Configurable FEC

David Van Wie, Adam Torgerson, R Chris Roark

June 2, 2017 / Revised Apr 18, 2018

Introduction

The ALERT2 Protocol includes a significant amount of forward error correction (FEC) in the AirLink layer, which allows for robust communication over marginal radio paths. This is an essential capability, allowing the protocol to be used in locations, such as steep canyons, where good radio paths do not exist and other means of communication simply do not work. However, this high level of error correction is applied uniformly, despite the fact that in many cases it is not needed. By reducing the amount of forward error correction applied in cases where the signal-to-noise ratio permits it, it is be possible to dramatically increase system capacity or decrease reporting latency.

The AirLink protocol includes two types of forward error correction which work together to provide both error correction and error detection. The two layers are *convolutional coding* and *Reed-Solomon error correction*.

Convolutional coding generates a series of parity symbols through the application of a boolean polynomial to a fixed-length window of the data stream. In the AirLink protocol, the convolutional coding algorithm generates two bits of parity data for every bit of data input, so the output stream is twice the size of the input stream. This coding system allows the recovery of individual bits in a noisy environment. Different coding rates (the ratio of input length to output length) can be achieved by "puncturing" the code, where some of the output bits are deleted according to a specific pattern.

Reed-Solomon error correction operates on a block of data, producing a number (t) of parity symbols that are appended to the end of the data block. The algorithm is then able to correct up to $t/2$ symbols in the data block. In the AirLink protocol, the symbol size is 8 bits (one byte per symbol) and t=16. This error correction algorithm is particularly well suited to correcting error bursts where a string of consecutive bits all become corrupt. For example, a sequence of nine consecutive bit-errors will affect at most two symbols. The Reed-Solomon error correction code also provides validation that the incoming message was received without errors.

In this proposal, we describe two additional FEC modes that use the same algorithms, with different parameters, to achieve higher throughput on links where the highest level of error correction is not required. The fundamentals of the AirLink protocol remain the same, so the addition of these modes adds minimal complexity to the protocol.

The new FEC modes have been optimized for use in a 250ms TDMA slot. Allowing systems to migrate from 500ms as the minimum slot length to 250ms would allow them to either increase the system capacity significantly, or reduce the TDMA frame length (and, thus, the reporting latency).

Proposed Updates

Summary

The proposed FEC modes are:

Table 1: FEC Modes

Mode	Max MANT size in 250ms slot	Max MANT size in 2s slot	Block Size (first / add'l)	Convolutional Coding	RS Parity Symbols	Rx Sensitivity
0	22 bytes	353 bytes	24 / 32 bytes	k=7, r=1/2	16	0dB
1	42 bytes	531 bytes	44 / 44 bytes	k=9, r=2/3	20	-1.5dB
2	52 bytes	646 bytes	54 / 54 bytes	k=9, r=3/4	20	-2.5dB

- Max MANT size assumes a slot delay of 25ms, CO time of 5ms, AGC time of 30, and a tail time of 5ms.

TDMA Slot Size

This proposal changes the minimum TDMA slot size from 500ms to 250ms.

Using the new error correction modes, it should be possible to fit most sites into a 250ms slot, cutting the effective channel utilization by a factor of 2.

API

The IND Configuration API specifies a command to set FEC mode at type ID 100. Under this proposal, valid values for that field would be 0, 1, or 2, reflecting the modes in the Table 1.

Frame Sync

In order to properly decode a message, the decoder needs to know the FEC mode that was used to encode the message. By choosing different frame synchronization sequences for each FEC mode, the decoder can determine which FEC mode was used to encode the data.

Receivers which are not capable of processing the variable FEC messages will continue to receive and process Mode 0 messages without modification.

Table 2: Frame Sync Sequences

Mode	Frame Sync Sequence
0	0x352E F853
1	0x5898 233E
2	0xEE43 4E88

The sequences for Modes 1 and 2 were chosen to ensure a larger number of bit differences between any two of the three sequences. There remain other Frame Sync Sequence values that have a similar number of bit differences to allow for future expansion.

