Achieving a Target FFR with the Least Complexity in the Context of Convolutional Coding with an Optimized Cyclic Redundancy Check and the List Viterbi Algorithm

Ethan Liang, Hengjie Yang, Richard D. Wesel
Department of Electrical and Computer Engineering
University of California, Los Angeles

Fast Track to SUCCESS summer scholars program
Electrical Engineering Department

UCLA Samueli School of Engineering
Summer Undergraduate Scholars Program

Introduction and Motivation:

The development of the Internet of Things requires a short block-length code that guarantees each of the following: (1) low latency, (2) high reliability, and (3) low decoding complexity. List decoded convolutional codes offer a relatively low frame failure rate and decoding latency. A recent development allowing design of a cyclic redundancy check (CRC) that minimizes the probability of an undetected decoding error has renewed interest in list-decoded convolutional codes as a low-latency solution.

This work explores how complexity and performance in terms of undetected errors (UEs) and negative acknowledgements (NACKs) vary as the degree of the CRC and the number of states of the convolutional encoder (or the encoder constraint length) are varied.

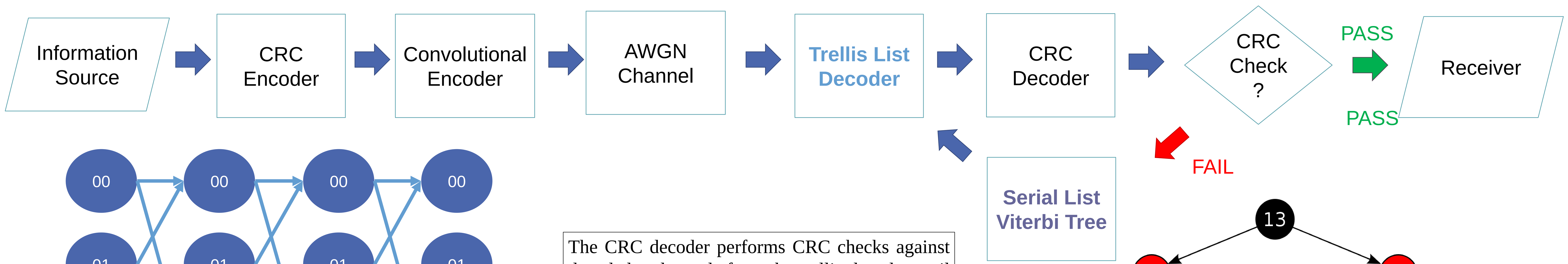
UEs and NACKs are collectively referred to as frame failures. The relationship between the frame failure rate, probability of UE, and probability of NACK are analyzed as a function of cyclic redundancy check degree, constraint length of convolutional code, signal to noise ratio, and maximum list depth for the serial list Viterbi decoding algorithm. An expression to quantify the complexity of the serial list Viterbi decoder as a function of maximum list depth is introduced. Given a target frame failure rate and probability of undetected error, the list Viterbi decoded convolutional code is optimized to achieve the target parameters with the minimum decoding complexity possible.

Materials and Methods:

C++ MATLAB

Coffee

Overview of List Viterbi Decoding:



