Upstream Burst Structure

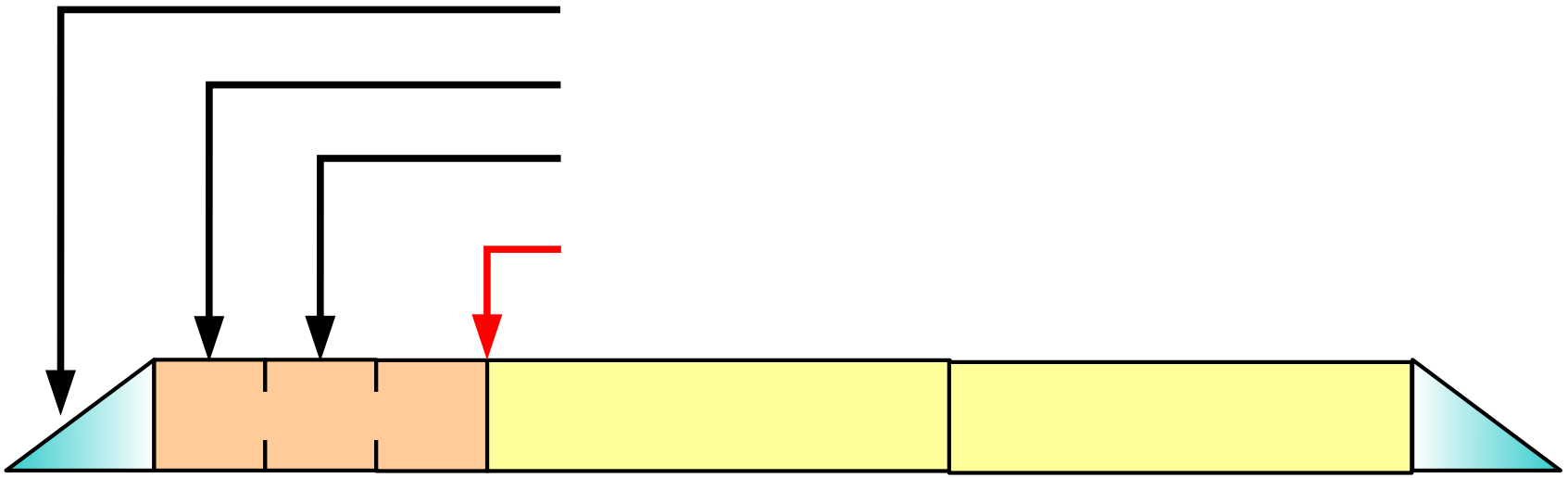
Marek Hajduczenia, Charter

Highlights

- Overview materials sourced from 10G-EPON project: 3av_0701_effenberger_1.pdf. Burst structure, locking mechanism, and general analysis methodology is equally applicable to 100G-EPON as well
- Burst loss and false lock probabilities were updated and are now cumulative rather than approximate.
- Calculations carries out using Matlab script (attached) for delimiters lengths of 66, 132, 129, and 257 bits, aligned with different line code options.
  - 66-bit long delimiter insufficient for NG-EPON with raw BER of 10^{-2} and higher burst rates (0.011 years' long burst loss)
  - Little difference between 132- and 129-bit long delimiters, both could be adopted for NG-EPON
  - Alignment with selected line code will decide between 129, 132, and 257-bit (or any other) long delimiters

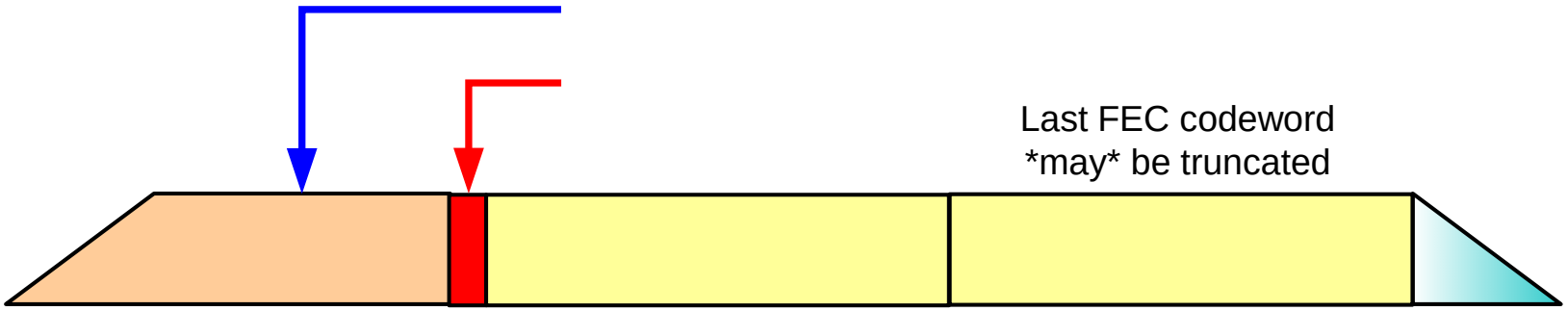
Upstream Synchronization Issues

- OLT adjusts gain and recovers clock (AGC/CDR) during special burst portion (0xBF4018E5C549BB59, SP per 76.3.2.5.2) called Sync Pattern / Time
- OLT cannot use FEC until FEC codeword boundary is reliably detected using pre-FEC (uncorrected) data
- OLT must be able to reliably locate X-bit block boundary to identify start of FEC-encoded data (X could be 66, 132, 129, 257, depending on line code selection)

