A Testing Method For Wide Receivers in SMPTE ST 2110-21

by Francois Gagnon, Daniel Tremblay, Renaud Lavoie

Introduction

This document explains a testing methodology on a wide receiver, defined in the SMPTE ST 2110-21 specification. This specification describes and defines the traffic shaping of uncompressed video. In other words, it defines how the uncompressed video sender should send video packets over the network. Although it might sound obvious, we should state upfront that there are two different sender types. The narrow type is an isochronous type of sender. It is predictable (usually a hardware-dedicated engine is responsible for sending the packets) and will send packets at a constant pace. The second type is the wide sender. This type of sender will send packets in order but not in a predictable manner. The figure below (Figure 1.) shows the two types of senders.

We will consider here a network where the switches can buffer packets, add delays, and send out the packets in a wide sender manner. During the various installations we carried out ourselves, we did not see this behavior, but because Riedel IP products are based on ST 2110-21, any impact created by the network will be accepted.

A live network will always have some challenges. Packets might be delayed or corrupted, arrive in the wrong order or get entirely lost on the way. The line may be completely down at times. Also, there may be times with bursts of traffic or network overload. In order to recreate these live network conditions to test robustness, we carry out something called IP impairment.

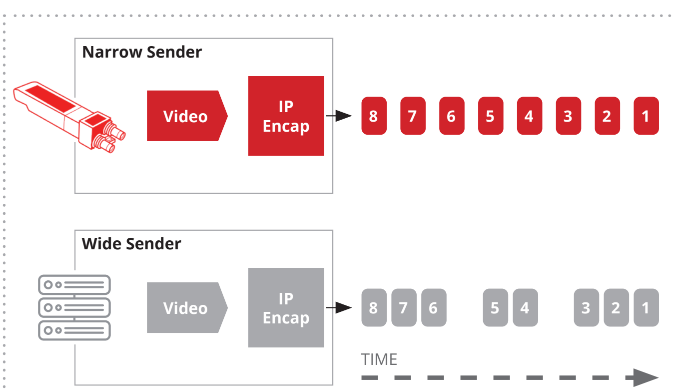

Figure 1. Narrow vs wide sender

In addition to the above-mentioned challenges, the ST 2110-21 sender can be a linear or gapped sender. So what is the difference between these senders? The term gapped defines a sender which usually takes an SDI or HDMI signal with blanking. During blanking, the sender can stop sending the packets (gapped) or can spread the packets in a linear manner (linear). The following figure (Figure 2.) shows a gapped vs a linear sender.

For simplicity's sake, we will assume that narrow and wide senders can be both types: gapped and linear. Now let's see how we tested our ST 2110-21 products against the wide transmitter. Many of our customers have asked us how we can guarantee that a wide receiver will work. This is what we're going to explore in this white paper.

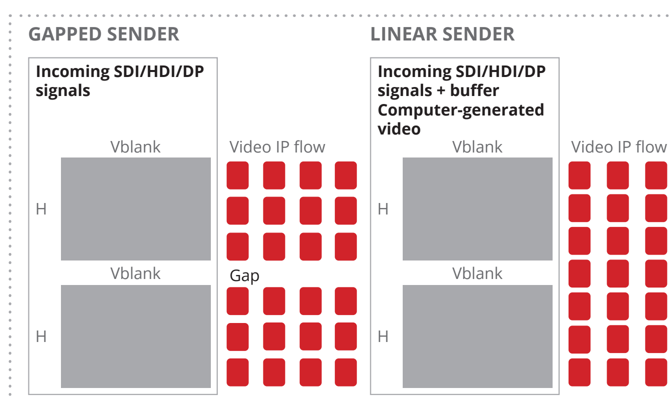

Figure 2. Gapped vs linear sender (example with narrow sender)

1. Definitions

For starters, let's define the different types of senders and receivers.

1.1 Narrow Sender

A narrow sender is an isochronous type of sender, meaning that it is predictable and sends packets at a fixed interval, usually around 7 usec from packet to packet (for HD signals). This process is similar to USB packet transmission. In short, it is a sender that sends packets at a fixed pace with a fixed spacing between them.