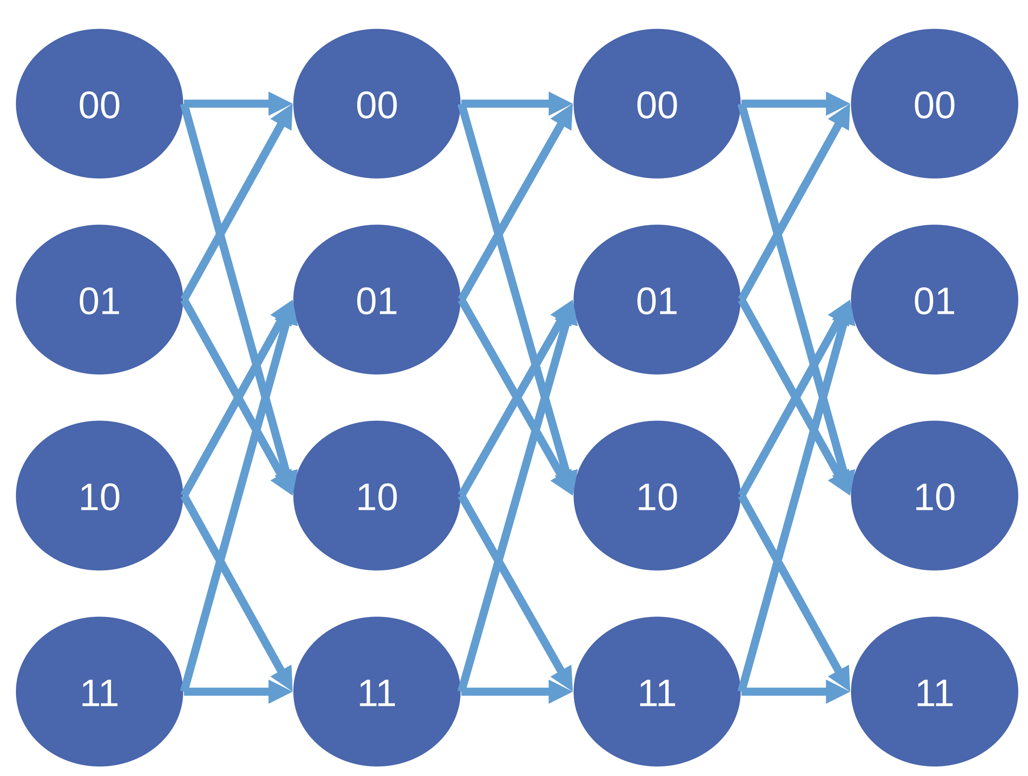


Fig. 1: Trellis List Decoder

The CRC decoder performs CRC checks against decoded codewords from the trellis decoder until one of two exit conditions is met:

1. The decoded code word passes the CRC check.
2. The maximum number of iterations (max list depth) is reached and decoder declares a NACK).

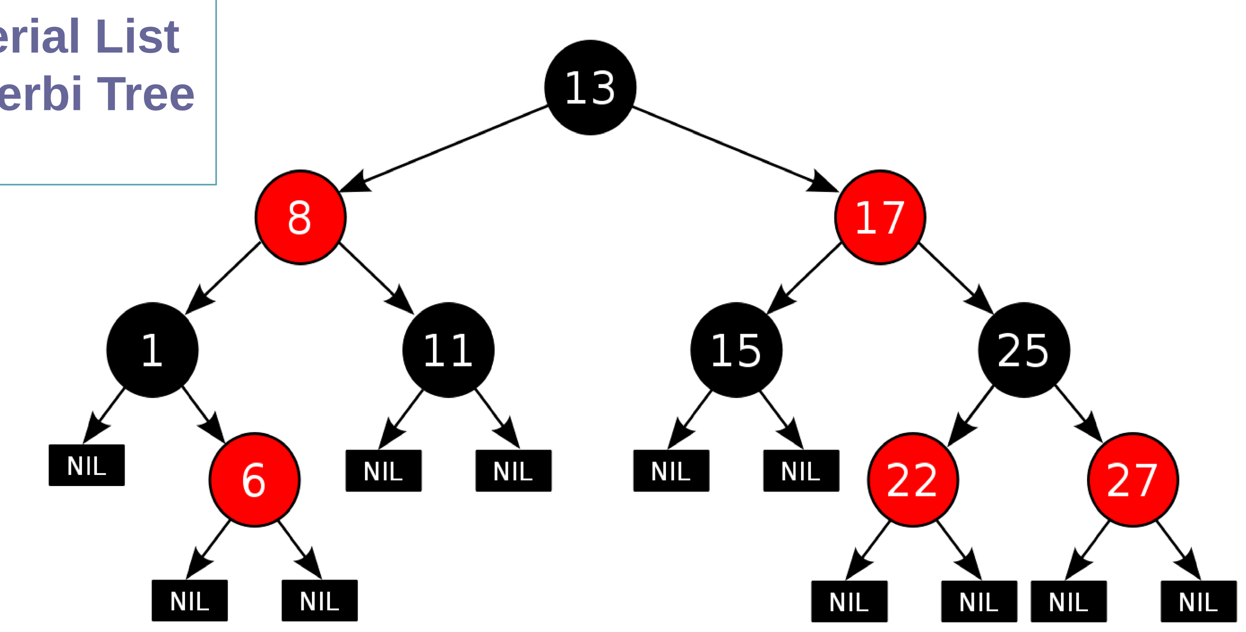


Fig. 2: Serial List Viterbi Tree (Red-Black Tree with Max List =10)

