FEC Codeword Size Selection Based on Burst Size

Rich Prodan

Upstream Full FEC Codeword Sizes Previously Adopted

- **Long codeword:**
  - 16200 total bits
  - 14400 information bits including 40 CRC bits
  - 1800 parity bits
  - 88.9% code rate with 88.6% overall efficiency
- **Medium codeword:**
  - 5940 total bits
  - 5040 information bits including 40 CRC bits
  - 900 parity bits
  - 84.8% code rate with 84.2% overall efficiency
- **Short codeword:**
  - 1120 total bits
  - 840 information bits including 40 CRC bits
  - 280 parity bits
  - 75% code rate with 71.4% overall efficiency

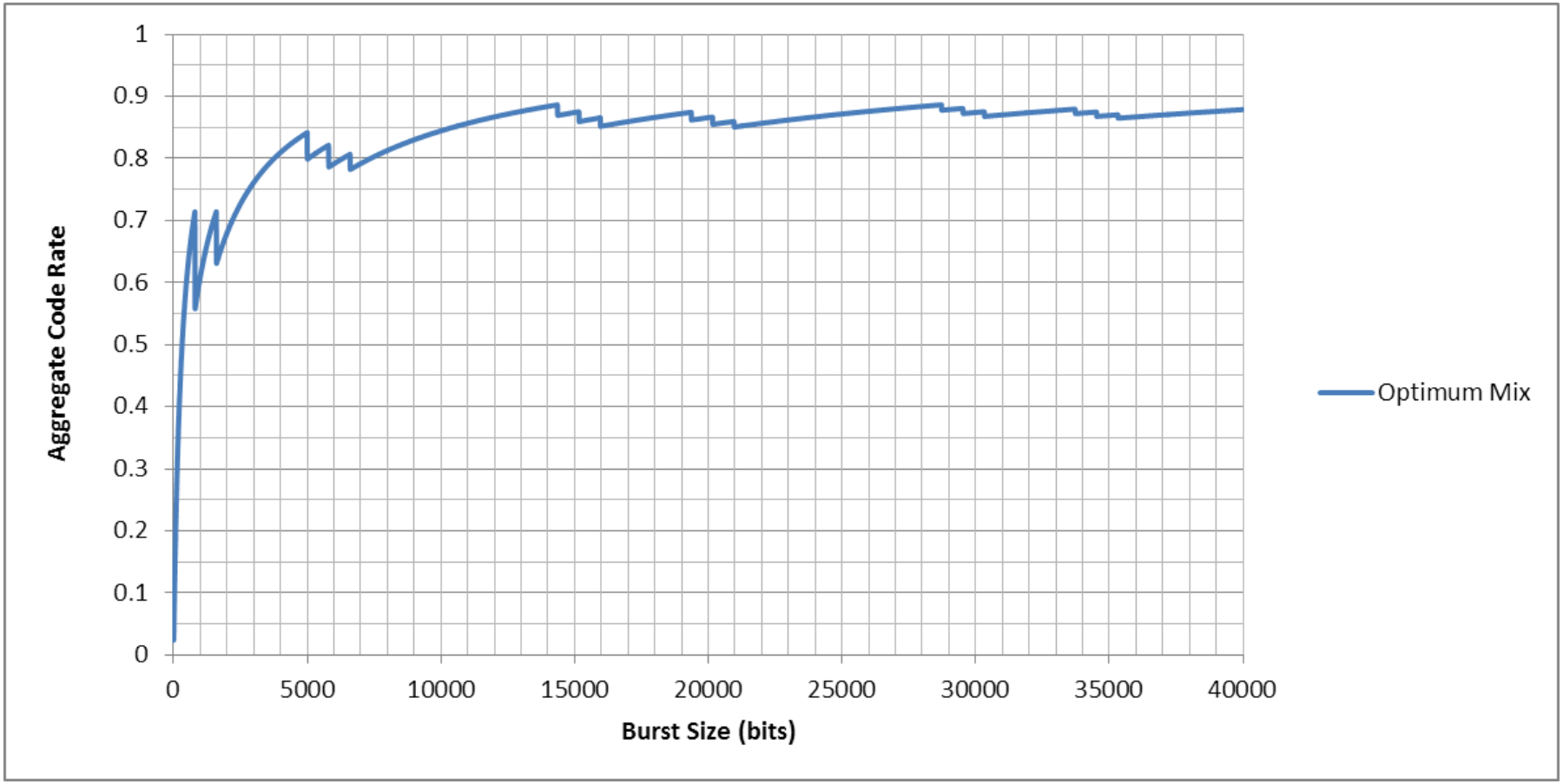
Simplest Maximally Efficient Upstream FEC Codeword Shortening