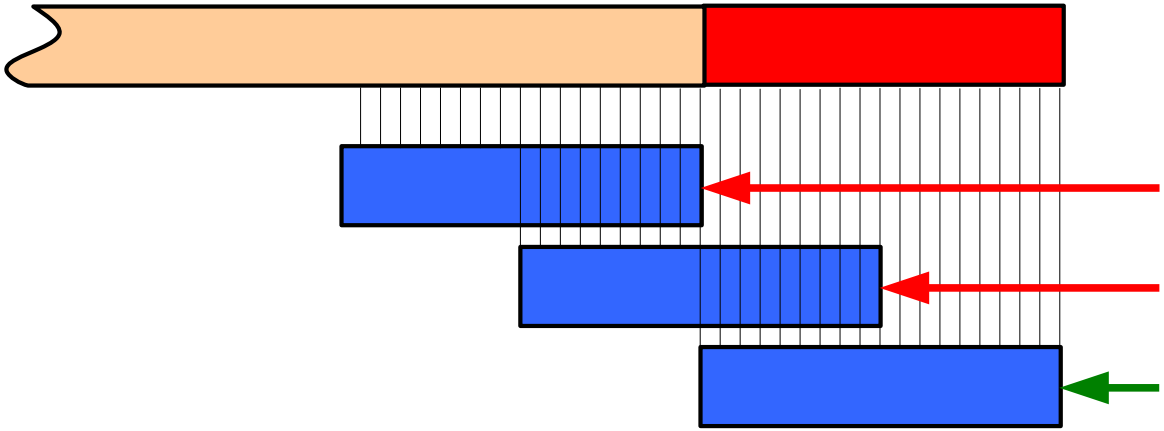


Upstream Burst Structure

- Similar to 10G-EPON, Sync Time pattern needs to be followed by well-known Start of Burst (SoB) delimiter

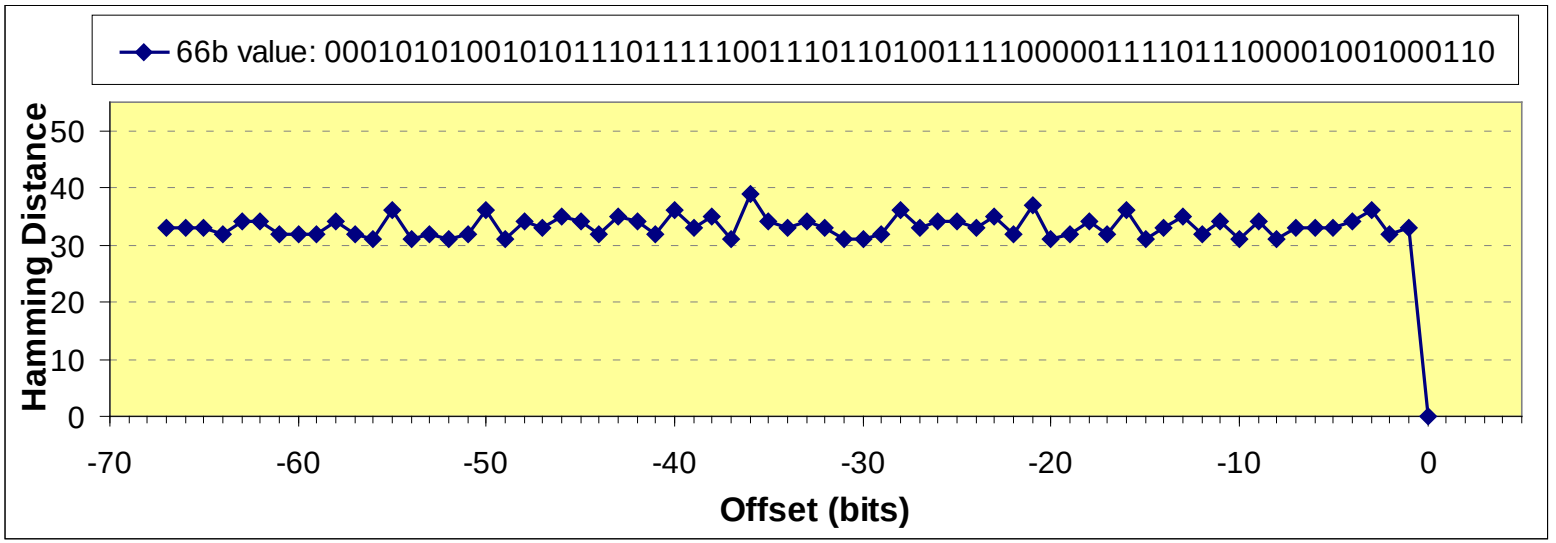


- SoB pattern must have high Hamming distance with any false synchronization candidate position



