Burst Delimiter Options

BROADCOM

Glen Kramer

BURST DELIMITER and SP in 802.3av

BURST_DELIMITER

TYPE: 66 bit unsigned

A 66 bit value used to find the beginning of the first FEC codeword in the upstream burst.

Value: binary 01 followed by 0x 6B F8 D8 12 D8 58 E4 AB (transmission bit sequence: 01 1101 0110 0001 1111 0001 1011 0100 1000 0001 1011 0001 1010 0010 0111 1101 0101)

SP

Type: 66 bit unsigned

A 66 bit value used to for the burst mode synchronization pattern.

Value: binary 10 followed by 0x BF 40 18 E5 C5 49 BB 59 (transmission bit sequence 10 1111 1101 0000 0010 0001 1000 1010 0111 1010 0011 1001 0010 1101 1101 1001 1010)

- SP and BD values were selected for these special properties:
  1) Perfect balance (33/33)
  2) Limited Run-length (≤ 6)
  3) High hamming distance between SP and BD as well as any offset with itself

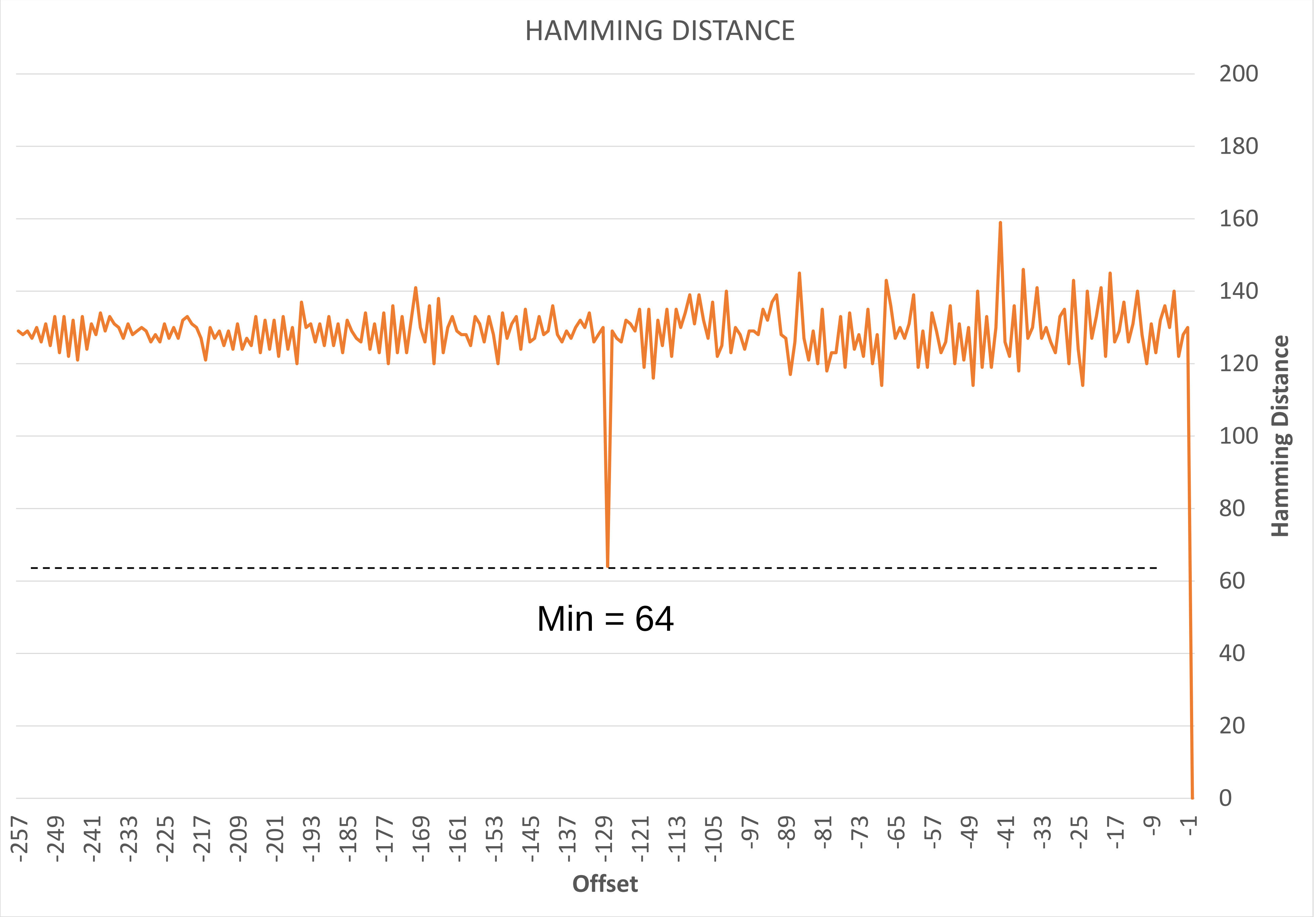
- Probably a good idea to construct super-delimiter (257 bits) from existing SP and BD

