



# TECHNICAL COLUMNS

Official archives of articles and columns written by Ron Hranac for Communications Technology and some of its sister publications, published by Access Intelligence, LLC. Reprinted with permission of the author.

By **Ron Hranac**, former *Senior Technology Editor*, *Access Intelligence* and *Communications Technology Magazine*

Originally appeared in the **February 2011** issue of *Communications Technology*.

## WHAT'S A CODEWORD OR TWO AMONG FRIENDS?

By RON HRANAC

During Communications Technology's December 2010 "Did You Know?" Webcast, several participants asked about codewords, codeword errors and acceptable percentages of DOCSIS upstream codeword errors. A good way to answer these questions is first to define what a codeword is and how it applies to the world of data transmission over cable networks. All of this falls under the umbrella of something known as forward error correction (FEC), a combination of techniques and algorithms used to identify and fix data transmission errors. I'll start with a look at DOCSIS downstream FEC — the upstream is similar, but the downstream is somewhat easier to explain.

North American downstream DOCSIS transmission has as part of its foundation an International Telecommunication Union (ITU) standard known as ITU-T J.83, Annex B. Within that standard, one will find a description of the FEC used in DOCSIS, which comprises four processing layers. In a cable-modem-termination system's (CMTS's) quadrature amplitude modulation (QAM) modulator, the FEC encoder includes a Reed-Solomon (RS) encoder, an interleaver, a randomizer and a Trellis encoder. A cable modem's downstream QAM receiver FEC decoder includes a Trellis decoder, a de-randomizer, a de-interleaver and a RS decoder. Let's focus on the RS part of this. This tutorial is intentionally being kept at the one-fairly-weak-cup-of-coffee level, because FEC is a pretty gnarly subject that easily can fill a textbook.

As data enters a RS encoder, it is grouped into chunks called "blocks" or "codewords" or more specifically, RS blocks or RS codewords (I'll use the latter terminology throughout the remainder of this discussion). Each downstream RS codeword is made up of 128 RS symbols, and each RS symbol is 7 bits. (Refer to the accompanying figure, and note that RS symbols are not the same kind of symbols that make up each symbol point in a QAM constellation.)

"There is no perfect cable network that is completely free of data transmission errors."

Note that 122 of each RS codeword's 128 symbols are data symbols, and the remaining 6 are parity symbols used for error correction. Indeed, ITU-T J.83, Annex B states that the data is "...encoded using a (128,122) code over GF(128)...," which shows each RS codeword consists of 128 RS symbols (first number in first parentheses) and the number of data symbols per RS codeword is 122 (second number in first parentheses), leaving 6 symbols per RS codeword for error correction. DOCSIS downstream RS FEC is configured for what is known as "t = 3," which means that the FEC can fix as many as three errored RS symbols in a RS codeword.

What happens when there is, say, a burst of noise that causes a bit error or errors in one RS symbol? It doesn't matter to the RS decoder if one bit in that RS symbol is errored or if all seven bits are errored; the entire symbol is considered broken. There can be anywhere from three to 21 bit errors in a RS codeword that has three errored RS symbols. For a t = 3 configuration, as long as there are no more than three errored RS symbols in a given codeword, the FEC decoder in the cable modem can fix all of them. That would result in what's known as a "correctable FEC error" or "correctable codeword error." But as soon as the number of errored RS symbols in a given RS codeword is four or more, the entire RS codeword is toast. FEC can't fix three RS symbols and leave the remaining ones broken; if more than three RS symbols are errored, the entire RS codeword is errored. That codeword is now an "uncorrectable codeword," and we have an uncorrectable FEC error or uncorrectable codeword error!



DOCSIS downstream FEC is fairly straightforward, at least from the perspective of a fixed RS codeword size (128 RS symbols) and  $t = 3$  configuration. When DOCSIS 1.0 was introduced, upstream FEC supported codeword sizes ranging from 18 bytes (16 data bytes plus two parity bytes) to 255 bytes ( $k$  bytes + parity bytes), and  $t = 0$  to  $t = 10$ . DOCSIS 2.0 brought improved FEC, supporting  $t = 0$  to  $t = 16$ . The same basic principles apply in the upstream as is the case in the downstream: RS FEC can fix up to “ $t$ ” errored symbols (bytes) per RS codeword, but if the number of errored symbols per codeword exceeds “ $t$ ,” the entire codeword is considered uncorrectable.

