



The

Broadcasters' Desktop Resource

www.theBDR.net

... edited by Barry Mishkind – the Eclectic Engineer

DIGITAL BROADCAST

A Closer Inspection of Audio over IP Links



**By Keyur Parikh, VP and GM, Intraplex GatesAir
with Brian Galante**

[June 2018] As more and more audio is moving via digital paths, it is important to know how to ensure the audio is clean. GatesAir's Keyur Parikh looks at the conditions and solutions that impact signal integrity and network health for the broadcaster.

The proliferation of public IP networks for audio contribution and distribution has provided broadcasters with several benefits, including operational cost reduction, broader connection flexibility, and greater interoperability.

Yet, while these benefits exist, the packet loss impairment that happens within the IP network can greatly reduce the quality of the broadcast signal. Such packet loss impairment is uniquely a symptom of IP networks, and for long has been among the main concern for broadcasters considering a transition from circuit-based systems.

CODEC CONSIDERATIONS

Unlike TDM networks, where the networks did the heavy lifting, end devices on IP networks – notably, audio codecs - must be equipped with capabilities to overcome network impairments.

Real Time Protocol (RTP) is the most widely used and specified protocol for real-time media transport over IP, whether audio, video, or voice. RTP operates over UDP, a computer networking protocol that is a core member of the Internet Protocol suite. As such, UDP transport is preferred for real-time media, as it allows users to have a better control on playout delay due to the lack of variability of lost packet retransmission.

Also, UDP can operate over multicast and unidirectional connections, which provides added flexibility for how broadcasters manage their contribution and distribution in balance with overall network traffic and bandwidth usage.

FORWARD ERROR CORRECTION

Without retransmission, RTP over UDP transport relies on one of two methods for packet loss protection.

One of those methods is packet-level Forward Error Correction (FEC). In this method, parity packets are created at the encoder side and sent across the network along with the original data packets.

The decoder uses the FEC packets to recover lost packets.

The effectiveness of FEC to recover lost packets depends highly on the pattern of the packet losses and the size of the FEC matrix. For real-time implementation of FEC, the matrix sizes are typically limited to 16 or 24 packets, which limits its effectiveness in recovering packets with burst losses.

For random or isolated loss pattern, FEC is very effective and bandwidth-efficient. Knowing whether FEC will work on specific type of network losses is where real-time packet loss pattern analysis can be a great help to the broadcaster.

DUPLICATE PACKET TRANSMISSION

The other method for packet loss protection uses duplicate data packets transmission from the encoder.

Duplicate packets can be sent with both network or time diversity. Network diversity is the most popular method, as packets are sent over independent ISP paths. Time diversity is where the duplicate packets are sent over the same network connection, but with specific time-offset.

Because network connections can be unpredictable, having all three methods work in combination can provide very effective protection, even on the toughest public IP connection. This is where the combination of network and time diversity RTP streams provide the front-line defense against heavy burst losses, and FEC provides second line of defense to recover any residual losses.

The advantage of this method is that it provides protection against complete network loss, while also enabling a scalable solution for protection - even when operating on a single network connection. Additionally, these methods are agnostic of the payload carried by the RTP/UDP

streams. In other words, it can work for any signals, including discrete audio, FM MPX, GPIO, and even Ethernet packets.

THE PRESENT AND THE FUTURE

One of the important aspects of using public IP networks for broadcast audio is to understand the performance of the network connection, both in real-time and over time. This is where monitoring software with a deep analytical tool-set can provide a significant advantage.

Intraplex *LiveLook* is one such available application.



Observing audio with *LiveLook*

Specifically, *LiveLook* is an off-board network analytics and monitoring software for *IP Link* codecs.

It provides a graphical analysis of both real-time and historical network performance data as well as the monitoring of audio streams' state changes and optional email notification capability.

Using *LiveLook*, users can monitor and analyze the audio stream's performance. The built-in analytics engine can generate email alerts as well as reports to help guide the users on how best to configure the packet loss protection on their codecs.



The IP Link screen showing signal paths

The logging facility also enables users to monitor the Service Level Agreement (SLA) of their network connection. In short, the combination of *LiveLook* and *IP Link* audio

codec work together to get the most out of the public IP connection.

Additionally, there is report generation capability which analyzes the selected network performance data and recommends the best error protection technique of the *IP Link* codec to use.

For broadcasters transitioning to IP, having a single management application to analyze and monitor all the audio streams of the network with help them maintain the best possible operating conditions for audio and data transport over IP networks.

This will be especially important for networks and stations as they introduce new digital services to their contribution and distribution links moving forward.

Keyur Parikh is the Vice President and General Manager, Intraplex for GatesAir.

He can be contacted at:
Keyur.Parikh@GatesAir.com

If you would like to see more articles like this, we invite you to sign up for the one-time-a-week BDR Newsletter. [Just click here, it only takes 30 seconds.](#)

[Return to The BDR Menu](#)