Attempt #1

- Use SP and BD as defined in 802.3av
- Assemble the super-delimiter as

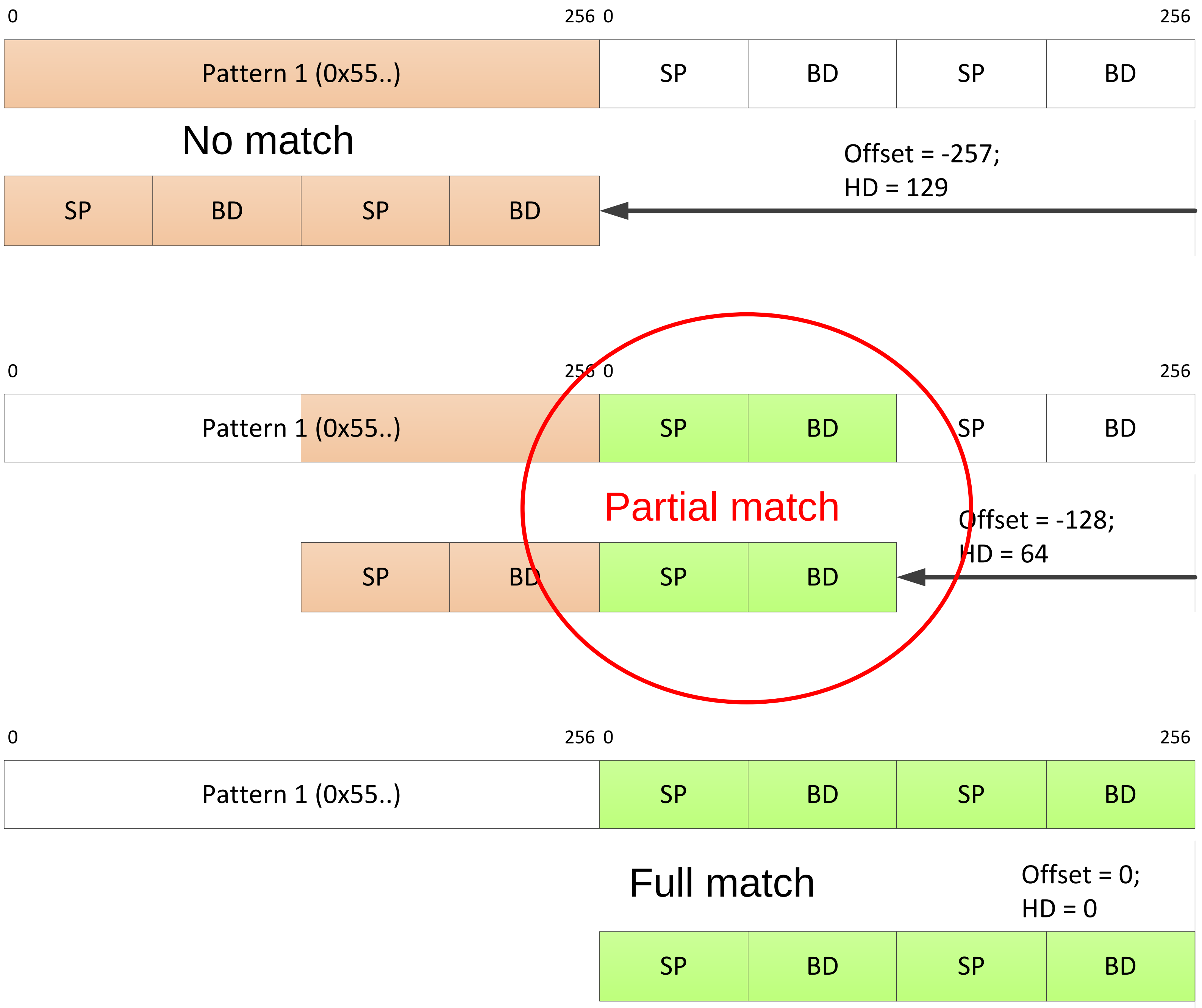
  ‘1’ +
  SP[0…63] +
  BD[0…63] +
  SP[0…63] +
  BD[0…63]

  (i.e., 64 LSB of SP and BD repeated twice)