Results:

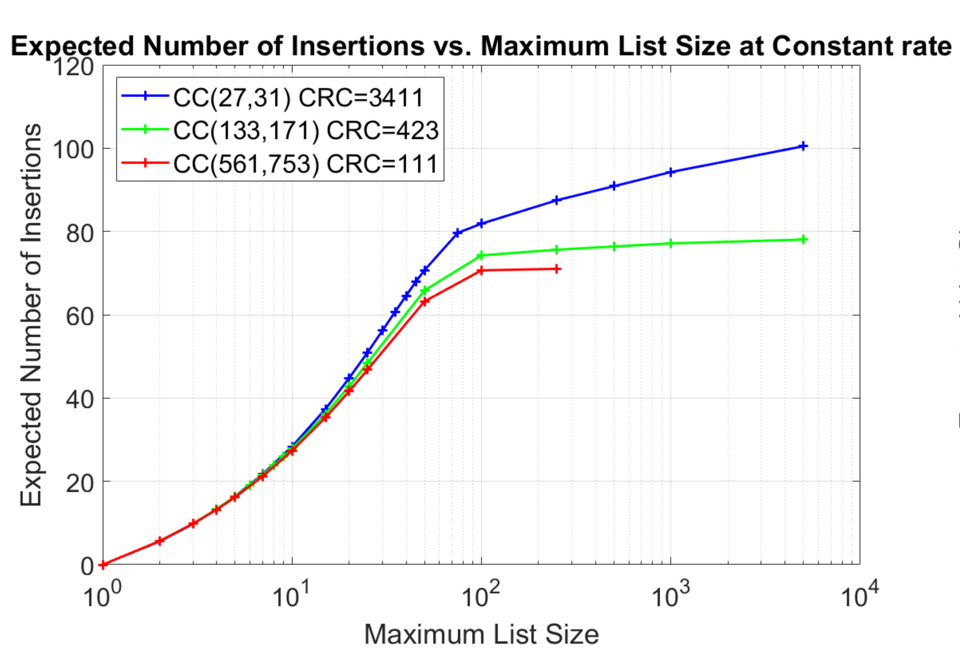


Fig. 3: The expected number of insertions into the minimum size list necessary for optimal decoding grows logarithmically with the maximum list size.

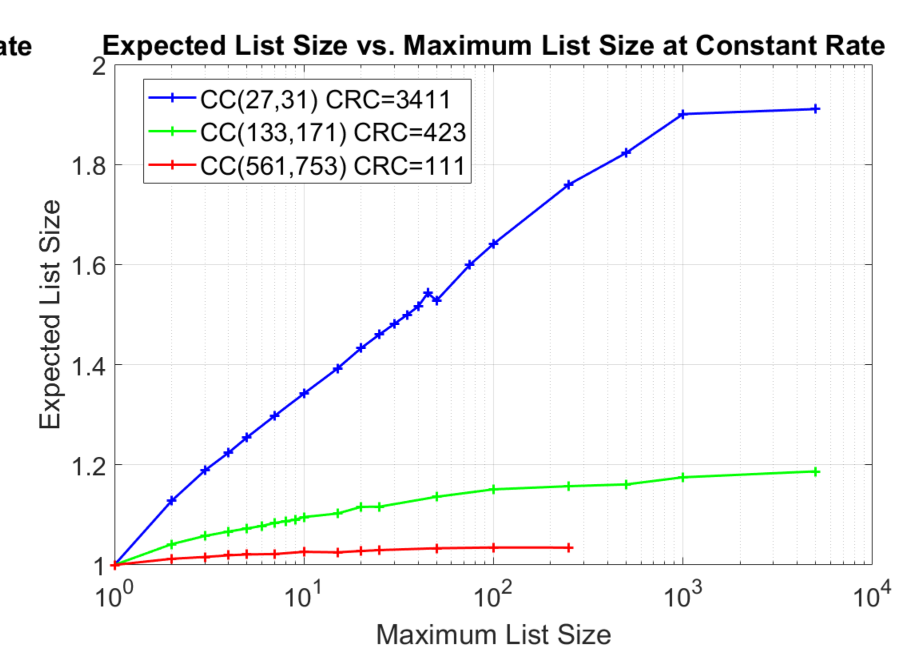


Fig. 4: The expected list size of the serial list Viterbi algorithm grows logarithmically with the maximum list size.

