

Radio World

ENGINEERING EXTRA

REPRINTED FROM AUGUST 23, 2006

WWW.RWONLINE.COM

WHITE PAPER

by Greg Massey and Jon McClintock

Packet Size Matters in IP Transport

A Look at Sending Audio Over IP Via Packets, and Network Considerations Such as Latency and Jitter

Greg Massey is the chief technology officer for APT; Jon McClintock is the commercial director.

Transporting broadcast quality audio over IP is the new “must have” technology for radio and TV networks. However, IP as a transport mechanism has a number of inherent characteristics that could potentially pose problems for codec manufacturers and broadcasters alike.

While massive IP bandwidth is now available, it comes with some caveats. These include:

- The packets do not always arrive in the order in which they were sent.
- The transport protocols introduce a natural delay to the link.
- Dropped packets are always a possibility.
- Bandwidth is not guaranteed.
- Packet delays through the network are variable.

Manufacturers already have put in place methods and protocols to cope with the plethora of issues surrounding audio over IP, including forward error correction, concealment and security mechanisms.

Experience also has shown that the use of MPEG derivatives in contribution and distribution networks has resulted in almost unusable coding delays and serious degradation of audio quality. The layering together of the inherent delays associated with MPEG and IP will create an impossible scenario for any

broadcaster wishing to put live content on-air.

This paper aims to explore the audio coding technologies available, and explain why the Enhanced apt-X algorithm benefits the broadcast community with a solution that maximizes the benefits of IP, retains acoustic integrity and ensures the coding delay is reduced to a workable value, i.e. under 20 milliseconds.

IP NETWORK CONSIDERATIONS

Latency

Networks have transport latency due to the natural laws of physics. Transporting an electronic signal through whatever medium takes a finite amount of time that cannot be removed.

In a switched network we must contend with both the standard transmission delay and also the packetizing delay. By definition, a packet must be assembled and consists of a

header plus payload. The size of that payload can be varied but ultimately it consists of an audio sample.

Take for example a system that uses a four-to-one compression algorithm and has a packet size of 128 bytes. That's 512 audio samples,

equal to 666 microseconds in mono and 333 microseconds in stereo. Then take the time it takes to propagate through the UDP stack after being assembled into a Real-time Transport Protocol packet. In real, live-unit tests this is approximately 20–30 milliseconds.

It is important to realize that this natural latency increases as the sample frequency decreases. With this inherent latency in the protocol stack the additional delay of the audio coding is critical in real-time audio applications.

Lost packets

Lost packets are a fundamental feature of switched networks and something that all audio codec designers must learn to live with or suffer the consequences. Losing an audio packet quite literally translates into losing audio, which is a major problem in the broadcast environment.

Several solutions exist, ranging from ignoring the problem to masking it through to retransmission. If one discounts ignoring

Lost packets are a fundamental feature of switched networks and something that audio codec designers must learn to live with or suffer the consequences.

the problem as a solution, what remains are the masking and retransmission scenarios.

Packet size

As with all packet-based systems there is a tradeoff between the size of the packet, the

processing overhead and network traffic/congestion.

Smaller packets increase overhead with the additional byte required to populate the packet header. Smaller packets are transported through the network at the same speed but an entire packet can be received more quickly due to its size. The difficulty with smaller packets is the increased likelihood that a packet will be lost or received out of sequence.

Larger packets have the opposite problem once they exceed the Ethernet limit of around 1500 bytes. The packet is then fragmented and transmitted in two or more parts, before being re-assembled at the opposite end. This complicates the receiver with no real benefit.

UDP vs. TCP

Transfer control protocol, or TCP, is a command and response architecture that guarantees delivery through an exchange of information between sender and receiver. This interaction requires time and has proven unworkable for streaming applications requiring low latency or real-time operation.

User datagram protocol (UDP) is simple send/receive architecture with little in the way of payload protection or guaranteed delivery. It does, however, lend itself to streaming applications since there is less processing delay on the protocol.

Jitter

These are packets that are received either side of the predicted arrival time. This is a feature of packet switched networks given that any packet can take any route from source to destination. Thus packet jitter will affect the way in which the receiver must handle the data sent.