Problem @ offset = -128

- Periodic delimiter has a partial match at each offset = N×Period

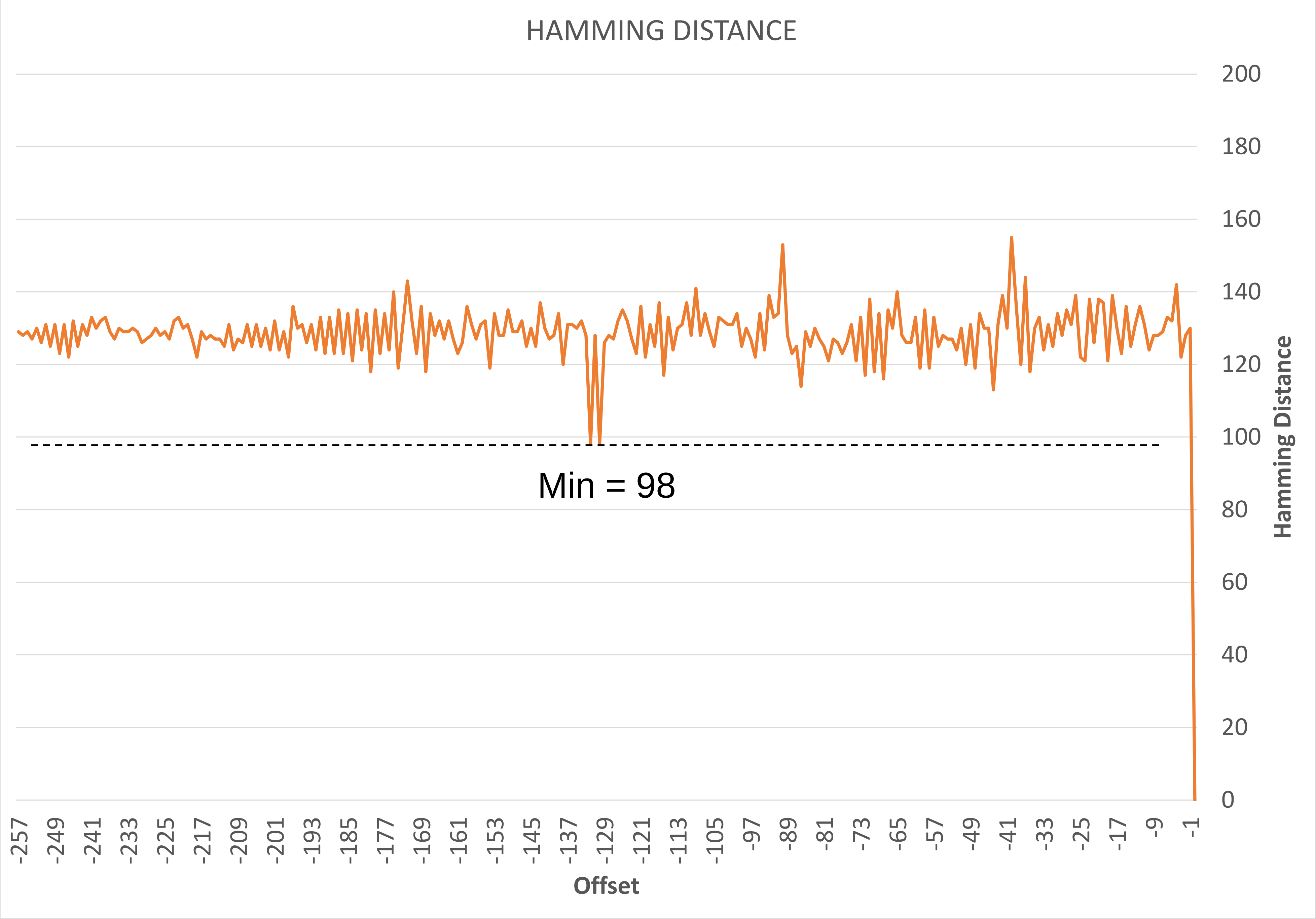


Attempt #2

- Assemble the super-delimiter as

  ‘1’ + SP[0…63] + BD[0…63] + SP[2…65] + BD[2…65]

  (i.e., 64 LSB of first SP and BD, 64 MSB of second SP and BD)

