MPEG-2 VIDEO STREAMS ON A DATA NETWORK
(or What Happens when Video gets the Jitters)

William Garrison
Motorola, Inc., Broadband Communications Sector

Abstract

This paper will review the issues involved in using standard wired and wireless IP data networks to carry MPEG-2 Transport Streams. Transport within the headend and within the home will be covered. IEEE 802.11a/b/g, HomePlug, 100BaseT Ethernet, Gigabit Ethernet and proprietary IP network protocols will be covered. The difference between carrying Constant Bit Rate and Variable Bit Rate streams will be examined. New revisions of data networking standards are including provisions for carrying video traffic. Will these enhancements be enough? What can we do until they are here?

INTRODUCTION

MPEG-2 Transport Streams as used in broadcast networks were designed to be delivered as a continuous flow of bits on a HFC (Hybrid Fiber Coax) plant. This is quite different from what you find on a data network, where bit streams are broken into packets and then sent in discrete bursts.

There also is another fundamental difference between data and video traffic. Data delivery is focused on accuracy while video delivery is focused on timeliness. Video decoding is designed to conceal errors while data transfers require perfection. Can both of these coexist on the same network?

Although there are other potential issues when carrying MPEG-2 Transport Streams on a data network, this paper will focus on the issue of jitter. Also this paper will focus on MPEG-2 Transport Streams as traditionally carried on HFC plants and decoded by standard settops.

What is Jitter?

For MPEG-2 Transport Streams, jitter is the "*short-term non-cumulative variations of the significant instants of a digital signal from their ideal positions in time.*"[1]

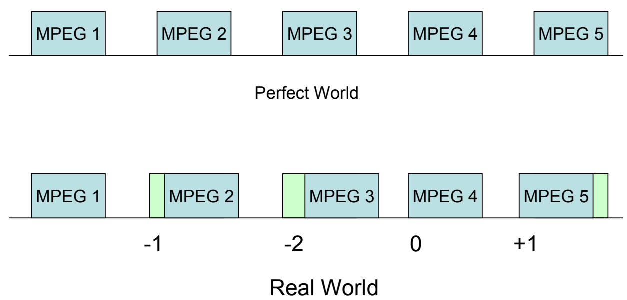

Figure 1 – Jitter Timeline

For the purposes of this paper, jitter is the time difference between when a MPEG-2 packet should arrive and when it actually does arrive. A packet might arrive early or late. In the Real World Timeline in the figure above, packets 2 and 3 arrive late and packet 5 arrives early. Packets arriving early could overflow the MPEG decoder buffer, resulting in lost information. Packets arriving late could result in MPEG decoder buffer underflow, resulting in a packet that was not available when it was scheduled for presentation.

If you observe these arrival time variations over a long period, you can characterize the jitter. The greatest positive arrival time jitter minus the greatest negative arrival time jitter is called the "peak to peak jitter amplitude".

It also may be of interest to see if the distribution of the jitter is completely random or correlated with other events. Correlated jitter might be due to a protocol that packs multiple MPEG-2 packets into a single transport unit. Random jitter might be due to random noise that causes packet loss and therefore packet retransmission. While the type of jitter does not matter to the MPEG-2 Transport Stream specification, knowing the type helps us select the best way to mitigate jitter.

MPEG-2 JITTER REQUIREMENTS

The MPEG-2 PCR (Program Clock Reference) jitter tolerance is ± 500 nanoseconds.[2] The MPEG-2 standard does not define a jitter budget on a per-device basis. The standard instead places an absolute limit on packet jitter as received by the MPEG-2 decoder. Unfortunately, some device manufacturers see this number for the first time and conclude that they may add up to ± 500 nanoseconds of jitter. However, the MPEG-2 PCR jitter tolerance is a system level specification, not a device level specification.

By the time a MPEG stream reaches your home, the devices along the way may have "spent" all but 50 nanoseconds of the MPEG-2 jitter budget. And why not? Every device along the distribution chain is trying to perform its function with the greatest efficiency and at the lowest cost. The lowest cost often results in the maximum allowable jitter. So, as a practical matter, any in-home redistribution system is limited to adding no more than ± 50 nanoseconds of jitter.

An Alternate Jitter Requirement?

Some people read the MPEG-2 Annex D and get excited because it says "4 milliseconds is intended to be the maximum amount of jitter in a well behaved system." [3] What happened to the ± 500 nanoseconds? The jitter they are describing in this Annex is something completely different from the PCR jitter we have been discussing. This jitter is called multiplex jitter. Multiplex jitter is often caused when packets are moved around to create or modify a MPEG-2 MPTS (Multi-Program Transport Stream). After you reposition a MPEG-2 packet, you are still required to re-timestamp the PCRs in the packets to within the original PCR jitter tolerance of ± 500 nanosecond.

Jitter Allocation

Most currently deployed VOD (Video on Demand) systems that use MPEG-2 Transport Streams allocate:

- a small portion of the jitter budget to the MPEG encoder (no device is perfect!)
- a portion to VOD servers
- a slightly larger portion to the output multiplexing and QAM (Quadrature Amplitude Modulation) processing
- a bigger portion to possible in-line remultiplexers/re-encoders that may be present (there could be 2-3 of these)
- a negligible amount to the HFC distribution system

Notice that there are quite a few devices in the path which are "spending" from a very small jitter budget.