- **Starting Proposal:**
  - Support shortened LONG codewords down to 6601 information bits
    - This is the point where the number of parity bits used for a long codeword is less than the number required if medium and short codewords were used instead
  - Support shortened MEDIUM codewords down to 1601 information bits
    - This is the point where the number of parity bits used for a medium codeword is less than the number required if short codewords were used instead
  - Support shortened SHORT codewords down to as little as 1 information bit
- **Shortening a codeword down to 1 bit could be problematic**
- **The next couple slides will show the algorithm based on the starting proposal**
- **Then an adjustment will be made to address the 1-bit-codeword problem**

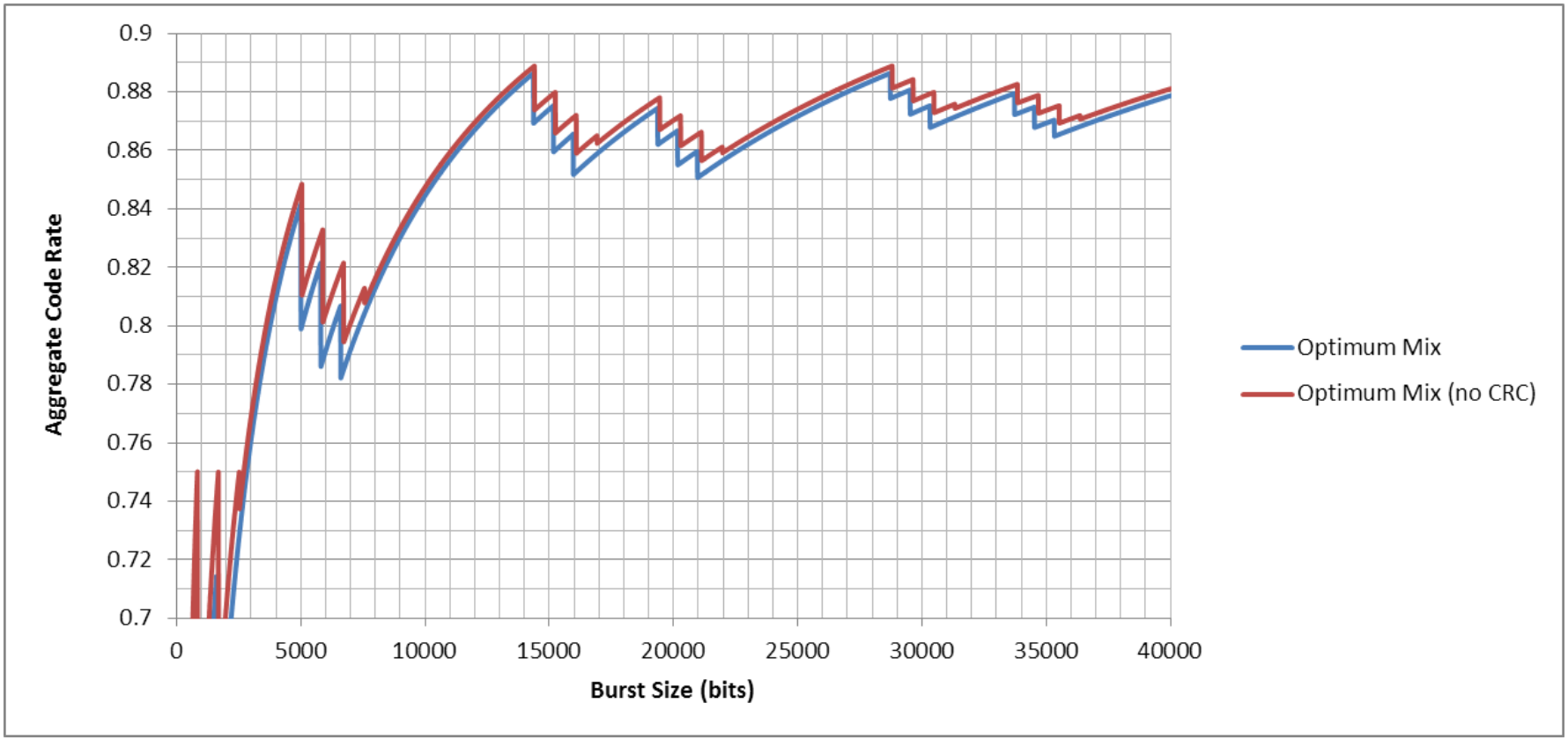
Basic Steps For Determining Codeword Sizes