SoB Requirements

- There are several SoB requirements
  - Large Hamming distance with Sync Pattern in the upstream burst (0xBF4018E5C549BB59, SP per 76.3.2.5.2)
  - Large Hamming distance with its own shifted version
  - Large Hamming distance against both patterns even in presence of E bit errors within the sequence (pre-FEC)
  - Close to DC balance (same number of 1s and 0s)
- Example of 66-bit SoB and its Hamming distance



Locking Probabilities / 1

- Assume: N = block size in bits (66), E1 = admissible bit error count per block (e.g., 12), E2 = number of bit errors in sync pattern to turn it into a barely matching pattern, resulting in false detection (e.g., 20), and BER = bit error ratio for incoming data stream (e.g., 10^{-3}). **E2 = HD – E1 + 1**, where HD = Hamming Distance of SoB delimiter
- When a burst arrives, OLT correlator searches for the SoB, tolerating up to E1-1 errors

Probability of missing burst lock (burst loss, P_{loss}): *at least* E1 bit errors in delimiter of N bits and OLT correlator will not find the delimiter sequence properly in incoming data stream with raw bit error ratio of BER

$$\sum_{x=E1}^{N} \frac{N!}{x! \times (N-x)!} \times BER^{x} \times (1-BER)^{N-x}$$

Locking Probabilities/2

Probability of false burst lock (P_{false}): for any position within Sync Time, if two N-bit blocks have a Hamming distance of HD, then any random error has HD/N chance of reducing HD and (N-HD)/N chance of increasing it. The false lock only happens when the number of errors that reduce HD exceed the number of errors that increase HD by E2.

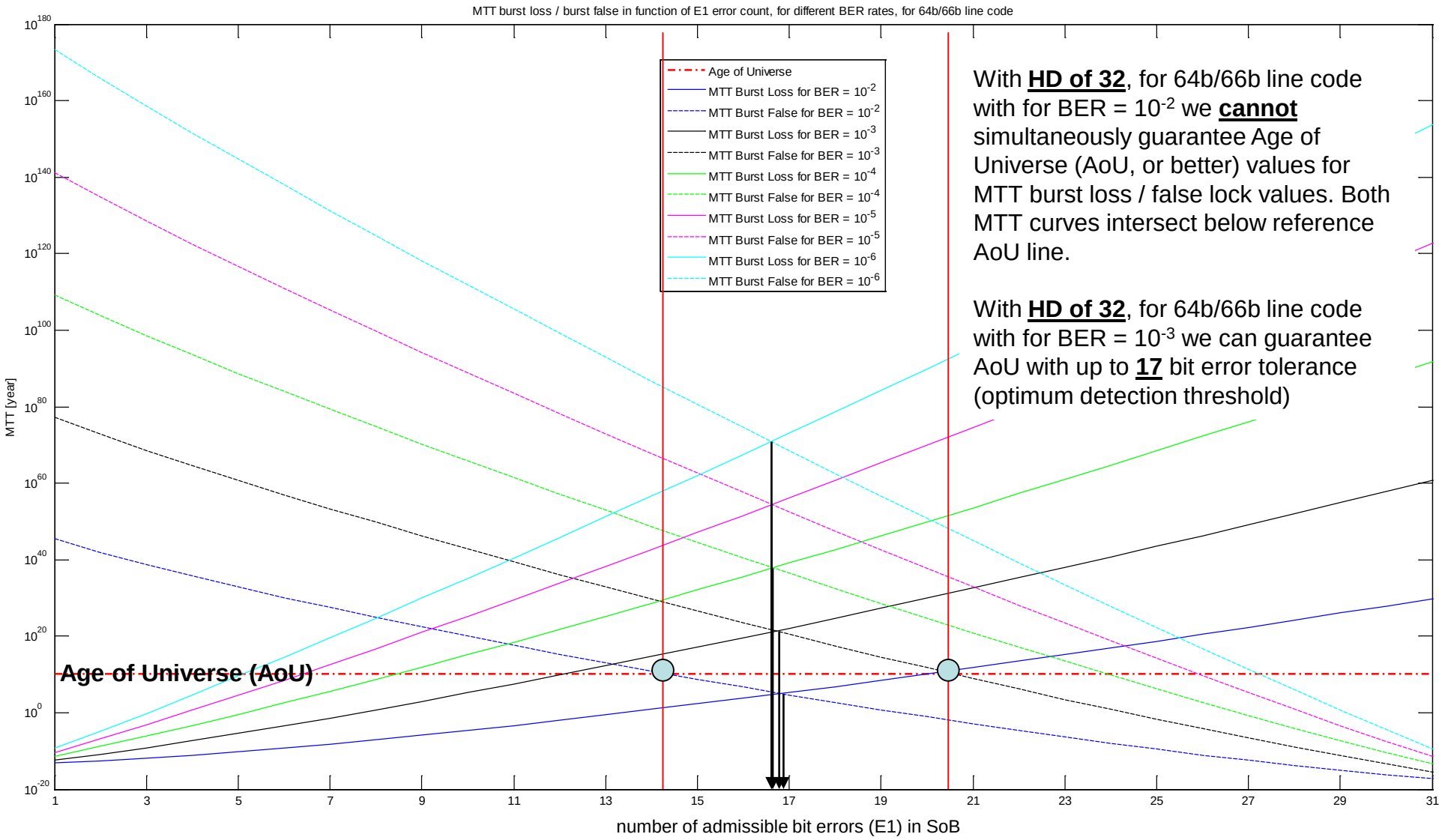
False lock can happen for any shift position (S) within the Sync Time on any of the active data lanes (L). Number of shift positions within Sync Time depends on Sync Time size (in ns, S_b), line code selected (in bits, code block pre coding, e.g., 64, N_{pre}, and code block post coding, e.g., 66, N_{post}), MAC rate (in Gbps, R_{MAC}), and SoB size (in bits, N)

