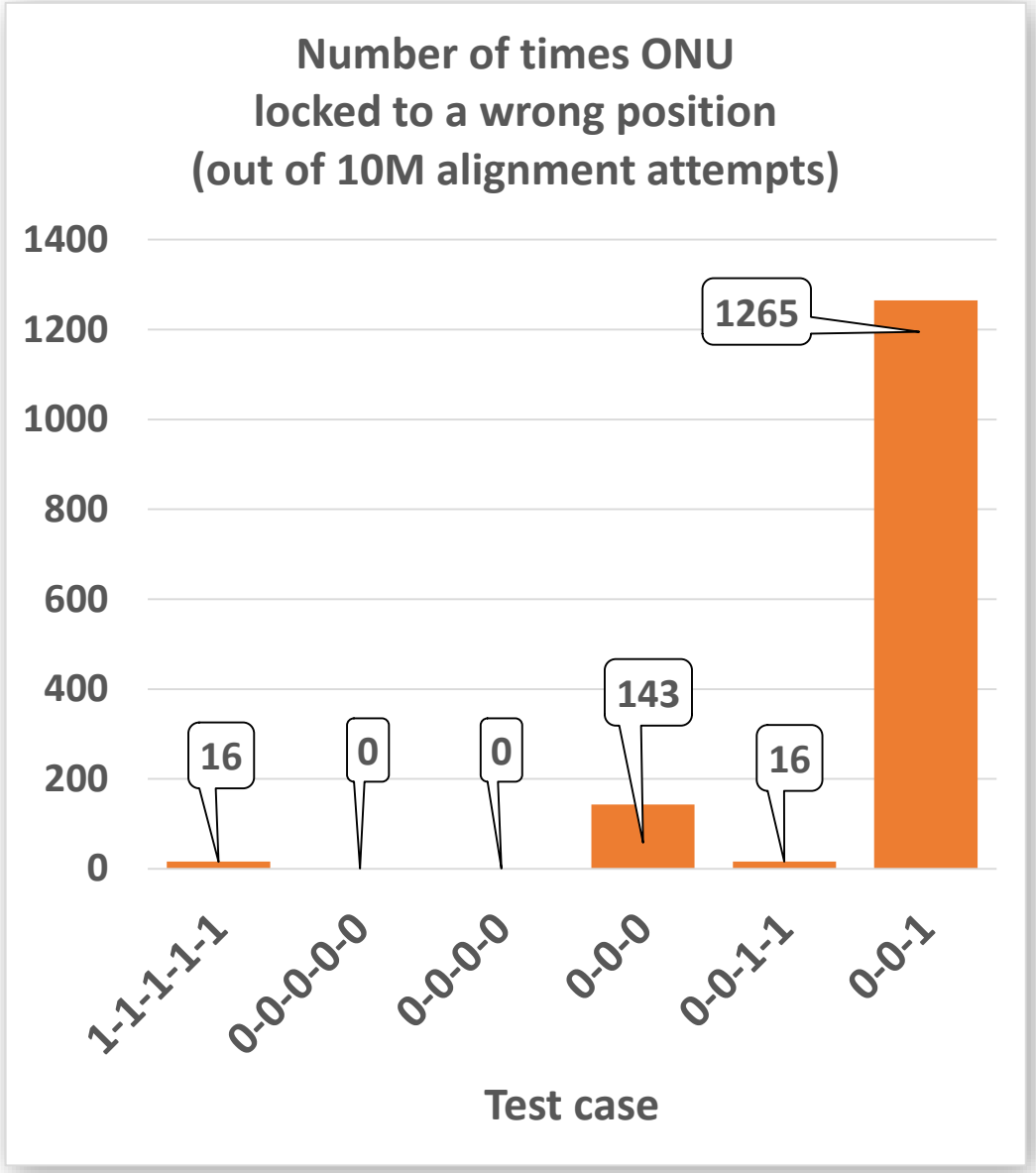




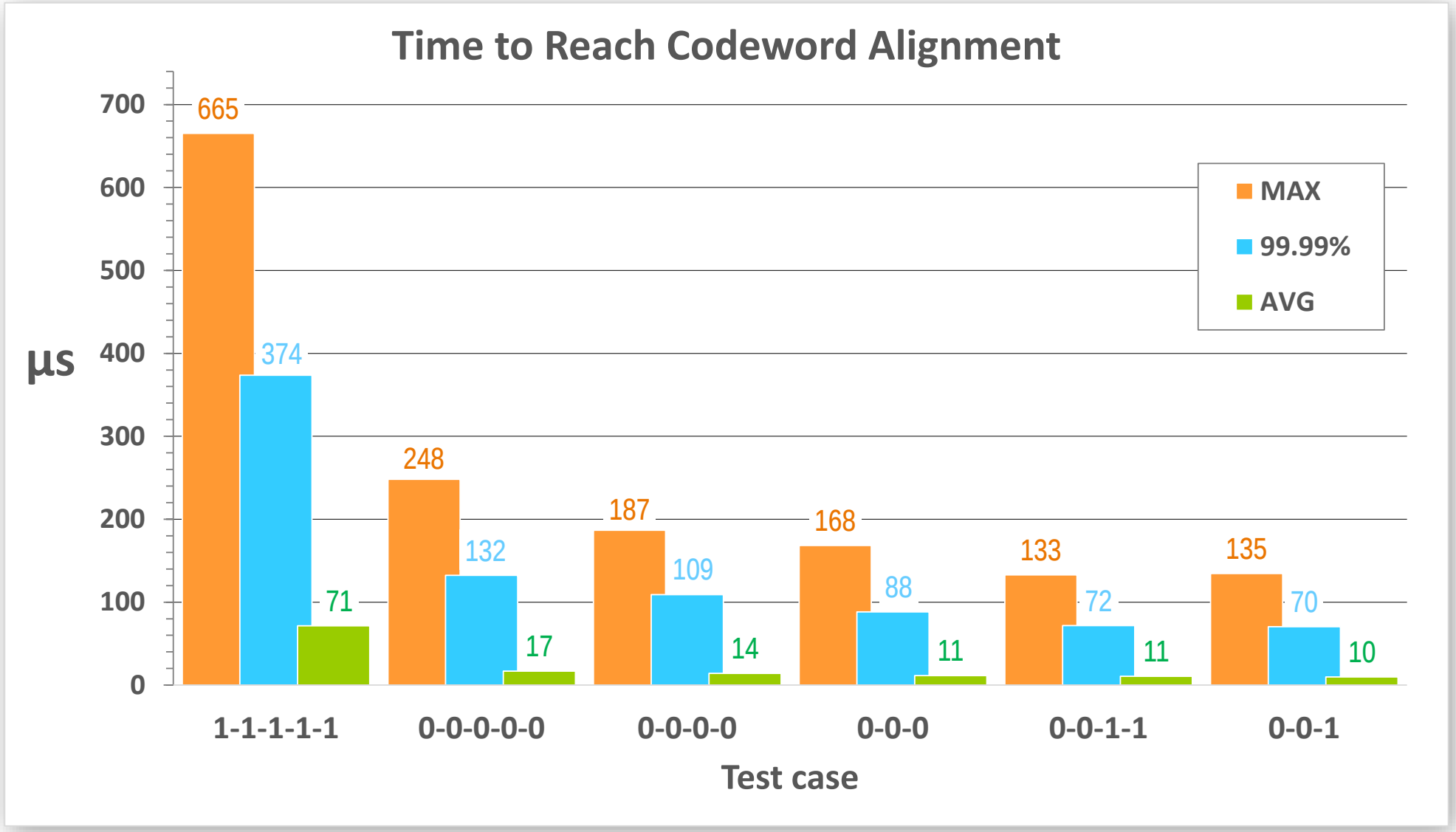
Wrong Alignment

- Some test scenarios encountered locking to a wrong position. In this case, after detecting 3 continuous FEC decoding failures, the ONU would repeat the alignment procedure.

- The time to detect misalignment (6.58 μs) and to repeat the alignment procedure is already included in the Time2Alignment measurements presented in pervious slides.



Average and Maximum Times to Reach True Codeword Alignment



Summary

- Excluding the first test case (Match Target = 5 with Hamming Threshold = 1), all configurations provided comparable results.
- It is impossible to determine externally the precise alignment algorithm employed by the ONU vendor.
- **Recommendation**
  - As the default model behavior, specify the alignment process that looks for four matches of the CD value with Hamming threshold zero (Test case 0-0-0-0)
  - Leave further improvements to ONU implementations