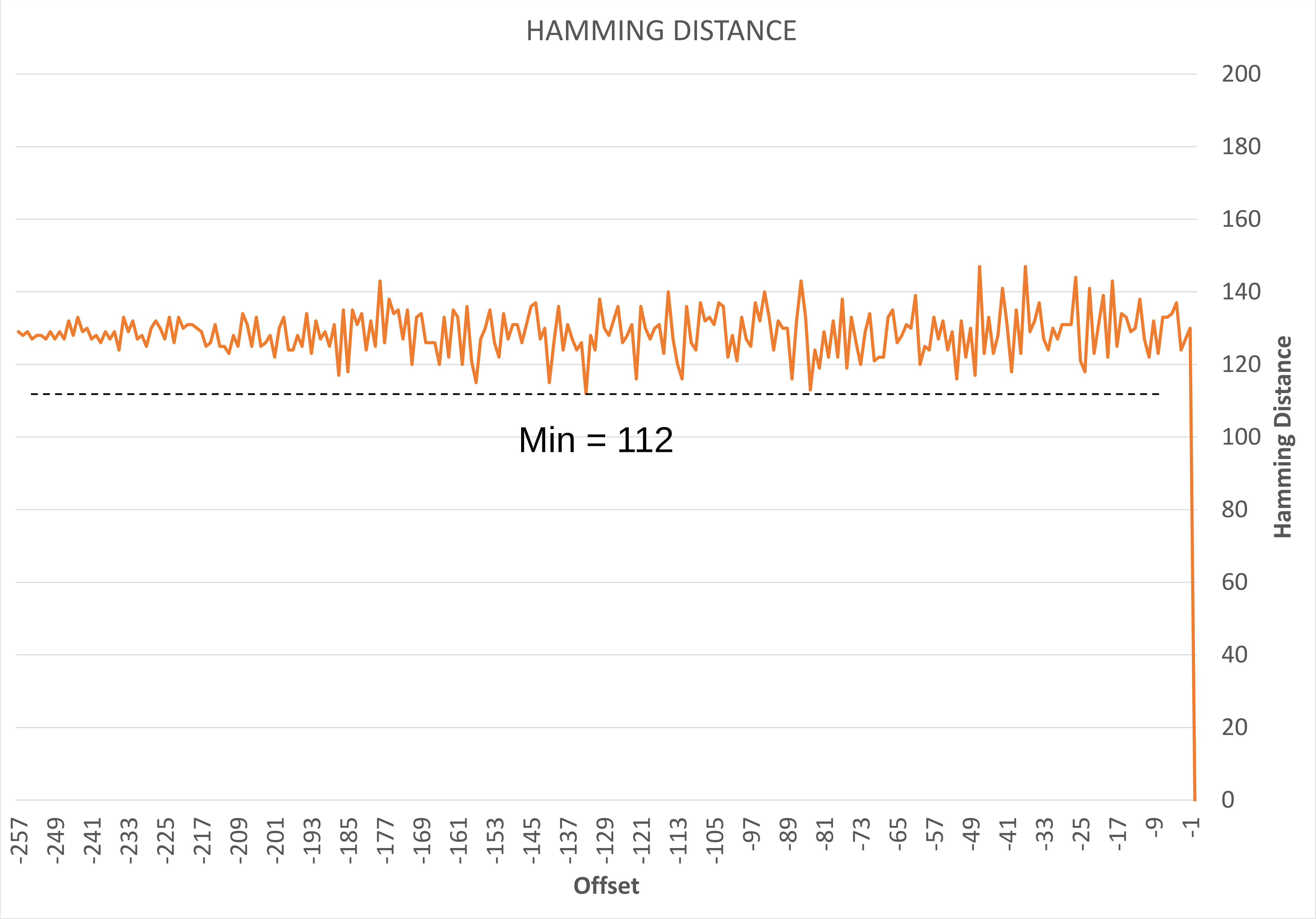


Attempt #3

- Assemble the super-delimiter as

  ‘1’ +
  SP[0…63] +
  BD[0…63] +
  SP[63…0] +
  BD[63…0]

  (i.e., reverse the second SP and BD)