Packet arrival jitter can be significant and can amount to several seconds depending on the network. The buffer depth is therefore critical in allowing the codec to provide enough time for the packet to be received and decoded before its playout time. Reducing the buffer size reduces the jitter time mask available and increases the potential of being forced to drop packets that have arrived beyond their playout time.

Fig. 1 shows the effect of network jitter on the reception of audio and its subsequent playout through an audio system. The buffer depth is usually set in milliseconds; in this case it is set to a two-packet buffer. For the purpose of this example, this allows for up to a two-packet jitter delay in the system.

Provided the network jitter is low, the system is unaffected and plays out the packets received in sequence. However, should the jitter increase, there is the distinct possibility

errors in the final reproduction of the audio due to packet loss. They range from simple repetition of the last good packet received, to silence/noise injection, or interpolation and

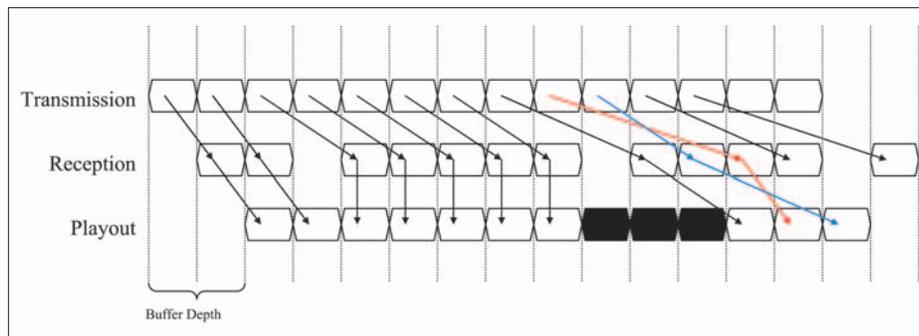


Fig. 1: Network Jitter Effects

that the packets will arrive after the determined playout time. In this example the packets have to be dropped, which results in the audio being corrupted.

CORRECTION/CONCEALMENT Overview

No Forward Error Correction is currently used in the UDP packet, which makes it susceptible to bit errors. It does, however, speed up the processing, as this additional correction stage is bypassed.

Forward error correction also has implications for end-to-end systems. For example, to what extent should the FEC correct the errors, i.e. all or just partially? Alternatively, does a failure indicate that a retransmission should occur?

Concealment

Assuming retransmission is not practical, then the decoder must implement some type of concealment to prevent or conceal audio loss (see Fig. 2).

Various methods can be used to conceal

retransmission. All have an impact on the reproduced audio.

In listening tests the injection of silence produced unacceptable breaks in the audio that led to a level of incoherence. The use of white noise improved the intelligibility of the reproduced audio but was again noticeable. The use of repetition of the last known good frame produced favorable results. The use of interpolation/pattern matching/waveform substitution to conceal the loss of packets is possible but the benefits vs. complexity are governed by a law of diminishing returns.

The results of these techniques are all governed by subjective improvements in audio quality and are also subject to the amount of audio lost that is being concealed or repaired.

Correction

The use of Forward Error Correction to ensure packet recovery can be effective in audio streaming applications but it has implications for real-time applications due

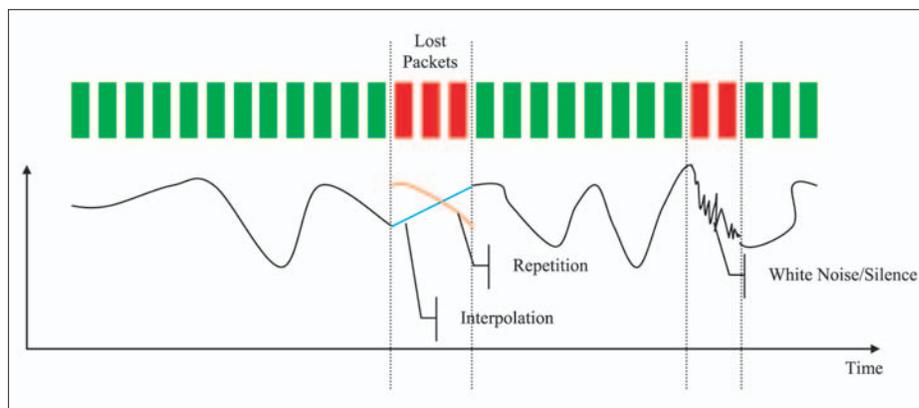


Fig. 2: Packet Loss Concealment

to the processing and data overheads associated with FEC algorithms (see Fig. 3).