- **Choice of codeword sizes is based entirely on burst size**
  - Unambiguous at both ends
- **Details of algorithm, including codeword sizes and thresholds for shortening, are embedded in spec**
- **Basic steps for conversion in either direction:**
  - If there are enough bits to create a full long codeword, do so. Keep doing this until there aren't enough bits left.
  - If there are now enough bits to create a shortened long codeword (subject to the thresholds above), do so, and end the burst.
  - Otherwise, if there are enough bits to create a full medium codeword, do so. Keep doing this until there aren't enough bits left.
  - If there are now enough bits left to create a shortened medium codeword (subject to the thresholds above), do so, and end the burst.
  - Otherwise, if there are enough bits to create a full short codeword, do so. Keep doing this until there aren't enough bits left.
  - Use whatever bits remain to create a shortened short codeword, and end the burst.

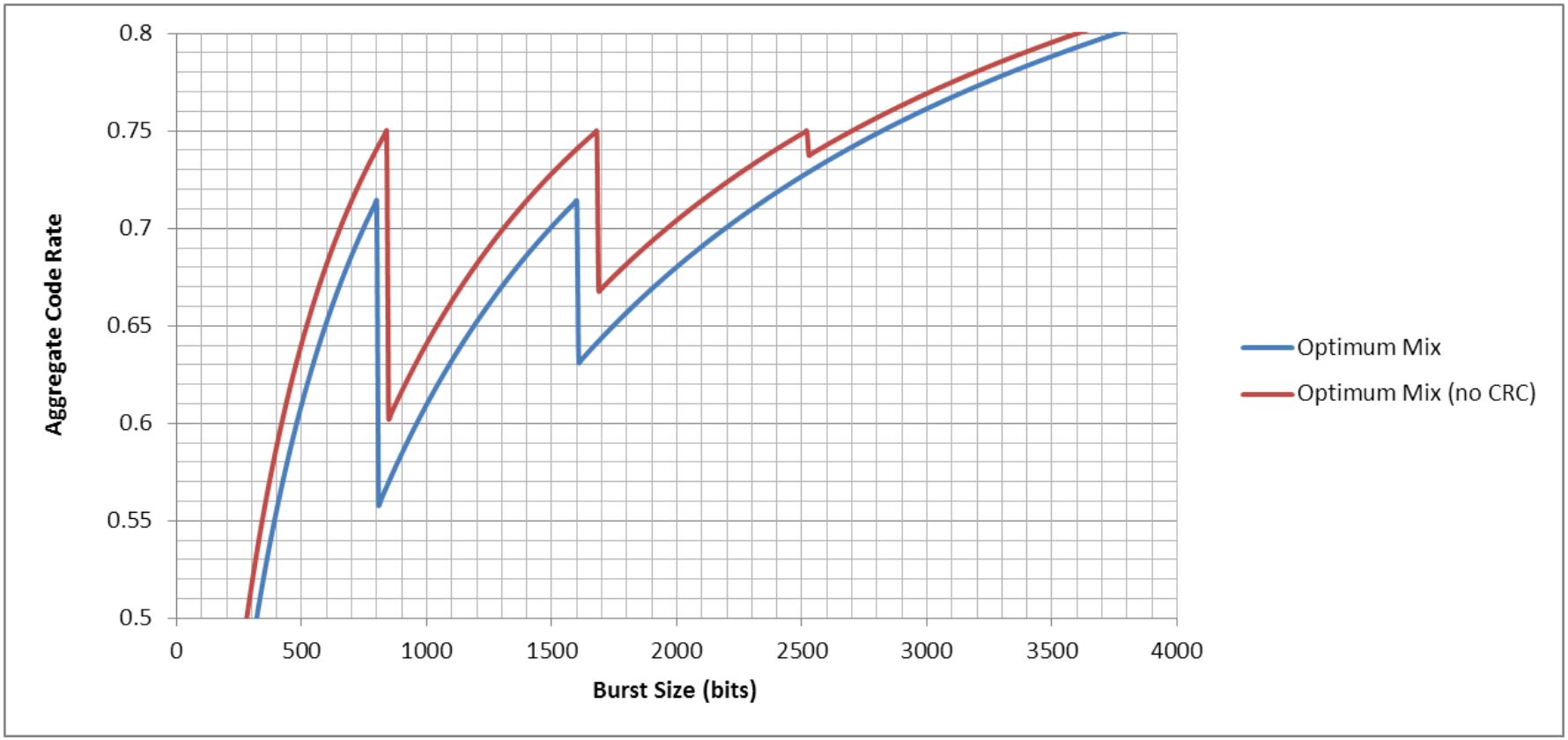
Efficiency (Code Rate) vs. Burst Size (Information Bits)