1.2 Wide Sender

A wide sender will send packets in order, but not in a predictable manner. The packets can have large gaps between them, they can be grouped together, and can be bursty. In fact, the wide sender needs to be PTP-aware and needs to respect the frame rate. It may seem like this scenario is a bit chaotic, however, ST 2110-21 established some boundaries to ensure that all this works. In particular, ST 2110-21 ensures that the gaps between packets are manageable and don't add massive delays that would be unacceptable in a production environment. Again, for this reason, ST 2110-21 specified some boundaries, such as a buffer minimum of 720 packets, which is recommended in the wide receiver. For this reason, Riedel IP products are equipped with configurable buffer size – from a few packets up to 4096 packets of buffer for HD, and Full HD streams. UHD buffer size is the size of one UHD frame.

1.3 Delay

There are multiple blocks that create delay in a network, such as serialization delay, propagation delay and queuing delay in network devices (sources or switches). As we already know, in video, delay can be troublesome for real-time applications. Delay is also present when the network includes low-speed links. To properly assess the delay experience, test equipment has been created, which we will discuss in more detail later in this paper.

1.4 Impairment

As with delay, impairment at the physical layer is another stressor that needs to be evaluated when rolling out a new network, application or technology. When bit level causes packet corruption or leads to a packet being dropped by the switches, the video receiver should be able to deal with these problems. If we use a TCP-based application, the retransmission will slow down application throughput. In a UDP-based system, when there is loss of integrity, the most obvious effect is loss of video, audio or metadata. To remedy the impairment, we use ST 2022-7 in combination with ST 2110 to ensure we always have a path of valid data.

1.5 Narrow Receiver

A narrow receiver is a receiver with limited buffer size. It should be used in combination with a narrow sender, otherwise no video will be readable after the IP transport. Narrow receivers are useful in a production environment, but our approach is to enable the customer to configure our core receiver to become a narrow or wide receiver with an extremely deep buffer – up to 15 ms of buffer per output (for HD signals), depending on the application and environment.

1.6 Wide Receiver

A wide receiver is a receiver that works with wide and narrow senders (transmitter). It has the buffer to handle isochronous senders and non-isochronous senders. As mentioned earlier, to avoid extra delay when used in a production environment, the Riedel wide receiver has a configurable buffer depth (Figure 3.).

1.7 Riedel Core Receiver

The Riedel IP core receiver used in all our MuoN, FusioN and VirtU solutions is built with a large buffer of 4096 packets per channel for SD, HD and 3G, and 16384 packets per channel for UHD. The following images show MediorNet FusioN 3B in 25G mode with single HDMI 2.0 (Figure 4.) and dual SDI output (Figure 5.). A Fusion 6B could provide dual HDMI 2.0 and quad SDI output.