The determination of how much FEC to use has to be related to the losses experienced

is that each sender and receiver has a different master clock; if they are sending at their basic rate they will eventually over-run or under-run the buffer.

This makes systems using retransmission unsuited for real-time applications.

In networks using multicast or multiple unicast the retransmission of packets from



Fig. 3: Perceived Audio Quality

with the medium being used, as adding more overhead to a heavily congested medium may exacerbate the situation. Also, the scheme used must be tailored to the type of loss being experienced. For example, is the FEC designed to correct burst packet loss, or a percentage of lost packets if the packets, which are lost, are non-consecutive?

FEC Techniques Comparison

To maintain compatibility and interoperability between codecs the FEC information should be sent via a separate port so that the audio codec does not become confused if it cannot handle the FEC scheme.

Clock skew

Assuming data is received from a central master unit, then each slave receiving the stream must adjust its ployout to match that of the master, otherwise the buffer will either overrun or under run. While this can be

The latency of the compression algorithm then becomes an imperative when considering the design of a broadcast network.

done easily, problems can occur in multicast and multiple unicast (see Fig. 4).

In a simple network setup there is only one master and many slaves. IP makes it possible to send and receive audio feeds to and from anywhere in the network. The difficulty

This is further complicated if multiple streams are being received from different sources. It implies that each stream must be monitored and the play-out adjusted to track the master clock.

Technique	Overhead	Complexity	Scope
FEC per n-1 packets	Low	Low	Uniform Loss No Burst Recovery
FEC packet per packet	Low	High	Full recovery possible
FEC packet per n packets	Med	High	Burst loss recovery possible depending on scheme Increased delay

FEC Techniques Comparison

More basic systems ignore this and simply have a strategy that allows for over-run and under-run to occur before restarting the system.

Retransmission

This has implications for the real-time operation of the unit given the processing delay and use of Transmit and Receive buffers. A retransmission must insert the retransmitted packet into the buffer at the appropriate point (see Fig. 5).

RTCP feedback can be used to signal to the sender that packets should be retransmitted. There are benefits and problems associated with the retransmission of lost packets.

The primary advantage of retransmission is the ability to completely correct lost data. This is very effective for small groups and one-to-one streaming.

Disadvantages of retransmission include an increase in transmission overhead and increased latency. Since packet loss is generally due to high network traffic, retransmission will tend to exacerbate the problem and lead to more losses. Further, retransmission causes an increase in latency as the lost packets are sent, an error message is returned, and then packets are resent for a total of three trips over the network to receive the data.

multiple sources will contribute a substantial overhead to the network traffic and may also swamp a single sender unit if all slaves request retransmission of a single packet.

If retransmission strategies are used in

real-time applications to ensure no audio dropout, then they will have to incorporate enough buffering to compensate for all possible packet-loss scenarios.

Audio Algorithms

Having thoroughly investigated the intricacies of IP as a method of moving program content from Point A to Points B, C and D and through to Point X, the next step is to look at the best method of layering in audio on top of the transport stream.

In essence there are two options — PCM/linear or using compression to reduce bit rates. Within compressed there are two sub-options, perceptual or ADPCM.

PCM or linear audio is well defined in terms of the audio; what you get in should be what you get out, assuming there are no problems relating to analog-to-digital conversions, signal-to-noise ratios or quantization issues. The compelling reason not to choose linear is directly related to the data bandwidths required.

A stereo signal sampled at 44.1 kHz, with a word depth of 16-bit, will require a data rate of 1.411 Mbps (plus 10–15 percent overhead and additional for FEC and synchronization algorithms). This data rate band-