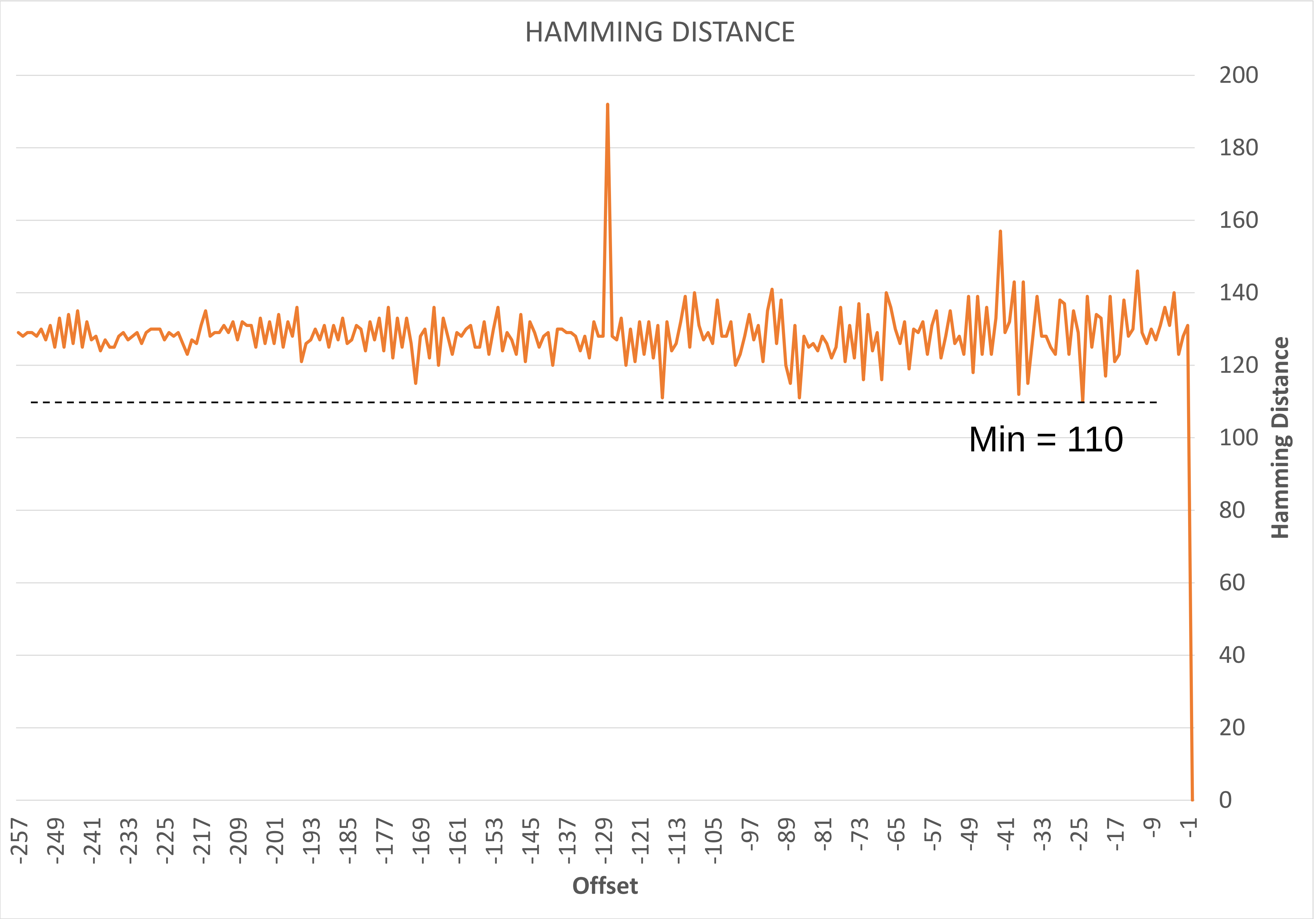


Attempt #4

- Assemble the super-delimiter as

  ‘1’ +
  SP[0…63] +
  BD[0…63] +
  $\overline{\text{SP}}$[0…63] +
  $\overline{\text{BD}}$[0…63]

  (i.e., invert the second SP and BD)

