VIAVI Solutions

White Paper

Testing Network Devices with the Xgig Jammer

Abstract

Quality — a concept easy to grasp but, ironically, hard to quantify. In a car it may be the way its doors sound when slammed, in a suit perhaps the way it flows when the wearer moves. With network devices it is most likely how well its developers have 'sweat the details'; not only how well the devices perform their function, but how well they respond to errors both commonplace and rare. While most developers test their devices extensively, they cannot gauge the device's true quality until it is exposed to the worst-case conditions of real-worlds networks. Unfortunately, these errors are often difficult to create in a laboratory. Design engineers are discovering a new tool that can test network components under such conditions — the Xgig Jammer. Exclusively from VIAVI, the Jammer is an important test tool when network devices have completed their initial testing, but need more testing to make sure they are capable of handling conditions they will face in networks. By manipulating real-world network traffic, the Jammer provides an array of error cases that allow an accurate view of how devices will react to these situations. This paper describes the Xgig Jammer and its usefulness in testing network devices' readiness, so developers can be sure they will provide the quality their customers expect.

Jamming Improves Designs and Shortens Development Cycles

In the competitive networking market, quality and timely delivery are fundamental to success. Any tool that improves quality, or saves design cycle time, delivers a competitive advantage.

The Jammer delivers the following value to your design process:

More Robust Designs – The Jammer enables you to test almost any "what if" error scenario that you can conceive, allowing testers to find and fix problems if the system does not react as planned. In this way, the Jammer supports a more robust and thoroughly tested network or component.

Faster Development Cycles – With the Jammer, you can test and verify your design more rapidly than any other tool, consequently streamlining and expediting your development cycle.

Design Confidence – By testing out all possible error conditions, the Jammer gives you the confidence of knowing your design will work.

Stronger Customer Relationships – Fast turnaround on an accurate tested network or component design is the best way to build relationships with customers and ensure that they keep coming back to you for all their networking needs.

Testing Error Response with the Jammer

The VIAVI Xgig Jammer is a test tool designed to help developers ensure that network devices recover from error conditions without data loss or corruption. The Jammer provides controllable and repeatable network traffic modification in real time to introduce errors into the network and verify recovery processes operate properly.

The Jammer is best used later in the design cycle — at the time when the design has stabilized and the device under test has started to transport data and send commands back and forth, but when the developer needs verification that the device will be able to recover from the myriad of possible network errors it may encounter. There are few ways to do this testing early in the design cycle. To accomplish it, you must have a working network running real traffic and you must be able to observe the response, in real time, to the errors introduced.

Jammer testing is an extension to early stage testing devices such as the BERT (Bit Error Rate Tester) or traffic generation tools. While these are essential, they only test the quality of the physical layer and simple commands early in the development cycle. BERT traffic that comes back without errors after going through the device under test confirms that your physical connections are sound. But the BERT does not give you any information about how the device will react to errors. Traffic generators do provide error response information, but usually only for the simplest error cases.

The Jammer is part of a comprehensive testing process. With the Jammer, it is easy to create errors, resulting in a wide range of real-world error cases. This allows testers to identify and solve problems related to the introduced errors. The Jammer shortens the testing of your network devices and does so more thoroughly. So when your device goes into production, you will have confidence that it is ready.

When the Jammer is used in conjunction with the VIAVI Xgig Analyzer, developers can capture, display, and examine the response of network devices to the errors introduced by the Jammer. Designed to work together, the Jammer can start data collection in the Analyzer (Jammer triggers Analyzer), and the Analyzer can start the Jammer (Analyzer arms Jammer). Both the Jammer and Analyzer can share templates for conditions to arm and trigger.

When the Jammer is used in conjunction with the VIAVI traffic generation products, developers can get a head-start on "what-if" situations. By utilizing standard test libraries in the generator, and modifying them with the Jammer, testers can perform variations on standard tests. This strategy ensures that products not only perform as they should, but also perform well to common problems.

The Xgig Jammer supports both black-box and white-box test strategies. For black-box testing, the Xgig Jammer can be configured to cause a wide range of potential error conditions. Under normal operation, the operating environment should trap, handle, and then report the actions the system took for each of these errors. If an error is not trapped or handled correctly, the Xgig Jammer can be configured to redo the specific test. The details of the traffic are captured by the Xgig Analyzer for examination.

Using the Xgig Jammer for a white-box test strategy requires a two-stage process. First, the tester analyzes the error checking software in the device and determines the expected response to specific classes of errors. Next, the tester exercises all error checking routines by creating specific errors with the Xgig Jammer and injecting them into the system to verify that the system responds correctly.

Examples: Injecting the Jammer into Your Testing Strategy

The Jammer normally acts as a retimer on a connection between two network devices (inline) that simply passes data between the two devices. Traffic passes through the Jammer until it detects a preconfigured specified event or sequence of events. The Jammer then makes a user-defined modification to a specified frame, network primitive (ordered set), or series of primitives. The following sections detail a few of the more common modifications performed by the Jammer.

Injecting a Bit Error

One of the most common network errors devices must overcome is the simple bit error. While a bit error may have no effect on a device if it occurs during a non-frame transmission, bit errors within a frame must be detected and corrected in order to guarantee data fidelity.