Convolutional Coding

FEC Modes 1 and 2 both use a punctured convolutional code to achieve a lower coding rate than the original. Mode 1 generates 3 bits of output for every two bits of input, and Mode 2 generates four bits of output for every three bits of input. Because the puncturing removes some of the bits from the output, it is desirable to use a larger constraint length (sliding window size).

The generator polynomials used are specified in Table 3. See section 3.4.2 of the AirLink specification for a further discussion of convolutional encoding.

Table 3: Convolutional Coding Generator Polynomials

Mode	Polynomial A	Polynomial B
0	0x6d	0x4f
1	0x1af	0x11d
2	0x1af	0x11d

The convolutional code is punctured according to the following *puncturing matrix*:

Table 4: Puncturing Matrix

Mode	Code Rate	Matrix
0	1/2	1 1
1	2/3	1 0
		1 1
2	3/4	1 0 1
		1 1 0

The puncturing matrix is applied starting with the most significant bit of the transmission, column down then row accross. For Mode 1, this results in the removal of the third bit of every 4 bit sequence. For Mode 2, the third and sixth bits of every six bit sequence are removed.

In FEC Modes 1 and 2, if the output of the convolutional coding is not byte aligned, padding bits should be appended in the same manner that they are appended in Mode 0.

Block Size and Reed Solomon Coding

The Reed-Solomon block correction performs two functions for the system: error detection and error correction. A block of length t can correct up to $t/2$ symbols. The error detection capability of the RS code increases exponentially with the number of parity bytes in the block. Effectively, this means that using fewer than 16 parity parity bytes increases the potential error rate to unacceptable levels. The new FEC modes increase the number of parity symbols from 16 to 20, significantly decreasing the potential of false positives. This is especially important given that there is an increased probability of errors at the RS coding level due to the decreased convolutional coding rate.

Table 4 shows the different Reed-Solomon configurations, along with the probability that the algorithm accepts a block of random input as a valid message (Pr(Err)).

Table 5: Reed-Solomon Coding

Mode	RS Block Len	First Block Ratio	Second Block Ratio	Pr(Err)
0	16	0.67	0.5	2.2E-05
1	20	0.45	0.45	<< 1E-7
2	20	0.37	0.37	<< 1E-7

Other than the number of parity symbols and the data block length, the parameters for the Reed Solomon algorithm remain unchanged. See the AirLink specification, Section 3.4.1, for more details.

Analysis

An analysis of 9 months of data from the Harris county ALERT2 system confirms the practical benefits of these changes. During that time, one receiver (Transtar) collected 12.2 million MANT PDUs from 155 sites. The present system uses 500ms slots for the majority of these sites.

In order to move a site to new 250ms slot using FEC Mode 1 or FEC Mode 2, the following criteria need to be met:

- A sufficiently good radio path must be present that sacrificing 1 to 2db in coding gain is acceptable.
- The maximum message length transmitted by the sites must be less than 42 bytes (Mode 1) or 52 bytes (Mode 2).

In order to make the changes without the new variable FEC features, the maximum message length would fall to 22 bytes. In Harris County's system, 81% of MANT PDUs sent were 22 bytes or less, and 99% of the MANT PDUs were 42 bytes or less in length. However, there were no sites where the maximum message length was under 22 bytes.

Of the 155 sites in Harris County, 152 (98%) meet the criteria for moving to a 250ms slot. The other three sites each have a maximum message length that is longer than the 42 byte maximum for Mode 1.

Implementation of the variable FEC features would allow a system like Harris County to move nearly all of its sites from 500ms TDMA slots to 250ms TDMA slots, nearly doubling system capacity or cutting the reporting latency nearly in half.

Conclusion

The ALERT2 variable FEC proposal is able to provide significant, practical benefits to ALERT2 users without making significant changes to the core technologies. For a small cost in coding gain (typically less than 2.5dB) systems would be able to move from a 500ms TDMA slot to a 250ms TDMA slot, effectively doubling the channel capacity. Because the FEC algorithms themselves did not change significantly, the implementation cost for existing and new vendors of ALERT2 products should be manageable.