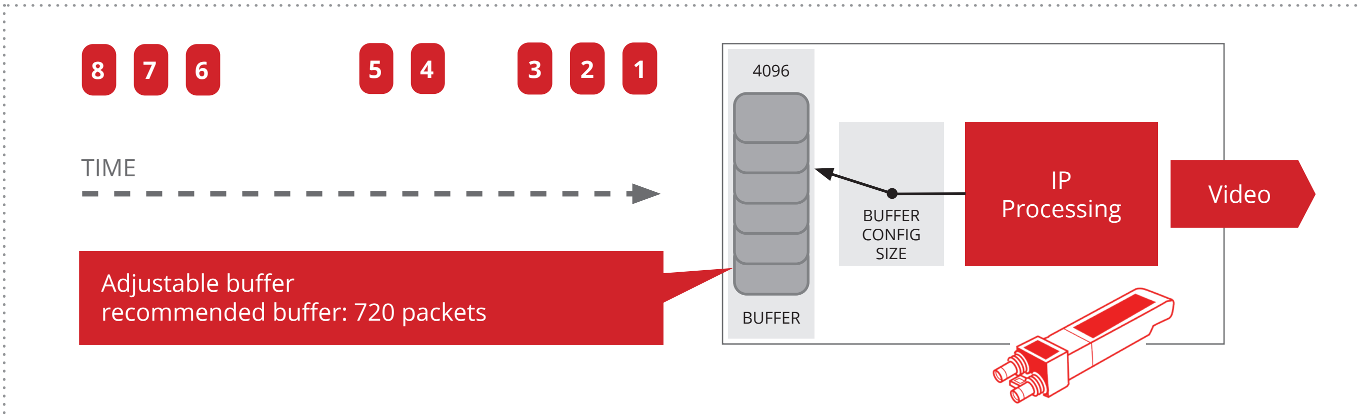

Figure 3. Wide receiver with configurable buffer size

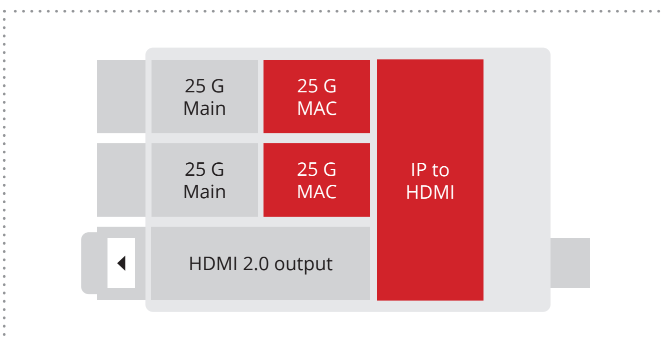

Figure 4. FusioN 3B HDMI details

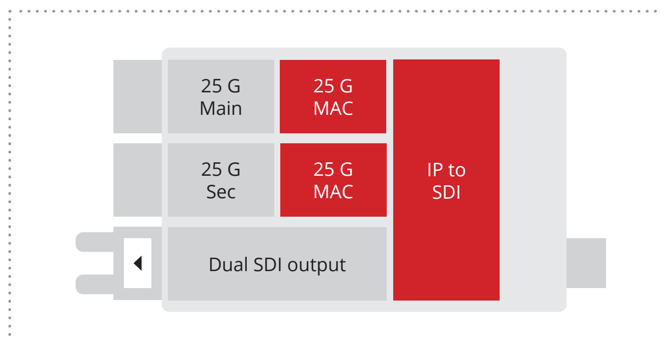

Figure 5. FusioN 3B SDI details

2.0 TESTING METHODOLOGY

Ethernet and IP are widespread technologies that we can't imagine living without these days. The challenging factor of Ethernet and IP technologies is that they can sometimes be unpredictable. A network can deteriorate a flow by introducing packet delay, packet loss, out-of-sequence packets, duplicated and corrupted packets. These issues are usually introduced by the buffering done by switches and routers. The other common problem is network congestion, which can cause delay or even packet drop. Additionally, packet reordering can be caused by multipath networks.

Engineers use different network impairment methods to emulate real-world scenarios to test their system or devices. These tools can introduce network delay or jitter, allowing users to validate that their system can work in any situation. Jitter is the delay difference between packets, and, in real-time applications is a big concern. The introduction of jitter on a narrow sender signal will produce the same effect as if it were a wide sender – in other words, it will make the flow bursty.

As video and media move to IP, these network impairment methods mentioned above will be useful for testing in production and post-production networks.

2.1 Network Delay and Impairment Validation Method

Note: Riedel does not claim this is the only way to validate a wide receiver in ST 2110-21, but it is the method that works well for us.

The following image shows the actual setup for the test. One or multiple senders is inserted in a top of rack switch. Instead of connecting the TOR switch directly to the core, we add the appliance that will help us introduce delay, jitter and all the other impairments we discussed previously. The core is then connected to the other TOR switch that hosts the de-encapsulators. The de-encapsulators will see the flows affected by the switches, but most importantly the impairment emulator. The SDI outputs are then connected to an analyzer to validate the output throughout the tests (see Figure 6.).