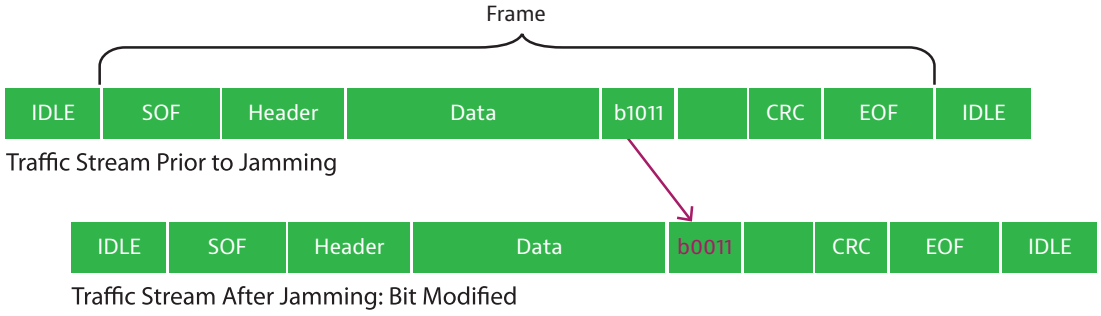

Figure 1: Bit Error Introduced into Frame

Injecting a bit error is surprisingly difficult for test engineers. Most network devices encode data as a final step just prior to network transmission. As this encoding is usually done in hardware, it is not an easy task for software to create this error condition. As it resides on the line between network devices, the Jammer provides a straightforward way to create a bit error in a frame — either in a random frame or in a particular frame of the designer's choice. (Figure 1)

Cyclic Redundancy Check (CRC) Modification

A common way for network test equipment to detect bit errors is through the use of a data frame's CRC. The device can determine if a frame's contents are good or corrupt using a mathematical formula to compute a CRC and then comparing it with the CRC included with the frame. Without the Jammer, testing this detection mechanism is often difficult, requiring special programming in the sending device's hardware to allow the tester to modify the CRC value. However, with the Jammer, testers can modify the CRC to any value of their choice without hardware modifications. Furthermore, the Jammer allows testers to modify the CRC of a specific frame of their choosing, allowing for detailed and controlled error introduction. (Figure 2)

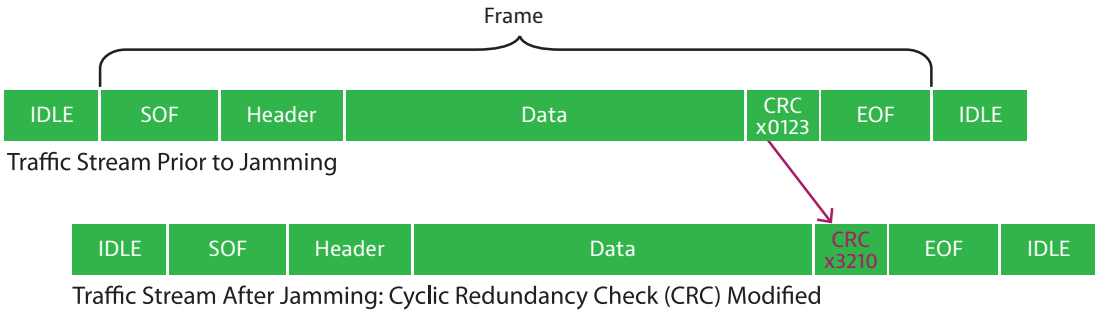

Figure 2: CRC Modification

On the flip side, if testers wish to corrupt the information within a frame, and have the network equipment pass this error up the protocol stack, they are often frustrated as modifying the frame invalidates the CRC. The Jammer can optionally correct the CRC within frames containing manipulated data. This ability provides a way for network devices to receive modified frames and not discard them as containing errors.

Frame Modification

A specific frame or series of frames can be modified in many ways such as replacing a frame with idles, truncating a frame, or changing any bits within a frame. In doing so, the Jammer allows testers to modify a normal frame and fabricate a bad frame designed to simulate an error.

A common error mechanism involves a frame that does not arrive at its intended destination because it was discarded due to an error detected by an intermediate device. This can cause difficulty for a destination device reassembling frames into their higher-level protocol constructs. The Jammer provides a way for testers to make frames "disappear" (replace the frame with idles) to test this situation, either randomly or when matching a predetermined template. (Figure 3)

Summary

The Jammer delivers a comprehensive set of late stage development error testing abilities. In addition to the features described above, the Jammer also provides the following advantages:

Consistency – The Jammer always produces consistent Fibre Channel and Gigabit Ethernet traffic without disparity errors or clock discontinuities.

Transparency – The Jammer is transparent to the network, acting like an additional length of cable, and functions in any Fibre Channel or Gigabit Ethernet architecture.

Cost-Efficiency – Using the VIAVI multi-function blade technology, a VIAVI Xgig Analyzer blade can be changed into an Xgig Jammer. This allows one blade to perform two functions at a lower total cost.

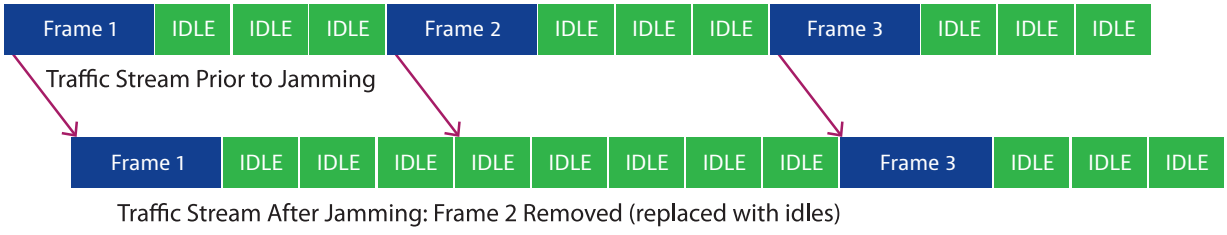

Figure 3: A "Disappearing" Frame (Replaced by Idles)

VIAVI Solutions

Contact Us **+1 844 GO VIAVI**
(+1 844 468 4284)

To reach the VIAVI office nearest you,
visit viavisolutions.com/contact

testnwdxgigjammer-wp-san-tm-ae
30162799 500 0809