# Definition of Burst Error

Before we begin, we need to define the definition of a burst error. The common definition of a burst error can be defined as a consecutive error preceded with and followed by correct bits. Note that a burst error definition in Wikipedia is slightly different, but I believe this definition is rarely used in wireline communication.

# PRE-FEC BER:

Since the burst error length is uniformly distributed between 1 and 20, and the system does not include a DFE in its receiver path, the PRE-FEC BER computation is relatively easy:

The numerator represents the expected length of burst error whenever a bit error occurs (with a chance of 0.0005), and the denominator represents the expected length of burst error followed by a correct bit, indicating the end of a burst error.

POST-FEC BER:

[1] presents a method to compute POST-FEC BER using Markov-chain and Trellis diagram. Some of the computation methods are useful for this problem (especially Eq. (28)), but this method cannot be directly applicable to this problem. This is because the burst error length is determined by uniformly distributed, making the model a non-Markovian process. Sweeping the entire bit (5280 for KR4 and 5440 for KP4) and computing the symbol and bit error probability in a brute-force fashion is also difficult since each bit has 21 different possible branches (1 for correct, 20 for burst errors).

Instead, one possible way is using the following assumption that sacrifices the accuracy but makes the computation feasible:

* We assume that only one bit error occurs for 1~3 symbol blocks.

Since the maximum burst error length is 20, it will create up to 3 symbol errors. For instance, if a bit error occurs on the second bit of a symbol with a burst error length of 20, then the last burst error will appear on the first index of the 3rd symbol. In this situation, if another bit error occurs between the second and the last bit in the 3rd symbol, then the consecutive symbol error may grow to infinitely long. The above assumption excludes this possibility, making the bit error computation simple.

For the rest of this report, I assume that KR4 FEC is used (528, 514, 7)

and represent the chance of getting and avoiding a symbol error, respectively.

Below table summarizes the burst error combinations for each index in a symbol to get a certain symbol errors (1~3).

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | total |
| 1x symbol error | 1~10 | 1~9 | 1~8 | 1~7 | 1~6 | 1~5 | 1~4 | 1~3 | 1~2 | 1 | 55 |
| 2x symbol error | 11~20 | 10~19 | 9~18 | 8~17 | 7~16 | 6~15 | 5~14 | 4~13 | 3~12 | 2~11 | 100 |
| 3x symbol err | X | 20 | 19~20 | 18~20 | 17~20 | 16~20 | 15~20 | 14~20 | 13~20 | 12~20 | 45 |

When a bit error occurs in a symbol, the probability of ending up having 1x, 2x, and 3x symbol errors are:

We can also compute the expected length of burst error for these burst error situations:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Burst len | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| combinations | 1 | 11  2 | 111  12  3 | 1111  112  13  22  ~~4~~ | 11111  1112  113  122  ~~14~~  23  ~~5~~ | 111111  11112  1113  1122  ~~114~~  123  ~~15~~  ~~24~~  33  ~~6~~ | 1 1 1 1 1 1 1  1 1 1 1 1 2  1 1 1 1 3  1 1 1 2 2  ~~1 1 1 4~~  1 1 2 3  ~~1 1 5~~  ~~1 2 4~~  1 3 3  ~~1 6~~  ~~2 5~~  ~~3 4~~  ~~7~~ |

# REFERENCE:

[1] M. Yang, S. Shahramian, H. Shakiba, H. Wong, P. Krotnev, and A. C. Carusone, “Statistical BER Analysis of Wireline Links With Non-Binary Linear Block Codes Subject to DFE Error Propagation,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 67, no. 1, pp. 284–297, Jan. 2020.