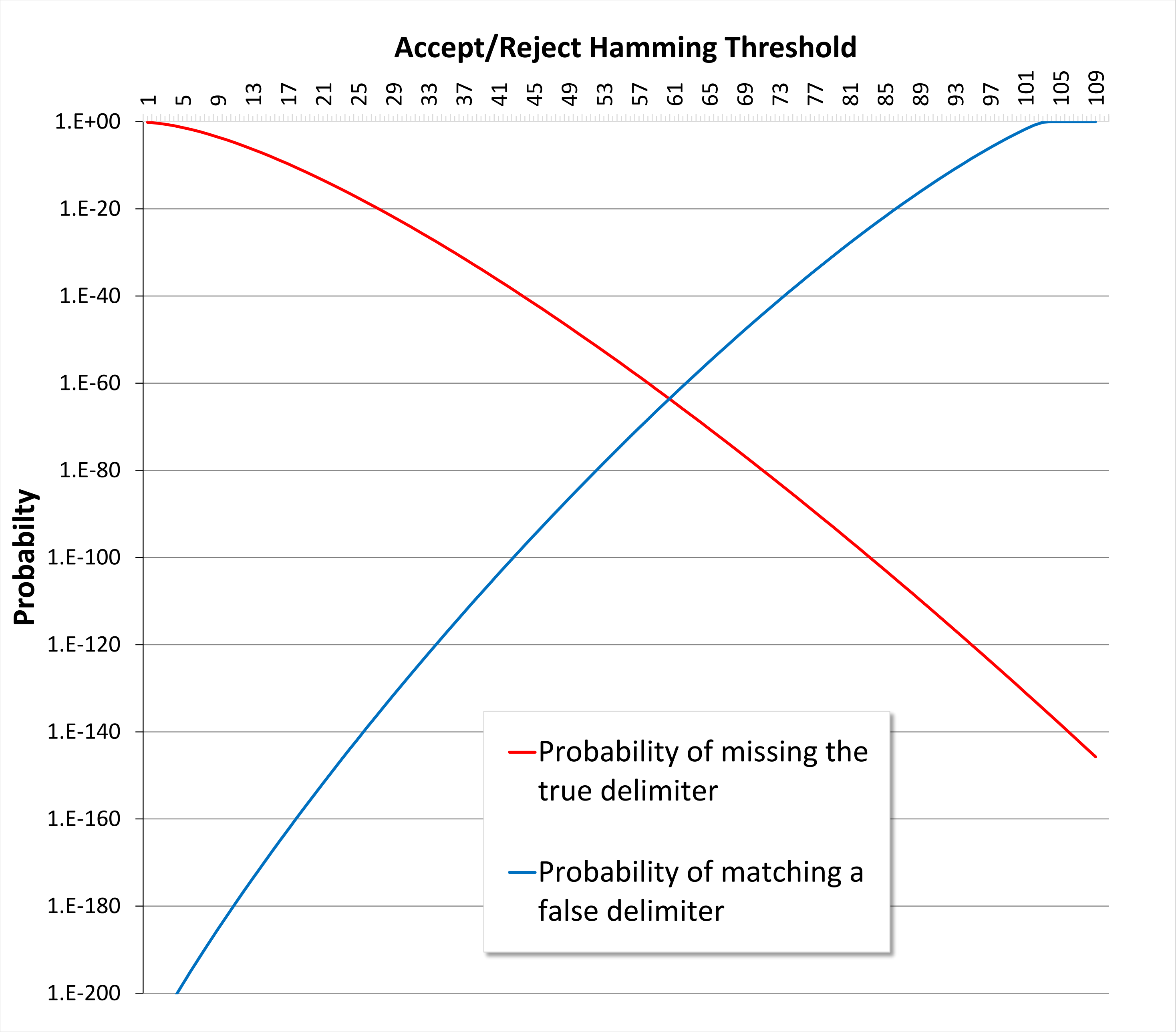


How to choose the Accept/Reject Hamming threshold?

- Match rule: **If *HammingDistance ≤ Threshold* then *declare delimiter match***
- Super-delimiter should have
  - Low probability of missing true position
  - Low probability of matching false position
- The Accept/Reject Threshold that is too small (i.e., too strict) will result is higher probability of missing true delimiter in presence of bit errors.
- The Accept/Reject Threshold that is too large (i.e., too forgiving) will result is higher probability of accepting false delimiter in presence of bit errors.
- Both missing the true position or matching the false position are equally bad and result in loss of the entire burst
- The optimal threshold is the one where the probability of missing a burst is the lowest.