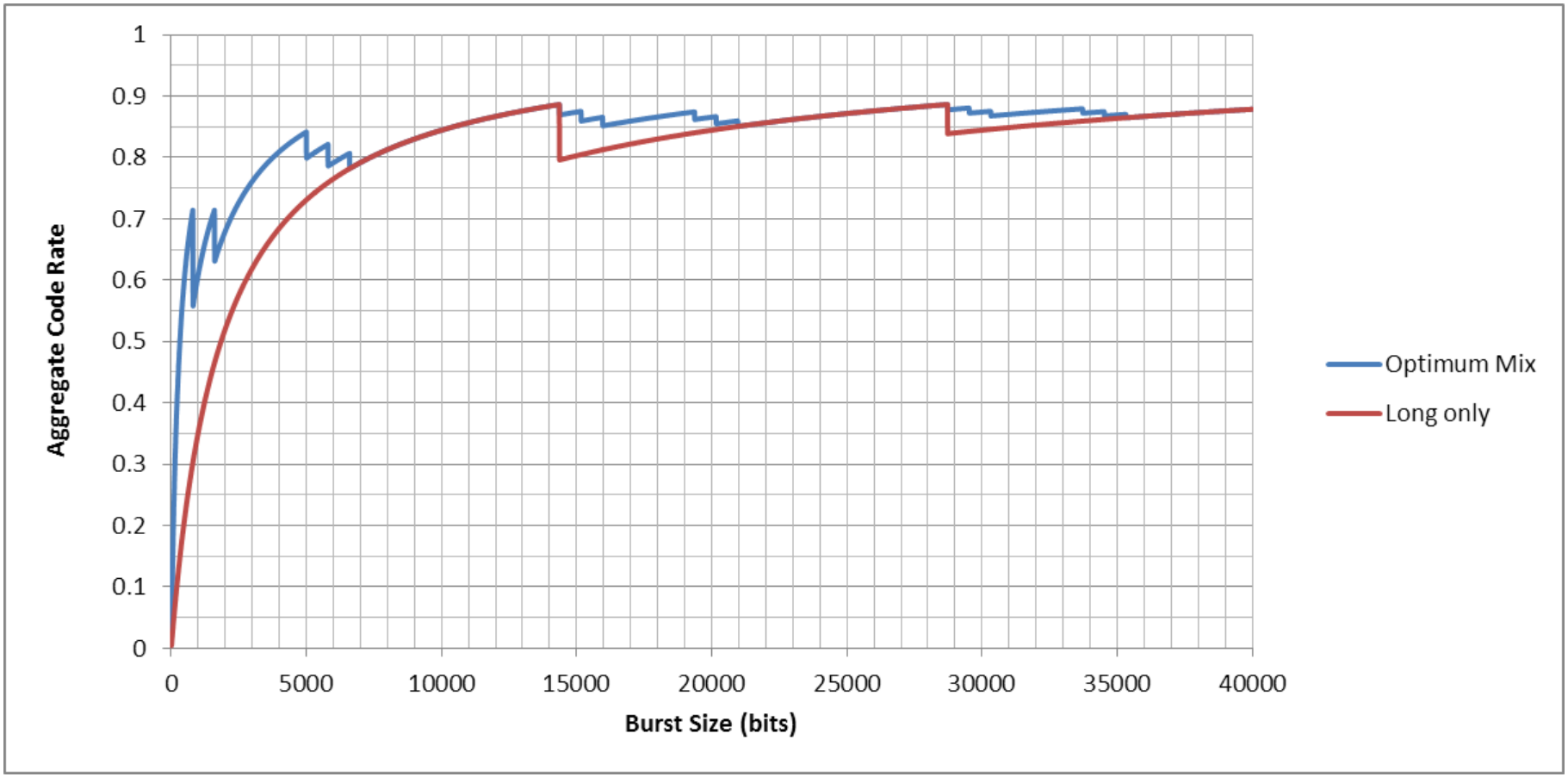
Efficiency (Code Rate) vs. Burst Size (Information Bits)