$$\sum_{x=E2}^{HD}\sum_{y=E2}^{x}\frac{HD!}{x!\times(HD-x)!}\times BER^{x}\times(1-BER)^{HD-y}\times(1-BER)^{N-HD}\times L\times\left(S_b\times R_{MAC}\times\frac{N_{post}}{N_{pre}}-N\right)\times$$

Locking Probabilities / 3

- Mean Time To (MTT) burst loss (SoB is not detected correctly) and false burst lock (SoB was detected at the wrong position due to bit errors in sync pattern) is related to the average burst rate and burst size. For calculation, assume 1 Mburst/sec (across 4 lanes) = 4 µs-long burst per lane (4 lane active, worst case scenario)
- Sync Time (S_b) length is derived from existing 10G-EPON values; use Figure 76–14 for general view of burst structure, 75.7.14 for Ton/Toff measurement procedure and values of individual parameters; Figure 60–8 shows details of burst structure timing
  - All max values: T_{on} = 512 ns, $T_{receiver\_settling}$ = 800 ns, T_{cdr} = 400 ns, $T_{code\_group\_align}$ = 0 ns, T_{off} = 512 ns
  - $S_b = T_{on} + T_{receiver\_settling} + T_{cdr} + T_{code\_group\_align} =$ $= 512 + 800 + 400$ ns = 1712 ns (max)

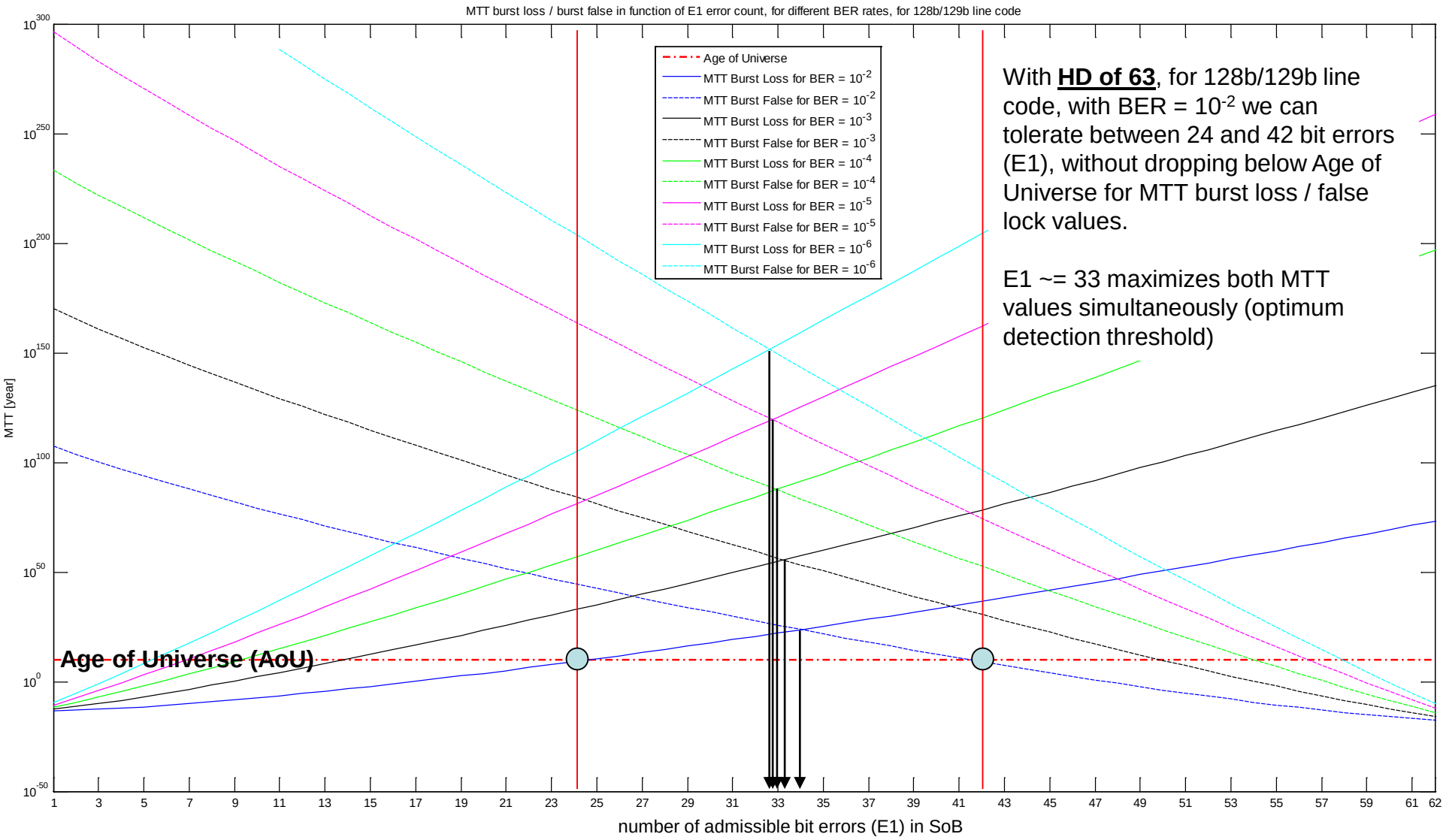
Results - 64b/66b line code



With **HD of 32**, for 64b/66b line code with for BER = 10^{-2} we **cannot** simultaneously guarantee Age of Universe (AoU, or better) values for MTT burst loss / false lock values. Both MTT curves intersect below reference AoU line.

