BER measurements reveal network health

Martin Rowe - July 01, 2002

Whether you design communications equipment, develop tests for production, install network subsystems, or maintain networks, you need to make bit-error-rate (BER) measurements. BER measurements give you insight into the health of an entire network; a network subsystem such as a switch, router, multiplexer, or hub; or a network component such as an electrical or optical transmitter or receiver. The measurements also provide benchmarks that network users require from communications service providers as part of quality-of-service agreements.

BER measurements tell you how many bit errors occurred for a given number of bits that passed through a network, subsystem, or component. That history lets you assign an error probability to each bit.

Error sources

A bit error occurs when an electrical or optical receiver makes an incorrect decision about a bit's logic level. Many factors can contribute to BER. They include signal power, noise, jitter, and EMI from radiated emissions or crosstalk. Noise at the point where a receiver decides the logic level of a bit can cause the receiver to misinterpret that bit. Poor signal-to-noise ratio (SNR) and poor extinction ratio (the average power level of logic-1 bits compared to the average power of logic-0 bits) make it easier for random noise to cause bit errors.

Jitter also causes bit errors. A change in clock timing can cause a receiver to lose synchronization with incoming bits. A rising or falling edge that occurs when a receiver doesn't expect it can also cause the receiver to incorrectly interpret a logic level. Errors caused by jitter can occur periodically or with certain bit patterns only, depending on the cause of the jitter.

Radiated emissions from oscillators such as those in switching power supplies and microprocessor clocks can produce anomalies that cause bit errors. Because those oscillators produce clock signals, the errors they produce often occur at regular, periodic intervals.

Crosstalk occurs when a portion of a signal's power leaks between adjacent transmission media and is more prevalent in electrical transmission systems than in optical transmission systems. In electrical signals, the high frequencies contained in signal edges can couple into adjacent circuits. The interference can produce a level shift that can cause a receiver to misinterpret a bit.

BER numbers

Even with so many potential error sources, today's networks often must perform with BERs of $10^{-12}$ or lower—rates so low that they're difficult to measure. A BER of $10^{-12}$ translates to one bit error for every $10^{12}$ bits (1000 Gbits) that arrive at a receiver. BER numbers represent a probability that a
receiver erroneously interpreted a bit, so a $10^{-12}$ BER means that every bit has a $10^{-12}$ probability of being in error.

Although a $10^{-12}$ BER sounds small, at 10 Gbits/s, that's one error every 100 s (1 min, 40 s). When you look at BER that way, you can see why a low BER is so important. Too many bit errors translates into poor voice quality, lost data, or retransmissions of data that produce unacceptable data throughput (Ref. 1).

What defines an acceptable BER? Standards provide BER performance limits, but network users have demanded BER levels well below the limits specified in standards. The O.15x series of recommendations published by the International Telecommunication Union-Telecommunications Sector (ITU-T; www.itu.int) specifies standards for BER measurements (Ref. 2). Other standards that specify BER measurements include ANSI T1.510 (Ref. 3), ANSI/IEEE 1007 (Ref. 4), and ATIS Technical Report 25 (Ref. 5).

ITU-T recommendation O.150 specifies that testers measure BER at rates from $10^{-3}$ to $10^{-8}$, a range considerably higher than the $10^{-12}$ that service providers often require. Recommendation O.150 is, however, six years old and was written when telecom networks carried more digitized voice traffic than data traffic. Now that data traffic exceeds voice traffic, communications service providers—not standards—dictate a network's transmission quality. Equipment designers and test engineers must produce products that meet those requirements. When installed in networks, each piece of network equipment typically must perform at a BER of $10^{-14}$ or lower so that a network can maintain a $10^{-12}$ BER.

**BER testers**

To verify the design of a network, a network subsystem, or a network component, you need a BER tester. A BER tester contains a pattern generator and an error counter. The counter counts all incoming bits, compares each bit to a reference pattern, counts the errors, and uses the following equation to calculate BER:

$$\text{BER} = \frac{\text{Number of bit errors}}{\text{Number of bits received}}$$

To measure BER, you must first send a known bit pattern through your transmission medium or system under test. You need a known pattern because a tester's error counter needs to know what bit pattern to expect. Otherwise, the tester can't determine a good bit from an error.

You can't send a simple repeating pattern like 10101010. Sending such a pattern doesn't simulate the type of traffic a network will see during normal operations. You can't send longer patterns such as 00001111 either. This pattern looks like a clock signal at one-fourth of the bit rate. The receiver's clock-detection circuits may mistakenly synchronize to that pattern.