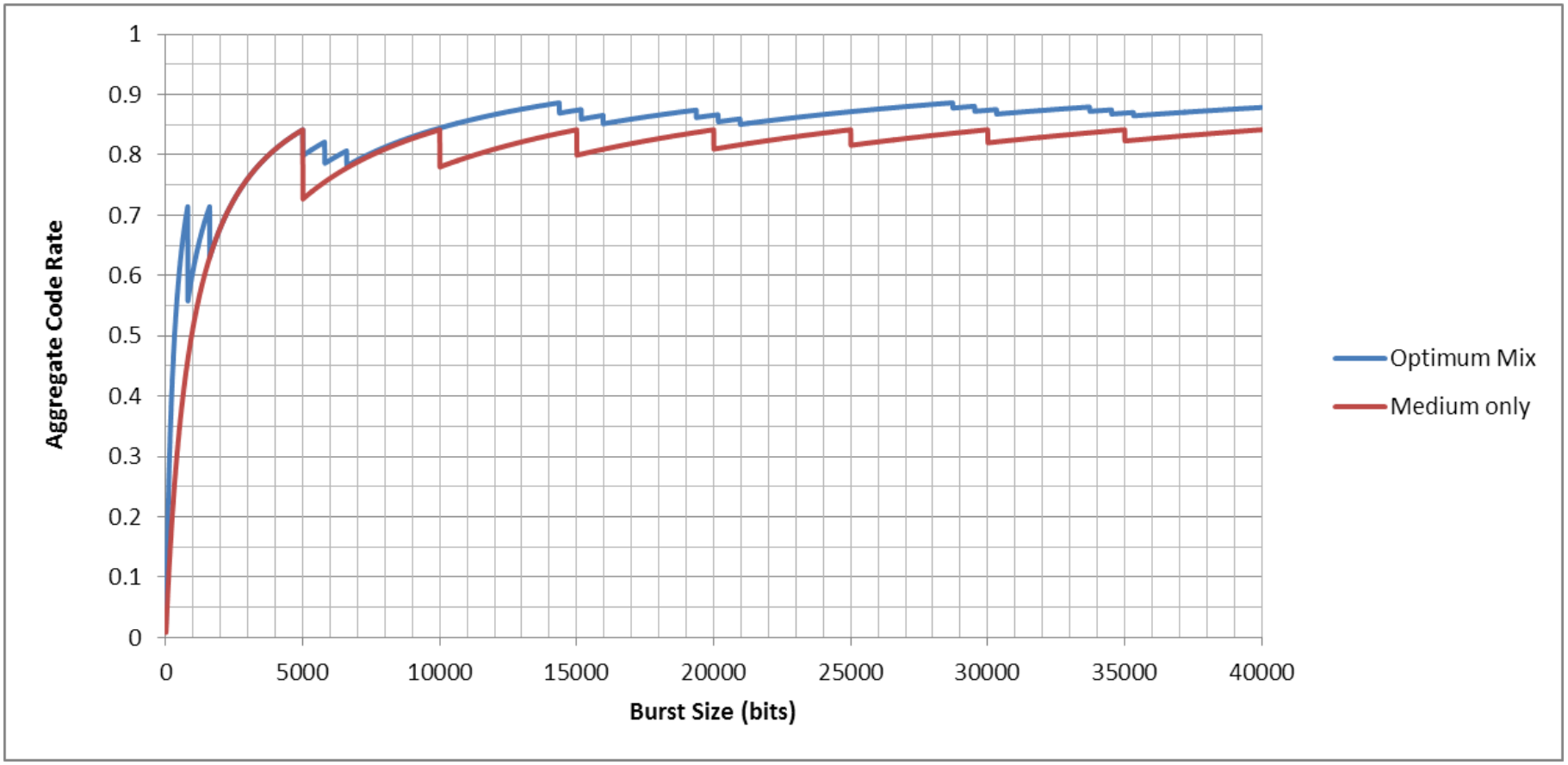
Efficiency (Code Rate) vs. Burst Size (Information Bits)