With **HD of 32**, for 64b/66b line code with for BER = 10^{-3} we can guarantee AoU with up to **17** bit error tolerance (optimum detection threshold)

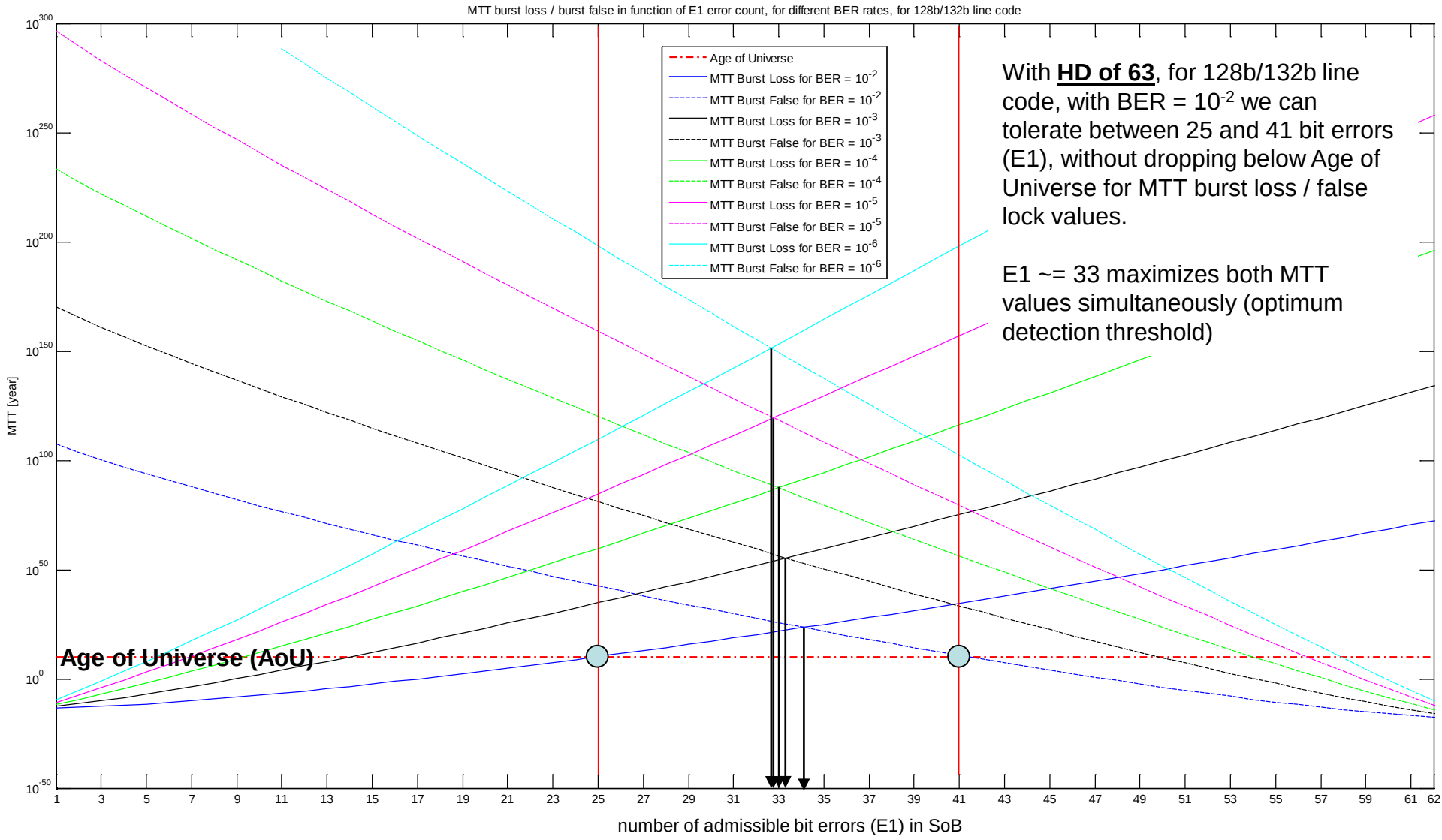
Results - 128b/129b line code



With **HD of 63**, for 128b/129b line code, with BER = 10^{-2} we can tolerate between 24 and 42 bit errors (E1), without dropping below Age of Universe for MTT burst loss / false lock values.