NOT ALL MPEG-2 PACKETS ARE EQUAL

Previously, it was mentioned that packets arriving:

- early could overflow the MPEG-2 decoder buffer
- late could underflow the MPEG-2 decoder buffer

This is true for all MPEG packets. Because there is some buffering at the MPEG-2 decoder, underflow and overflow are infrequent occurrences unless the jitter is quite severe. However, there is another effect that is more likely to cause problems replaying a jittered MPEG-2 transport stream – but only for certain MPEG-2 packets.

MPEG-2 requires a PCR bearing packet at least once every 100 milliseconds. This means that most packets will not have a PCR. Excess jitter between PCR bearing packets will typically disrupt the MPEG-2 decoder's timing recovery process. The maximum tolerable PCR jitter is a function of the decoder's PLL (Phase Lock Loop) design. The PLL design affects acquisition time. So, there is a tradeoff that every vendor must make between acquisition time and jitter tolerance. Different vendors have made different choices.

You may send a MPEG-2 Transport Stream with more than ± 500 nanoseconds of PCR bearing packet jitter to a settop and it may play without errors. Some decoders will operate properly when fed a MPEG-2 Transport Stream that does not meet all of the MPEG-2 specifications. However, you must remember that ± 500 nanoseconds is the MPEG-2 requirement. This is all any MPEG-2 decoder is required to support and this needs to be your design requirement.

DATA NETWORKING JITTER

A data network is normally shared among many devices and supports different types of traffic. There is no concept of a continuous stream of bits. Bits are packed into groups which are then sent in discrete bursts. The protocol used by the data network determines when these bursts can be sent.

No widely deployed data networking technology adds less than 50 nanoseconds of jitter when carrying a MPEG-2 Transport Stream. Let me back up this rather sweeping assertion by an example.

Let's look at the time required to send MPEG-2 packets over a common 100BaseT Ethernet. Most systems try to minimize IP (Internet Protocol) overhead by packing seven MPEG-2 packets into one IP frame. This is the maximum number of MPEG-2 packets that will fit into a single IP frame. To send one of these MPEG-2 bearing Ethernet frames requires:

- 12 Bytes for the InterFrame Gap
- 8 Bytes for the MAC Preamble
- 20 Bytes for the IP header
- 8 Bytes for the UDP header
- 1316 Bytes of data (7 MPEG-2 Packets)
- 4 bytes for the CRC (Check Sequence)

Total = 1368 Bytes

At 100 Mbps, each bit takes 10 nanoseconds. Therefore:

- 1368 bytes * 8 bits/byte = 10,944 bits
- 10,944 bits * 10 ns/bit = 109,440 ns

As you can see, the jitter caused by packing these MPEG packets together is far more than ± 50 nanoseconds. Fifty nanoseconds represents only 5 bit times!

Most other widely deployed Ethernet based systems, such as IEEE 802.11a/b/g and 10BaseT are slower and some of these have even more overhead. Step up to Gigabit Ethernet and the numbers get 10 times better than 100BaseT, but still far from ± 50 nanoseconds. Even at Gigabit Ethernet speeds, one bit takes one nanosecond, so 50 nanoseconds represents 50 bit times. The InterFrame Gap required by the Ethernet protocol between every transmission is 96 bits, which equals 96 nanoseconds!

In addition, many data networks provide reliable service by continuing to retransmit a packet until it is successfully received. This is the correct approach to use for delivering data but makes things worse for MPEG-2 Transport Streams. If video data is not delivered by its presentation time, it is better to skip this packet and move on to the next. But if a packet contains a PCR, discarding the packet may mean there is no PCR within the 100 milliseconds required by the MPEG-2 specification. ATM (Asynchronous Transfer Mode) has an encapsulation protocol with a concept of "PCR Aware", but I have not seen this proposed for any other system.

QUALITY OF SERVICE

IEEE 802.11e is currently being developed as a standard QoS (Quality of Service) mechanism for wireless systems. It promises to provide a QoS which meets entertainment video requirements. IEEE 802.11p provides a prioritized QoS for wired LANs and HomePNA 2.0 includes a prioritized QoS, but these schemes do not support entertainment level QoS. While HomePlug 2.0 does not support entertainment level QoS, HomePlug AV (the next version of HomePlug) will support entertainment level QoS.

So, will 802.11e or any of the other QoS initiatives meet the MPEG-2 Transport Stream requirements? No. In all fairness, how could they? As shown in the previous example, the physical transport network just does not have the timing resolution required to meet a ± 50 nanosecond jitter specification.

But I See it Work All the Time!

When a protocol claims to support entertainment quality video, they usually are referring to "Streaming Media". Streaming Media is a different kind of MPEG-2 delivery system. The big difference is in Streaming Media, the decoder works from a "pull" model. In a pull model, the decoder requests blocks of MPEG-2 packets as soon as it has the buffer space to store them. Streaming Media decoders usually have a much larger buffer, so jitter is less of a problem. In addition, they also run their display clock from a local source so that they don't have to worry about timing recovery.