To help you test a network, subsystem, or component, standards such as O.150 specify several pseudorandom bit sequences (PRBSs) that provide all combinations of 1's and 0's within a specified number of bits. The patterns must be long enough to simulate random data at the data rate being tested. If the pattern is too short, it will repeat rapidly and may confuse the clock recovery circuitry, which will prevent the receiver from synchronizing on the data.

**Test patterns**

Table 1 shows the different bit patterns, the standards that define them, and the communications
circuits that use them. The bit patterns follow a $2^N-1$ format. So, a $2^9-1$ pattern contains all 511 possible 9-bit combinations except for all logic 0's. The sequence repeats only after the pattern generator has sent all possible combinations.

The O.15x recommendations specify the bit-pattern lengths based on the speed of the communications circuit. The faster the link, the longer the bit pattern, although you can use some bit patterns at more than one data rate.

Figure 1 shows how a BER tester's pattern generator develops the bit patterns. For simplicity, the figure uses a 4-bit pattern. A "sliding window" 4 bits wide moves through a 15-bit pattern and produces all possible 4-bit combinations, except 0000, repeating every 15 bits. The pattern generator takes those patterns and produces a serial bit stream.

To perform a BER test, you need to send test patterns to a UUT and measure BER at its output. Figure 2 shows two setups for BER measurements—end-to-end and loopback. In Figure 2a, the pattern generator and BER tester reside in different locations, possibly miles apart. In Figure 2b, the pattern generator and BER tester reside at the same location, sometimes in the same test set. The test signal loops back through the network, subsystem, or component under test to the test set, which calculates BER.

End-to-end tests typically occur in network installations where you need to test a network's BER performance. Loopback tests more often occur in a design lab or in a production test area. Loopback tests require fewer pieces of test equipment because one BER tester can both generate patterns and count errors. You also get immediate test results. With either test setup, you select a bit pattern based on customer requirements or standards.
Test pattern length

When you run a BER test, you don’t simply send one set of PRBS patterns and measure the errors. Doing so might not produce any errors, leaving you with a false sense of security. When verifying a design, you must run BER tests long enough to get a realistic statistical probability that a bit error will occur. To accurately measure BER, you should run a test until your equipment detects at least 100 bit errors. That will ensure a statistically valid sample of data. Test times on high-speed circuits and systems typically run 8 to 10 hrs, but can reach 72 hrs. The longer you run a test, the more reliable your BER measurements will be.

Unfortunately, testing every subsystem or component in production for that long costs money, so manufacturers use techniques for reducing test time. A test known as "Q factor" attempts to estimate a network’s, subsystem’s, or component’s BER in minutes rather than in hours or days. But you lose accuracy because the Q-factor technique tries to predict the probability of bit errors by measuring received signal strength and noise levels rather than by counting actual errors. In using the Q-factor method, you also must assume a random occurrence of bit errors.

![Figure 3](image.png)

**Figure 3.** The area where the curves overlap represents the probability of a bit error.

Figure 3 shows the basis for Q-factor measurements. You need to build histograms of the average power levels for logic 1 bits and logic 0 bits. In this figure, the halfway point between the logic 1 and logic 0 average peak power levels defines the receiver's decision threshold. The area under the intersection of the curves represents the probability of an error. You can describe Q factor with the following equation:

\[
Q \text{ Factor} = \frac{\text{Difference in means}}{\text{Sum of standard deviations}}
\]

Therefore, a smaller difference between peaks (average logic levels) reduces Q factor. A wider histogram representing the average power levels (larger standard deviation) also reduces Q factor.

In a Q-factor measurement, you alter a receiver's decision threshold, which increases the probability of bit errors. The increased probability of errors results in shorter measurement time. Using at least two BER levels, you can find the mean and standard deviation of the error histograms. To perform the procedure, you must move a receiver's decision threshold first toward one average logic level and then toward the other. Move the decision threshold toward logic 1 until you measure a BER of 10⁻⁹. You can make these measurements far more quickly than you can measure a BER of 10⁻¹².

Continue moving the decision threshold until you measure BER at 10⁻⁶. Next, move the decision
point until the BER reaches 0.5, which occurs at the peak of the curve. Repeat the procedure for the other logic level. You now have the points you need to calculate the mean and standard deviation of the curves for both logic levels, with which you can calculate the Q factor. You can then estimate BER at the halfway point with the following equation (Ref. 6):

$$\text{BER} \approx \left[ \frac{1}{Q\sqrt{2\pi}} \right] \left[ e^{-c^2/2} \right]$$

Using the Q factor, you can cut test time, but in using the technique, you assume the probability that errors will occur is a Gaussian distribution. If periodic or pattern-dependent errors occur, the Q-factor method might not work. The only way to get a high confidence level for BER is to measure BER with a sufficient number of bits. Therefore, you may still have to test a network, subsystem, or component the long way, particularly in the design lab so you can accurately characterize a design.

