



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 **October 2007** issue of *Communications Technology*.

A CLOSER LOOK AT BER, PART 1

By **RON HRANAC**

I'm occasionally asked about BER, one of several performance metrics used in the world of data transmission. Quadrature amplitude modulation (QAM) analyzers commonly report BER, as do some digital set-tops and cable modems. BER is an abbreviation for bit error rate or bit error ratio, the latter actually being more technically correct. BER is the estimated probability that a bit transmitted through a device or network will be received incorrectly. For instance, if a 1 is transmitted and it's subsequently received as a 0, the result is a bit error. BER is measured or estimated by transmitting some number of bits and counting the number of incorrect or errored bits received at the other end.

In general, BER is the ratio of errored bits to the total number of bits transmitted, received or processed over a defined length of time. Mathematically, two formulas are often used to describe BER:

$$\text{BER} = (\text{number of errored bits}) / (\text{total number of bits})$$

$$\text{BER} = (\text{error count in measurement period}) / (\text{bit rate} \times \text{measurement period})$$

Examples

Here's an example using the first formula. Let's say that 1 million bits are transmitted, and three bits out of the 1 million bits received are errored because of some kind of interference between the transmitter and receiver. The BER is calculated by dividing the number of errored bits received by the total number of bits transmitted: $3/1,000,000$ or 0.000003 . We can further express 0.000003 in scientific notation format - the way most BER measurements are shown. Scientific notation is nothing more than a shorthand method of expressing very large or very small numbers. Our example of 0.000003 is written in scientific notation as 3×10^{-6} . If your e-mail, computer, or word processing program can't display superscript characters used for exponents, it's acceptable to represent scientific notation in the format $3 \times 10^{\wedge}6$. The small " \wedge " sign simply means that the -6 after the \wedge is an exponent (10 raised to the -6 power). Another variation is to write scientific notation in the form $3.0E-06$, which means the same thing as 3×10^{-6} .

Here are a few examples of numbers expressed in scientific notation form:

$1,000,000 = 1 \times 10^6$ or $1.0E06$
 $100,000 = 1 \times 10^5$ or $1.0E05$
 $10,000 = 1 \times 10^4$ or $1.0E04$
 $1,000 = 1 \times 10^3$ or $1.0E03$
 $100 = 1 \times 10^2$ or $1.0E02$
 $10 = 1 \times 10^1$ or $1.0E01$
 $1 = 1 \times 10^0$ or $1.0E00$
 $1/10$ or $0.1 = 1 \times 10^{-1}$ or $1.0E-01$
 $1/100$ or $0.01 = 1 \times 10^{-2}$ or $1.0E-02$
 $1/1,000$ or $0.001 = 1 \times 10^{-3}$ or $1.0E-03$
 $1/10,000$ or $0.0001 = 1 \times 10^{-4}$ or $1.0E-04$
 $1/100,000$ or $0.00001 = 1 \times 10^{-5}$ or $1.0E-05$



1/1,000,000 or 0.000001 = 1×10^{-6} or 1.0E-06

...and so on.

Pre- and post-FEC

What about pre- and post-FEC BER? These just state the BER before forward error correction (FEC) attempts to fix the errored bits and the BER after the FEC fixes as many broken bits as it can. In a typical QAM analyzer, pre-FEC BER is estimated after the Trellis decoder, descrambler (derandomizer), and deinterleaver, but before Reed-Solomon error correction. Post-FEC BER is after Reed-Solomon error correction.

Here's a simplified explanation: Let's say that in our previous example of three errored bits received out of 1 million bits transmitted, the receiver's FEC is able to fix only two of the three errored bits. The pre-FEC BER is, of course, the "raw" BER of 3×10^{-6} - that is, the BER before FEC attempts to fix the three broken bits. The post-FEC BER is 1×10^{-6} , since there is still one errored bit after the FEC did its magic on the other two broken bits. A QAM analyzer would show these values as 3.0E-06 and 1.0E-06 respectively. Ideally there should be no post-FEC errors, so if they show up, you need to find out why and fix the problem.

A quick side note: I chatted with industry friend and colleague Bruce Currivan about bit errors and FEC, and he offered the following. In reality, a QAM receiver's Reed-Solomon decoder (this is the FEC part of the receiver) can easily fix the three bit errors in the previous example, although it is still a good illustration of the principle involved. In fact, the Reed-Solomon decoder in a DOCSIS cable modem can fix any three errored Reed-Solomon symbols in a codeword. This capability is commonly expressed as " $t = 3$ ". Each Reed-Solomon symbol is a group of seven bits. A Reed-Solomon codeword or block consists of 128 Reed-Solomon symbols, of which 122 are actual data symbols, and six are "parity" symbols that allow for error correction. It doesn't matter to the decoder if one bit is wrong in a symbol or if all seven bits are wrong; the symbol is still considered wrong. So in three errored Reed-Solomon symbols, there can be anywhere from a total of three to 21 bit errors. Thus, the Reed-Solomon decoder can correct up to 21 bit errors in a codeword, depending how the bit errors are grouped.

It is possible for the Reed-Solomon decoder to fail to decode. Let's say there are four input bit errors in a Reed-Solomon block, spread out with each bit error in a single Reed-Solomon symbol. This is more than $t = 3$, so the decoder can't correct the errors. It usually detects this condition and outputs the data unchanged. (The decoder should do it this way, depending on implementation.) What does this mean? The output will include all four bit errors! If the decoder doesn't know it couldn't fix the block, it may try to fix it anyway and output even more errors than four. This is less probable since most of the time the decoder can detect the errors. But for weak Reed-Solomon decoders - $t = 3$ is pretty weak, $t = 8$ is better and is more common - the probability of not detecting the error condition is not negligible. So the Reed-Solomon decoder output may be bursty.

All of this makes BER more complicated than it needs to be when Reed-Solomon decoders are involved. Indeed, some prefer codeword error rate (CER) over BER in DOCSIS high-speed data.

Interesting bits

Here's where things get interesting. If we've measured a BER of 3.0E-06, is that really the BER? It's beginning to look as if the answer might fall into the category of "it depends."

First, would you expect a \$40 cable modem to report BER as accurately as a \$3,000 QAM analyzer? What about the QAM analyzer's accuracy compared to a \$120,000 lab-grade BER test set? Clearly, receiver quality is important and may impact reported BER.

One major consideration is the number of bits transmitted during the measurement. The greater the number of bits, the better the quality of the BER estimation. Ideally, an infinite number of bits will give us a perfect estimate of error probability, but that simply isn't practical. OK, how many bits are enough? One rule of thumb is that transmitting three times the reciprocal of the specified BER without an error gives 95 percent confidence level that the device or network meets the BER spec. If one wants 99 percent confidence level, the multiplier is 4.61 rather than three. The multipliers are derived from some gnarly statistics math involving binomial distribution function and Poisson theorem. These numbers strictly apply to independent bit errors, and as we have seen previously, the errors at the output of a Reed-Solomon decoder are not independent. But they are still a good illustration of the concept involved.

Next month I'll continue this discussion and also look at some practical examples of typical causes of degraded BER and even false BER.

Ron Hranac is technical leader, HFC Network Architectures, for Cisco Systems, and former senior technology editor for *Communications Technology*. Reach him at rhrnac@aol.com.