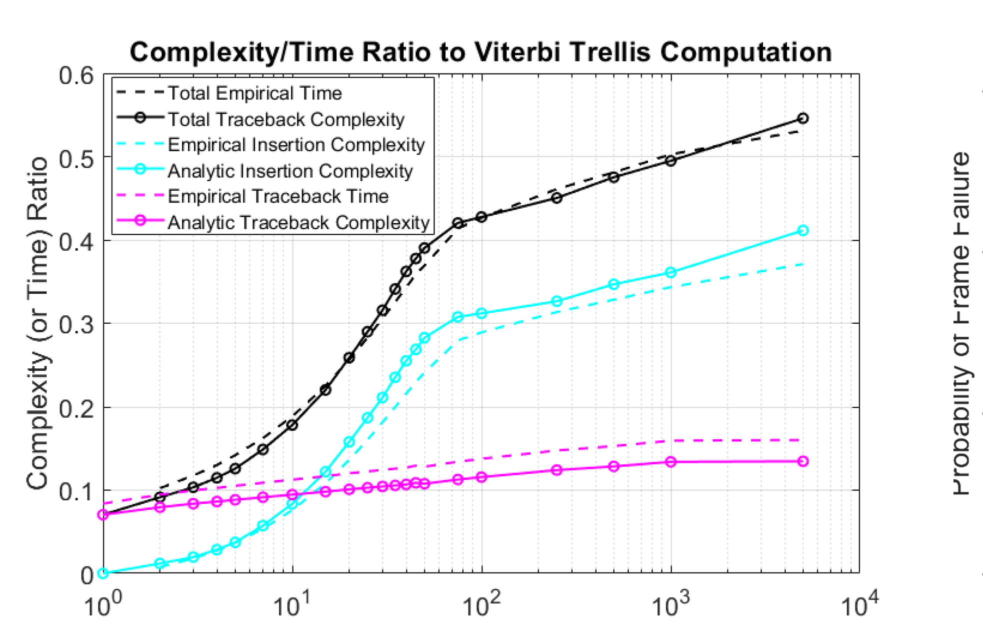


Fig. 5: The complexity expression, which describes the total number of arithmetic operations necessary to perform optimal serial list Viterbi decoding, can predict the additional time necessary to perform the list decoding. This ratio of total hardware operations in list Viterbi decoding to the total hardware operations in the soft trellis computation correlates very closely with the ratio of decoding times for both list Viterbi decoding and soft trellis computation.

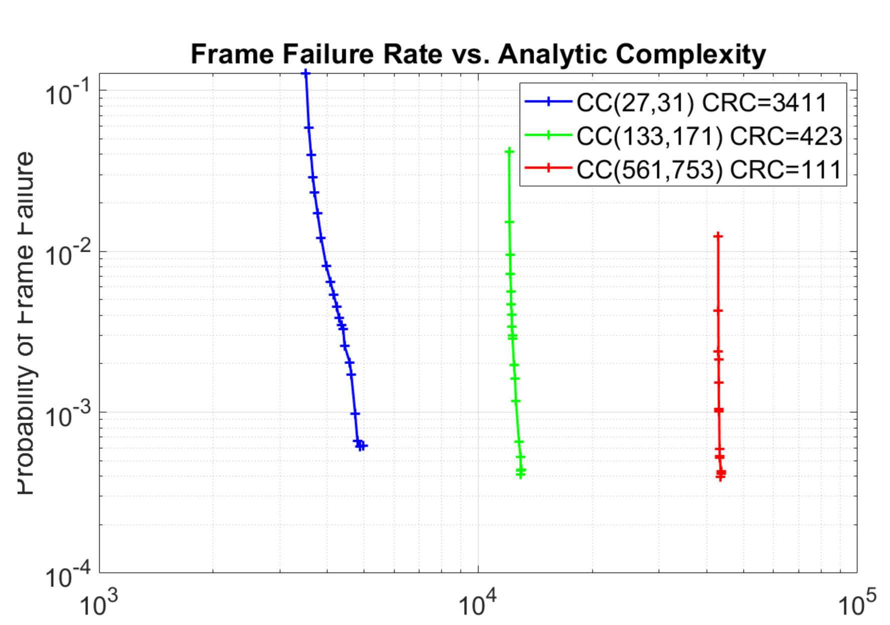


Fig. 6: A target FFR of 10^{-3} can be achieved with an order of magnitude less complexity by using a smaller convolutional code with a more powerful CRC. The complexity of the serial list Viterbi algorithm grows linearly with expected list size and quadratically with number of insertions. The expected list size and expected number of insertions grow logarithmically with maximum list size and reach a maximum value asymptotically, limiting the resulting complexity even with very large list depths.