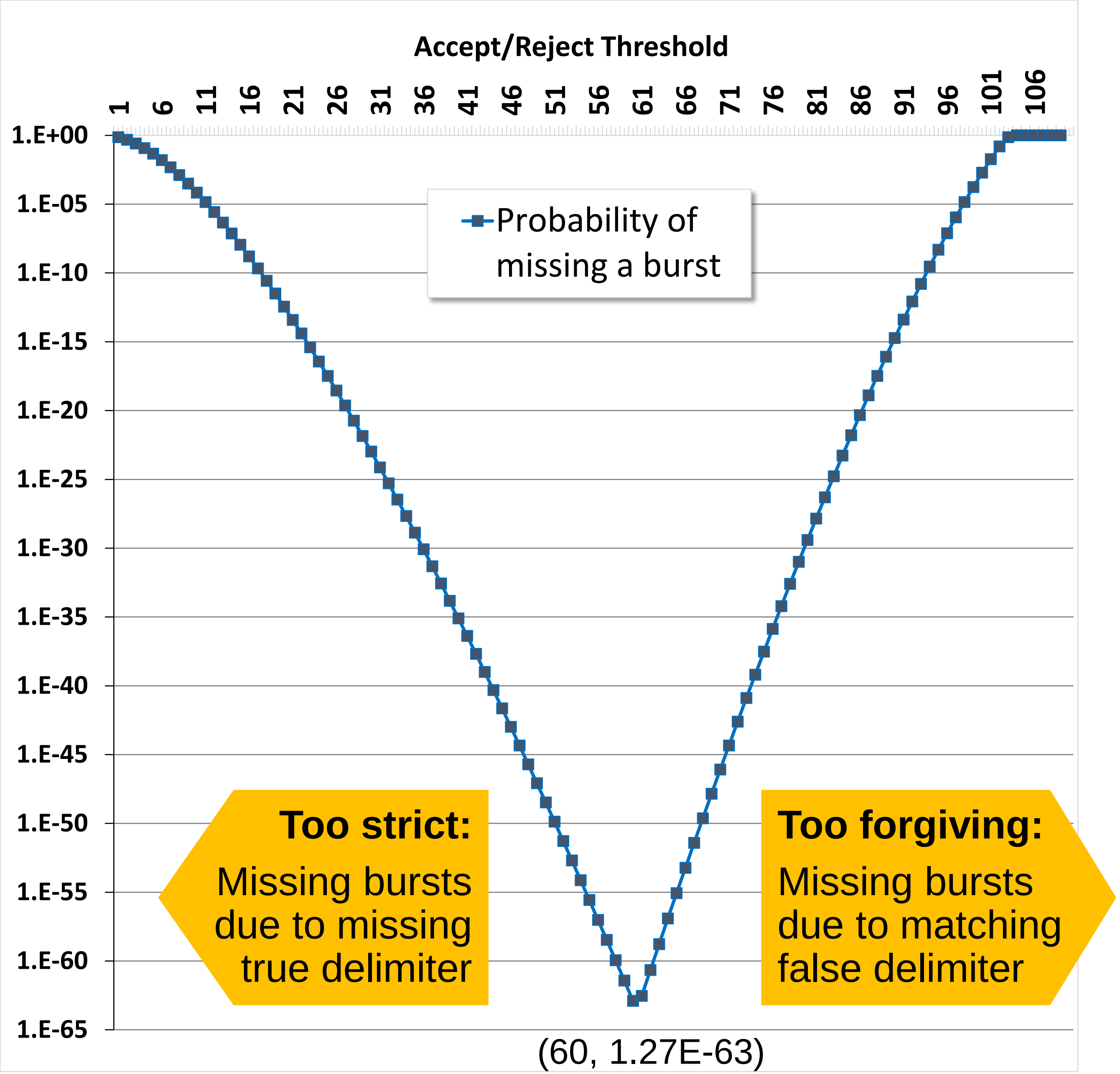
Probabilities of missing the true BD or matching a false BD

- Parameters:
  - Minimum Hamming Distance = 110 (per Attempt #4)
  - Input BER = 10^{-2}
  - SyncTime = 1712 ns (512+800+400) (longer SyncTime provides more opportunities for false match)



Optimal Accept/Reject Hamming threshold

- Best Threshold = **60 bits**, i.e., super-BD will be matched with up to 60 bit errors.
- Probability of missing a burst is **1.27×10^{-63}**
- Even taking the absolute worst case of longest sync time (1.7 µs) and smallest total burst length (1.9 µs), and assuming transmission over 4 lanes, we get the expected burst loss of one burst in **10^{49}** years



Conclusion

Proposal:

- Accept the burst delimiter constructed of
  ‘1’ + SP[0…63] + BD[0…63] + $\overline{\text{SP}}$[0…63] + $\overline{\text{BD}}$[0…63]
  - SP and BD values are as defined in 802.3av
  - 257-bit value = 0x1 + BF-40-18-E5-C5-49-BB-59 + 6B-F8-D8-12-D8-58-E4-AB
    + 40-BF-E7-1A-3A-B6-44-A6 + 94-07-27-ED-27-A7-1B-54

Advantages:

- Repeating pattern allows 4-stage x 64-bit pipelined comparator instead of a single 256-bit wide stage.
- Constructing the super-delimiter from the existing SP and BD allows the OLT and ONUs that support both .3av and .3ca to share the comparator logic and registers.

Backup

Probability of Missing a True Delimiter

- Calculation is straightforward

p – probability of bit error (BER, all bits are i.i.d)
T – BD match threshold
N – burst delimiter size (257 bits)

$$\mathrm{Prob\_BD\_miss} = P(\#err > T) = \sum_{k=T+1} \binom{N}{k} p^k (1-p)^{N-k}$$

Probability of Matching a False Delimiter

- Calculating probability of matching a false delimiter is more complicated than the calculation of probability of missing the true delimiter. On one side, a long sync time provides many opportunities for a false positive. On the other side, any random bit error has as much of a chance of increasing the Hamming distance as it has of decreasing it, so the errored sequence is typically as far away from the BD pattern as the error-free one.