Graphic representation of DOCSIS downstream RS symbols and codewords

Which brings me to a common question: What’s an acceptable percentage or number of codeword errors? That’s definitely an “it depends” question, but let’s start with an ideal goal: No correctable or uncorrectable codeword errors at all in either the downstream or the upstream. One could argue that, in a network performing so well that there are no codeword errors of any kind in the QAM signals, it would be possible to take advantage of eliminating FEC overhead altogether and picking up a little more throughput per channel. Of course, this goal is unrealistic. There is no perfect cable network that is completely free of data transmission errors. Still, the fewer codeword errors the better.

Let’s look at another metric for a moment. I have for several years suggested that upstream packet loss — that is, packet loss in just the outside plant’s upstream transmission path — be limited to no more than about 1 percent when the traffic is plain old high-speed data. If voice is thrown into the mix, upstream packet loss should not exceed about 0.1 percent to 0.5 percent, with the lower end of that range the better place to be.

How does upstream packet loss relate to codeword errors? That’s another “it depends.” First, if uncorrectable codeword errors exist, that means that packet loss is happening. However, uncorrectable codeword errors do not necessarily track lost packets on a one-for-one basis. For instance, one can configure upstream operation such that packet length equals the number of information bytes in a codeword. One also could have a configuration in which the packet length is longer than the number of information bytes in one codeword, but less than in two codewords.

It might be worthwhile to work with your network operations center folks to come up with a packet-loss estimate versus uncorrectable codeword errors in relation to the specific configuration used in your system. From there, target maximum packet loss goals similar to my previously suggested numbers. Once that’s done, start looking at trends over time.

Let’s say you find upstream packet loss to be less than 0.1 percent fairly consistently, but then you notice it starts to trend upwards to 0.1 percent, 0.2 percent, 0.25 percent and so on over some period of time. That upward movement in packet loss suggests something is degrading upstream performance, and it’s getting worse. Here, the trend itself is arguably more important than the actual packet-loss value, assuming the original value was reasonable to begin with.

From a practical perspective, tracking uncorrectable codeword errors should be considered one of many tools used to troubleshoot problems. Others include modulation error ratio (MER), a CMTS’s Flap List metrics and perhaps a third-party spectrum-monitoring tool.

Here are a couple examples of how evaluating more than one metric can better help the troubleshooting process. Let’s say the CMTS’s reported upstream MER (also called “upstream SNR”) is good, but uncorrectable codeword errors are out of whack. That may indicate impulse noise or laser clipping. Low MER and excessive uncorrectable codeword errors could indicate low carrier-to-noise ratio or maybe the presence of some significant linear distortions (micro-reflections, amplitude ripple, group delay). But back to tracking trends. One can certainly track raw correctable and uncorrectable codeword error numbers — and many cable operators do — but there are other ways to skin this cat. The following example is excerpted from one of my columns written back in 2003:

According to Motorola's Marc Belland, using upstream uncorrectable codewords is one of the best ways to complement the CMTS's [MER] estimate. Says Belland, "I know of several operators who use the following three management information base (MIB) values to calculate, using a script, what can be called codeword error ratio (CER) on their CMTS upstream ports. They are:

- docsIfSigQUnerrored
- docsIfSigQCorrected
- docsIfSigQUncorrectables

"A total of all three divided into the uncorrectables number will give you a CER at that point in time. Subsequent polls at user specified intervals—for example, every five minutes—can give you a trend as to whether the upstream is getting better or worse. Obviously if it's getting worse you'll need to pro-actively get someone to attend to the issue, depending on the rate of upstream degradation."

Belland provided the following example, which could be ported to an Excel spreadsheet:

- unerrored: 1,562,456
- correctables: 803,867
- uncorrectables: 209,134
- total: 2,575,457
- loss %: 8.120%
- CER = 8.12E-02

You'll find other useful background information and ideas in an older white paper, authored by my colleague John Downey, called "Upstream FEC Errors and SNR as Ways to Ensure Data Quality and Throughput," available at [www.cisco.com/en/US/tech/tk86/tk319/technologies\\_white\\_paper09186a0080231a71.shtml](http://www.cisco.com/en/US/tech/tk86/tk319/technologies_white_paper09186a0080231a71.shtml). An updated version of the paper is available but not online. Contact John directly at [jdowney@cisco.com](mailto:jdowney@cisco.com) to get a copy of the newer version.

Ron Hranac is technical leader, HFC Network Architectures, for Cisco Systems, and former senior technology editor for *Communications Technology*. Reach him at [rhranac@aol.com](mailto:rhranac@aol.com).