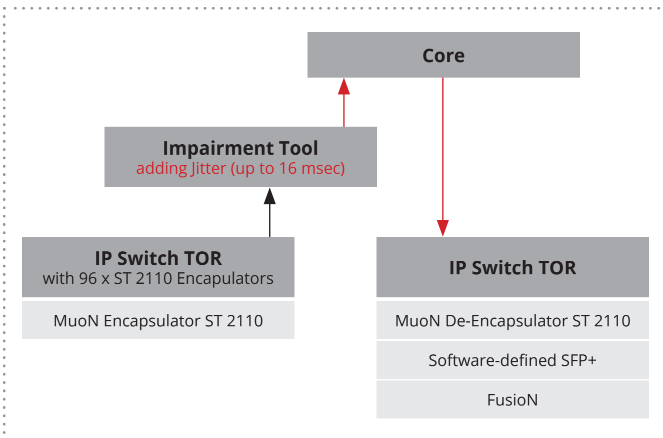

Figure 6. Testing setup

The impairment can be applied to a specific RTP flow or to all traffic. To apply to a certain flow, a rule is created based on parameters such as the VLAN or the destination IP. It can be useful to test the different essences separately and even more practical to validate the redundancy (ST 2022-7). The Riedel team did validate both the non-hitless path and the hitless path (ST 2022-7).

To test that the buffering works correctly on the de-encapsulator, one introduces a jitter from the minimum value accepted by the equipment to the configured buffer size of the device under test. Commonly the minimum is in the 10 microseconds range, but this differs from equipment to equipment. In our case, the tool allows us to do everything from a very small jitter variation to a big one.

Another test that can be executed is accumulating the packets in the impairment appliance buffers and then bursting them. This test will again test the buffer, but instead of controlling the delay, one controls the number of frames directly. Using this method allows us to test both the buffer size as well as the capacity for the device to cope with a large burst.

Now that we have proven that the buffer is meeting the expected values, it can be tested to see if the device is able to handle out-of-order packets. To configure this, we usually specify how many packets we want to reorder in one sequence and how often. The number of packets needs to be selected based on the configured buffer of the receiver, just like the jitter test. It is also a good idea to validate that the device is able to handle the duplicate packets. This is configured the same way as the packet out-of-order test. The only difference here is that we don’t need to exceed medium bandwidth when duplicating packets.

Our final test is a mix of all the tests above to validate that the receiver is rock solid. Here, we’ll need to adjust the value so as not to exceed the buffering capacity.

Once we have confirmed that the receiver conforms to the specifications, we can check if the device can recover from exceptional scenarios. These tests validate that the receivers can recover successfully from buffer overflow or underflow, as well as packet loss and corruption.

To test buffer overflows, one can configure the tool to burst many packets, which will exceed the buffer of the receiver. After the buffer is exceeded, we can remove the impairment and see if the receiver recovers correctly. Testing buffer underflows is done by configuring a long delay, letting the receivers starve for a while, then removing the delay and analyzing what we recover.

Finally, we want to confirm that we are not greatly affected by packet drops and corrupted packets. The objective of this test is to make sure that the output continues to play without much impact on the user experience when we have some packet loss. Again, once the impairment is removed, the receivers should be back on their feet immediately. Note that the corrupted packets should be dropped by the switches and might not reach the device being tested at all. Depending on the type of decapsulator used, one might want to connect the output of the impairment tool directly to the device, or as close as possible.

All Riedel IP products have been tested with all the above and successfully handled all the impairments.

2.2 Different Types of Impairment

JITTER

ACCUMULATION AND BURSTING

OUT-OF-ORDER PACKETS

PACKET DUPLICATION

PACKET DROP

PACKET CORRUPTION

3. CONCLUSION

In conclusion, ST 2110 receivers must be carefully designed to support wide senders (ST 2110-21). Receivers must also be carefully tested to eliminate problems introduced by wide senders or any problems stemming from network impacts (impairment or delay). In a production environment, it is recommended to rely on narrow senders as far as possible in order to keep the buffer size to a minimum.