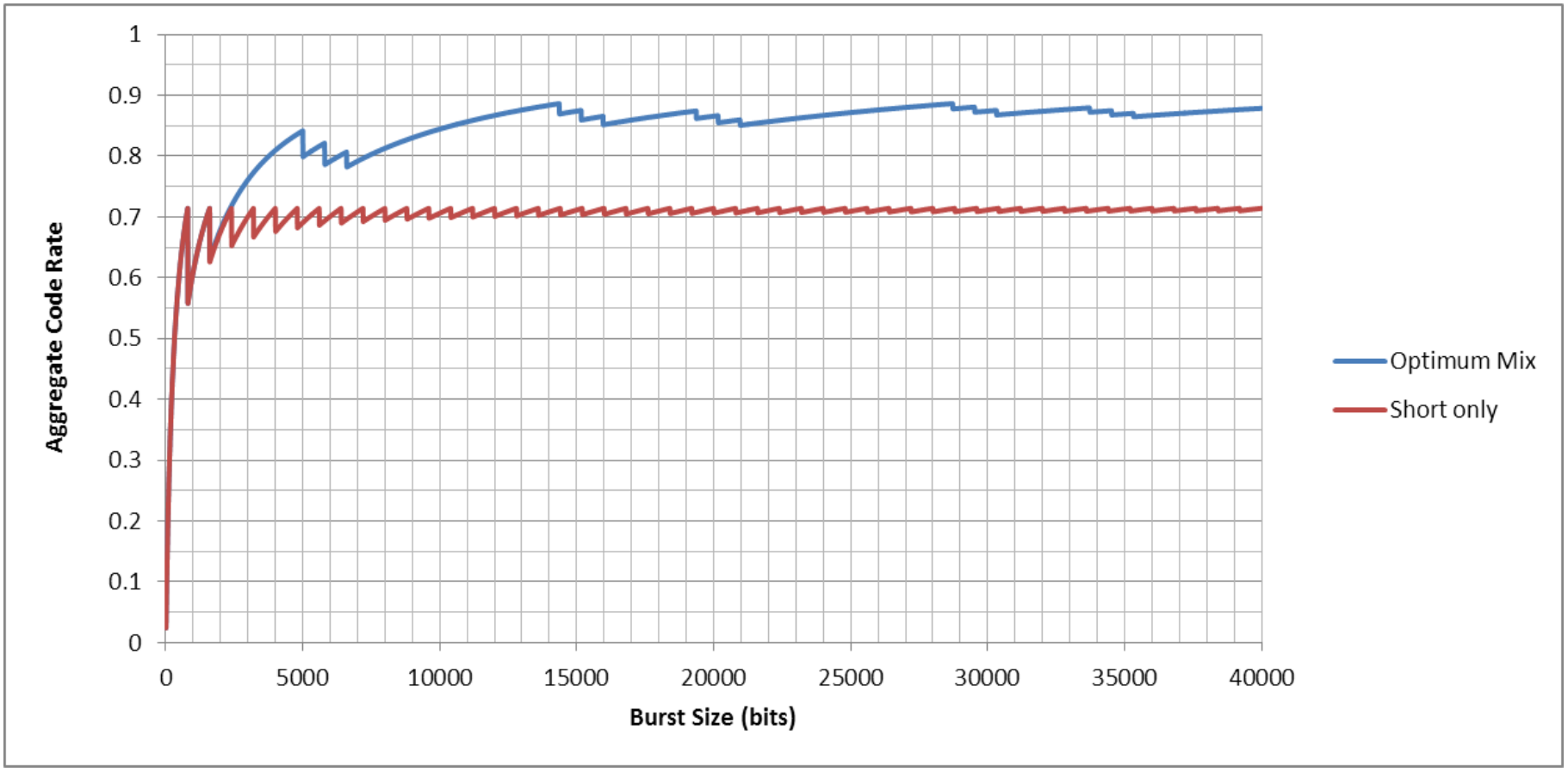
Efficiency (Code Rate) vs. Burst Size (Information Bits)



Efficiency (Code Rate) vs. Burst Size (Information Bits)



Efficiency (Code Rate) vs. Burst Size (Information Bits)



Optimum Shortening Problem

- The last shortened codeword may contain very few message bits, or even a single bit
- Such a last shortened codeword is transmitted in a much shorter time than a full codeword
- Such a shortened codeword needs to be decoded in a shortened time equal to the reduced transmission time
- This requires the decoder to operate at a much higher speed (and power) for this minimum size shortened codeword

Optimum Shortening Solution

- A solution to this last codeword shortening problem is to enhance the described algorithm with an additional step.
- Upon completion of filling the last shortened codeword, check if this codeword is at least half full of information bits. That is, check if the last codeword contains at least K/2 bits.
- If not, then move K/2 information bits from the next-to-last codeword into the shortened last codeword.
- This results in two last shortened codewords where each is at least half full.
- This limits the maximum decoding speed for a shortened codeword to less than twice that of a full codeword of the same type.
- Note that since moving bits from one codeword to another maintains the total number of message bits and the total number of parity bits, this does not change the aggregate code rate thus maintaining the optimum efficiency.

CONCLUSION

- **A method for selecting, combining, and shortening of multiple codeword sizes and rates that maximizes overall efficiency (aggregate code rate) is proposed**
- **The resultant efficiency (code rate) vs. burst size has been shown**
- **Loss of efficiency for single size codewords has been shown**
- **The problem of decoding a sparsely filled shortened last codeword is described**
- **A solution to mitigate the sparse last shortened codeword decoding problem is proposed which maintains the optimum efficiency**

PROPOSED MOTION

Move to:

Adopt the upstream codeword filling algorithm from prodan_3bn_01_1113.pdf for EPoC.

Moved: Richard Prodan

Second:

Thank You