E1 ~= 33 maximizes both MTT values simultaneously (optimum detection threshold)

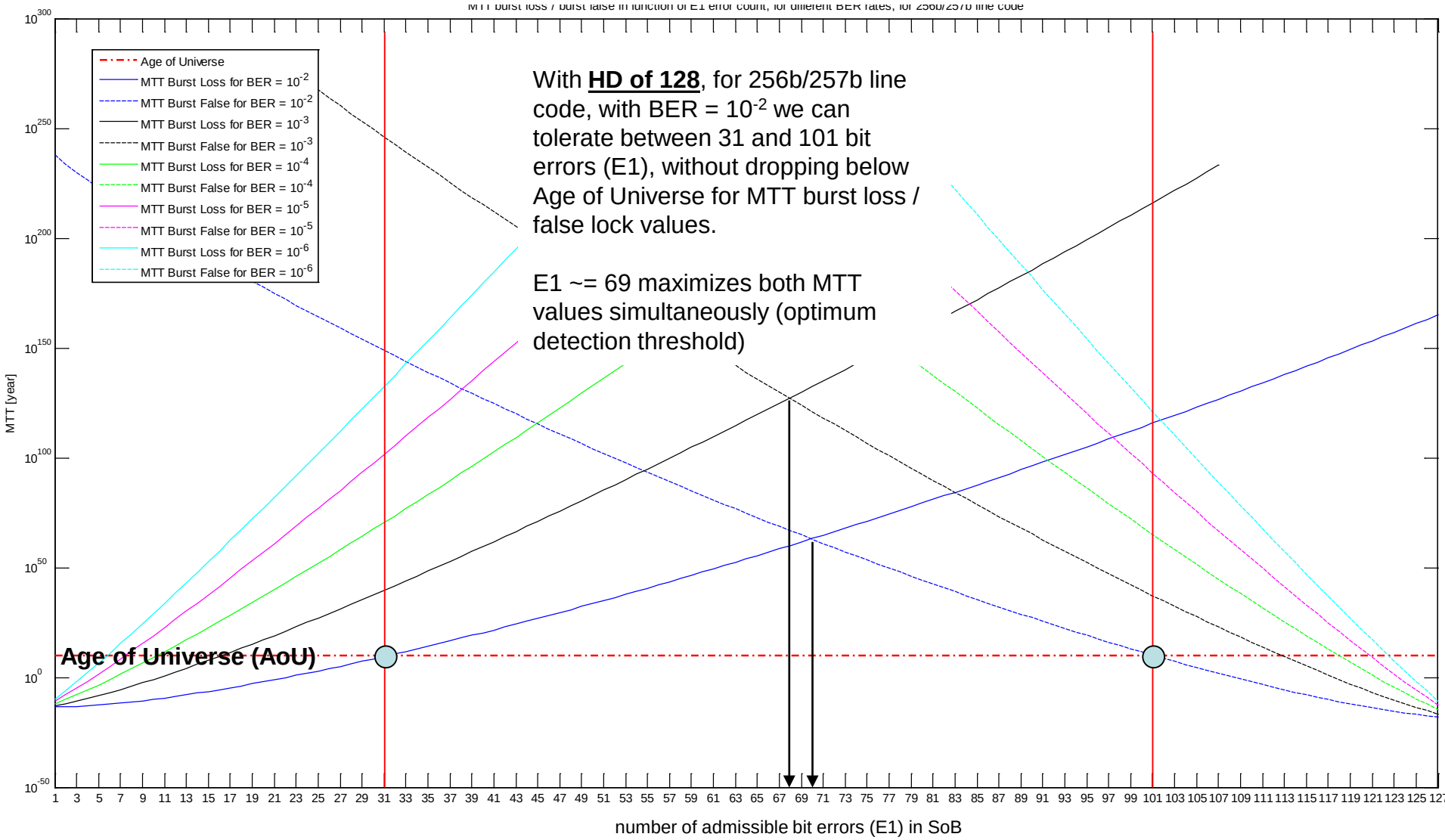
Results - 128b/132b line code



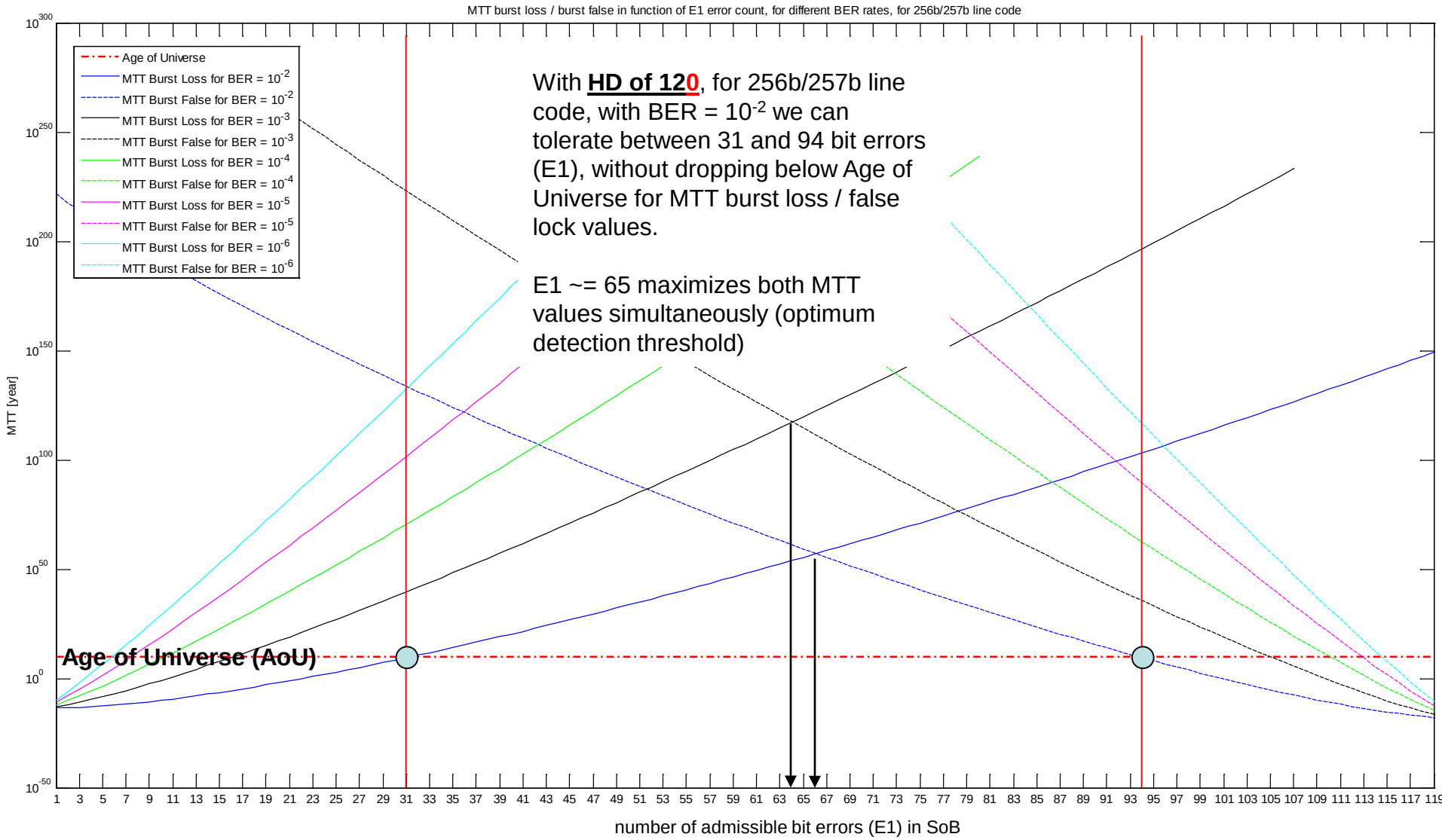
With **HD of 63**, for 128b/132b line code, with BER = 10^{-2} we can tolerate between 25 and 41 bit errors (E1), without dropping below Age of Universe for MTT burst loss / false lock values.

E1 ~= 33 maximizes both MTT values simultaneously (optimum detection threshold)