In contrast, a MPEG-2 broadcast decoder is at the end of a fire hose of bits that keep coming at it, whether it is ready or not! A broadcast receiver may need to either drop a frame when its buffer overruns or repeat a frame when its buffer underruns. This is how the broadcast model can use one source to feed millions of decoders. In the Streaming Media model, every decoder is making its own requests. So, if you replaced a broadcast system that served millions with a Streaming Media system, it would require a video server farm that supported millions of sessions.

RECOVERING FROM JITTER

Archimedes said that he could lift the earth if you gave him a firm enough fulcrum and a long enough lever. Similarly, given a large enough buffer and a large enough delay allowance, you can remove any amount of jitter. However, if you have a large jitter buffer, you must wait for the buffer to fill before you can view the video. This requires a patient viewer. Infinitely long levers and infinitely patient viewers are equally unlikely.

Most current PC Streaming Media players use this buffering approach. They normally buffer 5-10 seconds of video prior to playing the first frame. However, entertainment video is often interactive, so “solving” the jitter problem with a large buffer at the receiver will result in a system that seems sluggish. Channel changes would become quite lengthy. Again, this sluggish response is the typical experience we have come to accept with streaming media on a PC.

Even with a large buffer, video glitches are common today in the streaming environment. The proposed QoS initiatives will take care of these problems for Streaming Media, but won’t help our case of broadcast style MPEG-2 Transport Streams.

To use a standard data network to deliver MPEG-2 Transport Streams, you need to use whatever network features are available (such as QoS) to minimize the jitter added by the network. Then you will need to add de-jittering equipment at the edge of the data network.

Constant and Variable Bit Rates

There are two kinds of MPEG-2 Transport streams. Constant Bit Rate streams are, as their name suggests, a steady stream of MPEG-2 packets delivered at an steady rate. Variable Bit Rate streams deliver just enough MPEG-2 packets to present the content at the specified quality – and no more. So, while a Variable Bit Rate stream may be specified as having an “average” bit rate of 3 Mbps, there may be intervals of time where the short term bit rate peaks at 9 Mbps – or more! The MPEG-2 specification allows standard definition rates as high as 15 Mbps.

Most MPTS (Multi-Program Transport Streams) delivered from a satellite feed are a constant 27 Mbps, but the individual program streams that it contains are Variable Bit Rate. If you separate out a single program stream from a MPTS and then send it over a data network, you must be prepared to serve the stream at its peak rate, not its average rate.

What is the difference between Constant Bit Rate and Variable Bit Rate streams when it comes to de-jittering? With a Constant Bit Rate stream, you can buffer the incoming packets and then play them back at the same rate as the rate at which they were encoded. With Variable Bit Rate, you never quite know when to send them to the MPEG decoder. In this case, you need to time stamp every MPEG-2 packet when it arrives in the network so you will know when to play it out. This adds quite a bit of complexity to the protocol plus the added timestamp adds transmission overhead to every packet processed.

PROPRIETARY SOLUTIONS

Two proposed architectures, one wired and the other wireless, claim to have solved the MPEG-2 Transport Stream jitter problem. Air5™ is the wireless solution and MoCA (Multimedia over Coax Alliance) [4] is the wired solution. Both these solutions are chip sets that accept and deliver MPEG-2 Transport Streams. How do they do it? Both use a network protocol that guarantees there will be no collisions plus they have some de-jittering built into the receiving chip.

Unfortunately, as of the time of this writing the Air5™ developer, Magis Networks, has suspended operations and its future is uncertain. However, MoCA is going strong and MoCA products are expected to reach the market by the end of 2004.

CONCLUSION

Traditional broadcast MPEG-2 Transport Streams are not easily carried on a traditional data network. The broadcast MPEG-2 Transport Stream model is based on an unstoppable stream of bits directed at potentially millions of decoders. Strict timing rules are required to make this broadcast system work. If you follow the rules, the system works beautifully.

Streaming Media MPEG-2 systems were designed to be carried on traditional data networks. Therefore they work just fine on current data networks and the coming QoS initiatives will allow them to work beautifully. However, these systems put quite a load on the MPEG-2 source and the transport system because each decoder has a private session with the ongoing video source.

If you must carry broadcast MPEG-2 Transport Streams on a data network, use the best QoS that you can get. Then add a de-jittering device either at the edge of your data network or as a part of your MPEG-2 decoder.

[1] American National Standard Dictionary of Information Technology (ANSDIT), available at http://www.ncits.org/tc_home/k5htm/Ansdit.htm

[2] ISO/IEC 13818-1 Generic Coding of Moving Pictures and Associated Audio: Systems, 13 Nov 1994, section 2.4.2.2.

[3] ISO/IEC 13818-1 Generic Coding of Moving Pictures and Associated Audio: Systems, 13 Nov 1994, Annex D.0.4 SCR and PCR Jitter.

[4] More information on MoCA is available at www.mocalliance.org