Complexity of the Serial List Viterbi Algorithm:

Analytic Complexity

$$= [(k+m-2v)\times 2^{v+1} + (2^{v+1}+2^{v}+2^{v-1}+\cdots+4)] + [(k+m-2v)\times 2^{v} + 2^{v-1}+2^{v-2}+\cdots+1] + C_1\times E[L]\times\left[(k+m+v)\times 2+\frac{1}{2}m(k+m+1)\right] + C_2\times j\times \log_2(j)$$

Key:

- k - length of information sequence.
- v - number of memory elements in the convolutional code.
- m - number of CRC bits.
- j – expected number of insertions.
- $E[L]$ - expected list size.
- C_1 - a constant multiplier to adjust the complexity of the traceback operation.
- C_2 - a constant multiplier to adjust the complexity the insertion operation.

Note: C_1 and C_2 are hardware specific constants.

Analysis of the serial list Viterbi Algorithm complexity:

- The complexity of the serial list Viterbi algorithm is a function of the expected list size and the expected number of insertions. Choosing a maximum list size limits the complexity of the decoding process.
- The final complexity varies linearly with the expected list size. At each list depth, exactly one traceback operation and one CRC division must take place.
- The final complexity varies quadratically with the expected number of insertion operations. A red-black tree was used in the simulations to store the path metric differences.

Channel Parameters for all Figures shown:

- Signal-to-noise Ratio: 2dB
- Block Length (k): 64 bits
- Additive White Gaussian Noise (AWGN) Channel
- Number of CRC bits (m)
- Number of Memory Elements (v)
  - $m+v=14$

Conclusion:

- The serial list Viterbi decoding algorithm achieves a target frame failure rate and probability of undetected error for a given signal to noise ratio with less complexity than the soft Viterbi algorithm.
- The average complexity of the serial list Viterbi algorithm is a function of the expected list size and number of insertions, which can be designed by limiting the maximum list depth of the decoding algorithm.

Future Work:

- Model the codeword channel of the serial list Viterbi algorithm as an errors and erasures channel to compute its capacity in order to select the optimal CC-CRC pair.
- Extend this work to tail-biting convolutional codes. Develop complexity expression for a list decoded tail-biting convolutional code and calculate the associated frame failure rate and probability of undetected error.

References:

[1] H. Yang, S. V. S. Ranganathan, and R. Wesel, "Serial List Viterbi Decoding with CRC: Managing Errors, Erasures, and Complexity," *2018 IEEE Global Communications Conference*, Dec. 9-13, 2018, Abu Dhabi, UAE.

[2] J. W. Kim, J. W. Tak, H. y. Kwak and J. S. No, "A New List Decoding Algorithm for Short-Length TBCCs With CRC," in IEEE Access, vol. 6, pp. 35105-35111, 2018.

[3] M. Roder and R. Hamzaoui, "Fast tree-trellis list Viterbi decoding," in *IEEE Transactions on Communications*, vol. 54, no. 3, pp. 453-461, March 2006.

[4] N. Seshadri and C. -. W. Sundberg, "List Viterbi decoding algorithms with applications," in *IEEE Transactions on Communications*, vol. 42, no. 234, pp. 313-323, Feb-Apr 1994.

[5] M. Mohammad, H. Ramchandran, J. Jong, C. Ravishankar and C. Barnett, "Comparing List Viterbi Algorithms with and without tail bits," *MILCOM 2008 - 2008 IEEE Military Communications Conference*, San Diego, CA, 2008, pp. 1-6.

[6] B. Chen and C. -. W. Sundberg, "List Viterbi algorithms for continuous transmission," in *IEEE Transactions on Communications*, vol. 49, no. 5, pp. 784-792, May 2001.