- When BD is compared to any shift of {0x55… | BD} (a.k.a., the "target sequence"), some bits match, some don't. Let's separate these bits into two sets:

  H – a set of bits that are different between the BD and the true (i.e., error-free) target sequence. These are the bits that contribute to Hamming distance. Per our example on slide 7, **|H| ≥ 110**.

  G – a set of bits that match. **| G | = N – |H| ≤ 257 – 110 = 147**.

  E_H – number of bit errors in set H. Every bit error in this set decreases the measured Hamming distance.

  E_G – number of bit errors in set G. Every bit error in this set increases the measured Hamming distance.

- Then the actual measured Hamming distance is $h = |H| - E_H + E_G$

$$\text{Prob\_false\_match} = \text{Prob}(h \le T) = \text{Prob}(E_H - E_G \ge |H| - T)$$

Probability of Matching a False Delimiter

$$\text{Prob\_false\_match} = \text{Prob}(E_H - E_G \geq |H| - T)$$

- As an example, let's pick **T = 50** (i.e., we declare match if a target sequence and the BD have a measured Hamming distance of ≤ 50).

- To match a false delimiter, we need a situation where …

  a) ≥ 60 errors bit errors occurred in a 257b block (**|H| - T ≥ 110-50 = 60**).

  b) All or most of these errors were concentrated in the set **H** and no or very few errors were concentrated in the set **G** (even though size of **H** is often smaller than the size of **G: | H | ≥ 110** and **| G | ≤ 147**)

  – In other words, we need $\mathbf{E_H - E_G \geq 60}$

Measured Hamming distance *h* for given numbers of E_H and E_G errors

		E_G							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**…**
E_H	**60**	$\boxed{50}$							…
	61	49	50						…
	62	48	49	50					…
	63	47	48	49	50				…
	64	46	47	48	49	50			…
	65	45	46	47	48	49	50		…
	66	44	45	46	47	48	49	50	…
	…	…	…	…	…	…	…	…	…

Dominating term

$$\text{Prob\_false\_match}_{1\text{try}} = \sum_{k=|H|-T}^{|H|} \text{Prob}(E_H = k) \times \text{Prob}(E_G \leq k - |H| + T)$$

Probability of Matching a False Delimiter in a single try

- A good first approximation is to assume that 60 errors happened in **H** (i.e., $\mathbf{E_H = 60}$) and no errors have happened in **G** (i.e., $\mathbf{E_G = 0}$)
  - The next closest term is when $\mathbf{E_H = 61}$ and $\mathbf{E_G = 1}$ and it is approximately two orders of magnitude lower (assuming BER = 10^{-2})

$$\mathrm{Prob\_false\_match}_{1\mathrm{try}} \approx \mathrm{Prob}(E_H = |H| - T) \times \mathrm{Prob}(E_G = 0)$$

$$\mathrm{Prob\_false\_match}_{1\mathrm{try}} \approx \binom{|H|}{|H| - T} p^{(|H|-T)} (1-p)^{(|H|-(|H|-T))} (1-p)^{|G|}$$

$$\mathrm{Prob\_false\_match}_{1\mathrm{try}} \approx \binom{|H|}{T} p^{(|H|-T)} (1-p)^{(N-|H|+T)}$$

Probability of Matching a False Delimiter per entire burst

- Finally, the burst synchronization pattern of an M-bit length provides (at most) M opportunities for false match
  - This is too, a slight oversimplification which assumes each shift by 1 bit produces a new independent trial. In reality, it is not, since 256 bits out of 257 bits of a new target sequence were already tried in the previous comparison.

$$\mathrm{Prob\_false\_match}_{\mathrm{burst}} \approx 1 - (1 - \mathrm{Prob\_false\_match}_{1\mathrm{try}})^{M}$$

- When $\mathrm{Prob\_false\_match}_{1\mathrm{try}}$ is very small, we can write

$$\mathrm{Prob\_false\_match}_{\mathrm{burst}} \approx M \times \mathrm{Prob\_false\_match}_{1\mathrm{try}}$$

$$\mathrm{Prob\_false\_match}_{\mathrm{burst}} \approx M \binom{|H|}{T} p^{(|H|-T)} (1-p)^{(N-|H|+T)}$$

Selecting the Best Threshold

- The best threshold T is where the function $f(T)$ is at a minimum (as shown on slide 9).

$$f(T) = \text{Prob\_false\_match}_{\text{burst}}(T) + (1\text{-Prob\_false\_match}_{\text{burst}}(T)) \times \text{Prob\_BD\_miss}(T)$$

$$\approx \text{Prob\_BD\_miss}(T) + \text{Prob\_false\_match}_{\text{burst}}(T)$$