Results - 256b/257b line code / a



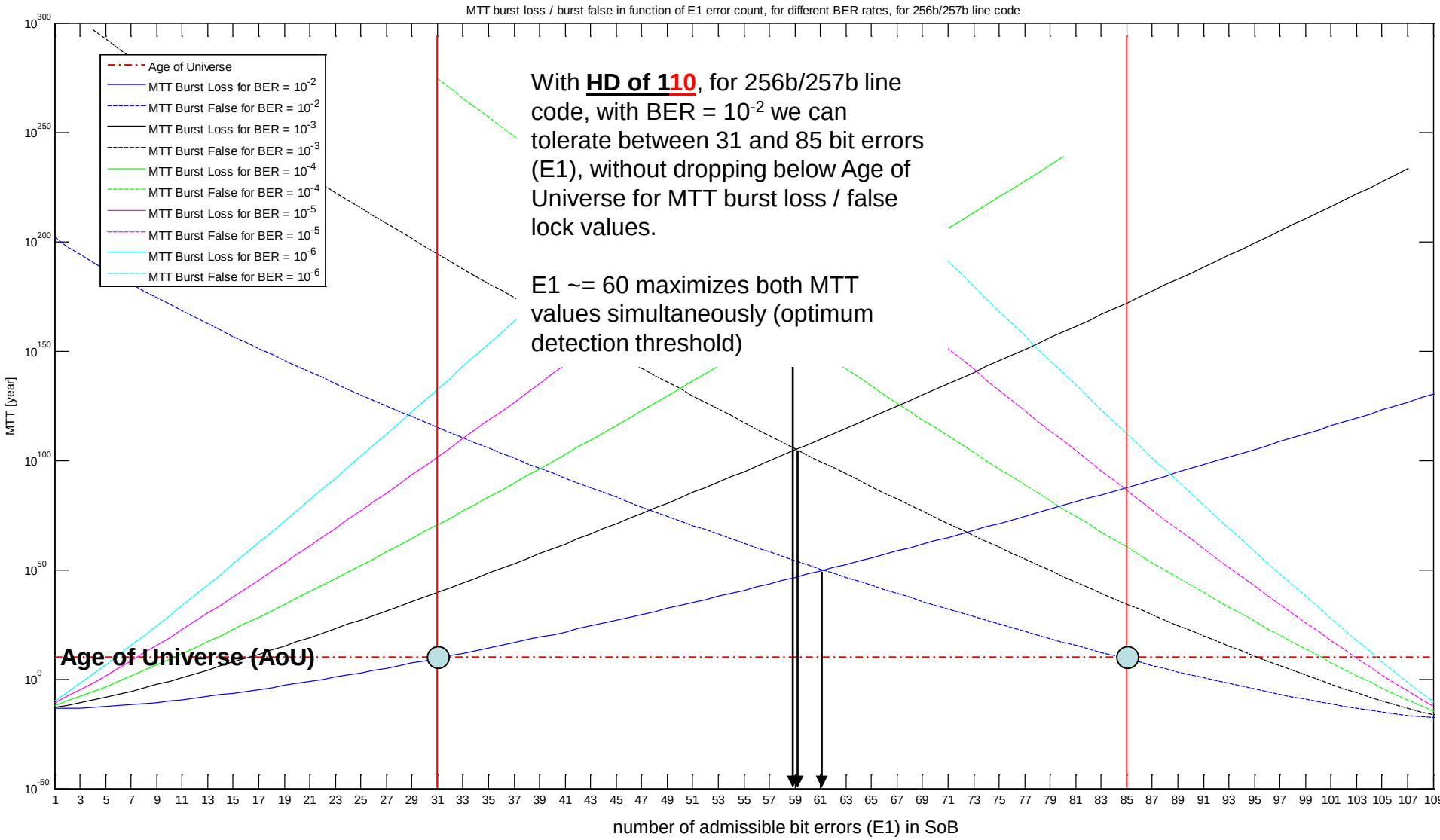
Results - 256b/257b line code / b



With **HD of 120**, for 256b/257b line code, with BER = 10^{-2} we can tolerate between 31 and 94 bit errors (E1), without dropping below Age of Universe for MTT burst loss / false lock values.

E1 ~= 65 maximizes both MTT values simultaneously (optimum detection threshold)

Results - 256b/257b line code /c



Calculation Scripts / Matlab

epon_burst_mtt_plot.m

- Matlab .m file implementation of cumulative P_{loss} and P_{false} calculations is attached (above)
- Calculation parameters are described in the files:
  - varMacRate = MAC data rate, in Gbps
  - varLaneCount = number of active lanes
  - varBurstCountPerSecond = burst number per second
  - varBlockSizeCoded = line code block, post coding, e.g., 66 bits
  - varBlockSizePreCoded = line code block, pre coding, e.g., 64 bits
  - varHammingDistance = target Hamming distance
- For calculations, a year is assumed to be 31,556,952 seconds long (solar year, not an average calendar year)
- Script plots MTT burst loss / false burst lock for BER of 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, and 10^{-6}

Delimiter Search / 1

- Based on C++ optimized Gosper algorithm (also known as SNOOB = Same Number Of One Bits)
  - Mechanism relies on bitwise operations to find **the next smallest** (N+1) number following current number (N, where both numbers have the **same** number of 1s/0s in binary form.
  - Calculations start from the smallest number meeting DC balance criteria, e.g., 66-bit number, with 32 x 1s and 32 x 0s, where all 1s are rightmost, i.e., 0x1FFFFFFF.
  - Iterate through the search space, calculating N+1 delimiter meeting DC balance criteria based on N delimiter. Process is iterative and there is **no** known algorithm to calculate N+M (where M > 1) delimiter value based on N delimiter value
  - Once given DC balance space is exhausted, search through DC-unbalanced delimiters, e.g., 31 x 1s + 33 x 0s and inverse. Solution space depends on the size of acceptable DC balance for the target sequence. Search methodology does not change, though, irrespective of the starting DC balance condition.

Delimiter Search / 2

- Based on C++ optimized Gosper algorithm (also known as SNOOB = Same Number Of One Bits)
  - C++-optimizations possible only via aggregation of intermediate N+1 delimiters and calculation of search criteria values (run-length, Hamming distance) across multiple threads. For better compute efficiency, parallel calculations are offloaded to CUDA GPU due to large number of threads blocks (e.g., o GTX1080, 2560 x 1024 parallel calculations).
  - Implemented fast 512-bit long unsigned integer class to support delimiter searches past 64-bit long sequences, limited by native C++ *long long unsigned int* type. Existing large integer classes are flexible but suffer from low performance.
  - C++11 compatible Visual Studio 2017 multi-threaded implementation with **GPU (CUDA) offload** is attached below.

Delimiter Search / 3

- Output for 44-bit long delimiters (test run)

DEL.44.HAM.21.C1.21.RUN.10.txt

44-bit long delimiter, 21/23 DC balance, run length up to 10 bits, minimum Hamming distance of 21 or better

DEL.44.HAM.21.C1.22.RUN.10.txt

44-bit long delimiter, 22/22 DC balance, run length up to 10 bits, minimum Hamming distance of 21 or better

DEL.44.HAM.21.C1.23.RUN.10.txt

44-bit long delimiter, 23/21 DC balance, run length up to 10 bits, minimum Hamming distance of 21 or better

- Target sequence can be then selected based on DC balance and expected run length, e.g., 22/22 balance, 3 or shorter run length

DEL.44.HAM.21.C1.22.RUN.3.txt