Table 1. Bit patterns used for BER measurements in communications circuits

<table>
<thead>
<tr>
<th>Pattern (length) of:</th>
<th>Specified in Standard</th>
<th>Maximum number of zeroes</th>
<th>Commonly used to test circuit rates</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^n - 1$ (511)</td>
<td>ITU-T O.150</td>
<td>8</td>
<td>Up to 14.4 kbits/s</td>
</tr>
<tr>
<td>511 QRS (Note 1)</td>
<td>ANSI T1.510</td>
<td>7</td>
<td>56 kbits/s</td>
</tr>
<tr>
<td>$2^n - 1$ (2047)</td>
<td>ITU-T O.150</td>
<td>10</td>
<td>64 kbits/s, Nx64 kbits/s</td>
</tr>
<tr>
<td>$2^n$ QRS (2048)</td>
<td>ANSI T1.510</td>
<td>7</td>
<td>56 kbits/s</td>
</tr>
<tr>
<td>$2^n - 1$ (32,767)</td>
<td>ATIS TR25</td>
<td>14</td>
<td>1.544 Mbits/s, 2.048 Mbits/s, 34 Mbits/s, 45 Mbits/s</td>
</tr>
<tr>
<td>$2^n - 1$ inverted (32,767) (Note 2)</td>
<td>ITU-T O.150</td>
<td>15</td>
<td>1.544 Mbits/s, 2.048 Mbits/s, 34 Mbits/s, 45 Mbits/s</td>
</tr>
<tr>
<td>$2^n - 1$ (1048575)</td>
<td>ITU-T O.150</td>
<td>19</td>
<td>34 Mbits/s, 45 Mbits/s, 139 Mbits/s</td>
</tr>
<tr>
<td>$2^n - 1$ QRSS (1048575) (Note 1)</td>
<td>ITU-T O.150</td>
<td>14</td>
<td>1.544 Mbits/s</td>
</tr>
<tr>
<td>$2^n - 1$ (8388607)</td>
<td>ATIS TR25</td>
<td>22</td>
<td>34 Mbits/s, 45 Mbits/s, 139 Mbits/s, 155 Mbits/s, 622 Mbits/s, 2.4 Gbits/s, 10 Gbits/s</td>
</tr>
<tr>
<td>$2^n - 1$ inverted (8388607) (Note 3)</td>
<td>ITU-T O.150 and ANSI/IEEE 1007</td>
<td>23</td>
<td>34 Mbits/s, 45 Mbits/s, 139 Mbits/s, 155 Mbits/s, 622 Mbits/s, 2.4 Gbits/s, 10 Gbits/s</td>
</tr>
<tr>
<td>$2^n - 1$ (2.147*10^n) (Note 3)</td>
<td>ITU-T O.151</td>
<td>30</td>
<td>2.4 Gbits/s, 10 Gbits/s, 40 Gbits/s</td>
</tr>
<tr>
<td>$2^n - 1$ inverted (2.147*10^n)</td>
<td>ITU-T O.150</td>
<td>31</td>
<td>2.4 Gbits/s, 10 Gbits/s, 40 Gbits/s</td>
</tr>
</tbody>
</table>

Notes:
1. QRS stands for "quasi-random signal" and QRSS stands for "quasi-random signal source." Both patterns find use in T1 (1.544 Mbits/s) electrical transmission systems.
2. $2^{15}-1$, $2^{23}-1$, and $2^{31}-1$ use inverted patterns specified by the ITU-T; the BER tester inverts its output before transmitting the test pattern. Most modern test equipment offers all patterns in the normal and inverted formats, which accounts for the inclusion of the noninverted $231-1$ pattern in the table.

3. For the higher bit rates (2.4 Gbits/s to 40 Gbits/s), the $2^{31}-1$ pattern should be your first choice with the $2^{21}-1$ pattern as second choice. The longer pattern repeats less often and causes fewer problems for the clock recovery and fiber-optic receive circuits.

References

1. "What is BERTS and BER?" Anritsu, Richardson, TX.

Acknowledgements

Thanks to Mike Lewis of Acterna (Germantown, MD) for compiling Table 1. Thanks also to Hamish Butler and Malcolm Paterson of Agilent Technologies (South Queensferry, Scotland), Gary Mading of gnubi communications (Dallas, TX), Stéphane Le Gall of NetTest (Créteil, France), and Mike Botham of Tektronix (Chelmsford, MA) for their technical reviews of this article.