# AUDIO RELIABILITY OVER THE PUBLIC INTERNET:

## DESIGNING ROBUST IP-STREAMING FOR OUTSIDE BROADCASTS, STL, AND PROGRAM DISTRIBUTION.

### ABSTRACT

One of the challenges in transferring audio programming over IP networks, especially over the Public Internet, is data loss caused by two factors: congestion-related packet loss and varying latency or jitter. IP links may drop packets for several reasons; though some transmission protocols are designed to mitigate or correct such losses, they require extra bandwidth and extra time to make these corrections. In this paper we examine several data transmission error mitigation techniques in the context of their application to real-time, low-latency IP-audio transport. We suggest how different techniques may be applied to different data loss or jitter scenarios. And, we assist the reader in analyzing his own data transmission path, characterizing any difficulties, then selecting the best technical solution to mitigate or eliminate any final effects on the delivered audio product.

### INTRODUCTION

Many radio engineers are wary, even skittish, about relying on an Ethernet/IP network as the conduit for critical program audio. Perhaps too many unresponsive "Print" commands, unresponsive web pages, IP networking confusion, and wonky Internet connections have led to this skepticism about real-time, low-latency audio transmission over IP.

In recent years, however, we've come to accept and even embrace localized Audio over IP (AoIP) within and among local studios. Here we have end-to-end control of the network environment. Local Audio over IP (AoIP) is enabling low-latency, Ethernet/IP-connected audio at about 4,000 radio studios at this very moment. Localized AoIP works very well when all the equipment, wiring, and configuration are under our purview - our direct control.

### CHALLENGES OUTSIDE THE LOCAL STUDIO

What typically happens when we encode audio data into IP packets, and then hand those packets over to a third party for transport? What's the impact to IP-Audio when Real Time Protocol (RTP) packets must compete with a rush of other packet traffic, either outbound or inbound? And, how can we optimize an AoIP system for end-to-end low-latency, while assuring no perceivable audio dropouts?

For a couple decades or more, radio engineers have used ISDN, T1 and E1 circuits, satellite links, and various RF point-to-point solutions to transport audio to and from the studio. While not without their own issues, at least these telco-provided services are tariffed and come with a basic guarantee of performance. And, the RF links tend to be under the engineer's direct control with no ambiguities.

More recently, economics and availability are often dictating that we use third-party IP links in one form or another to transport real-time audio. There's a perception that such links are inherently less reliable than the technologies they replace. And while that may seem true anecdotally, it doesn't have to be.

Private or dedicated IP transport connections can be every bit as reliable as older technologies. They're usually less costly, available from multiple, competitive providers in the same area, and are far more flexible in terms of capability, utility, and pricing. Moreover, private or dedicated IP links usually offer one or more forms of Quality of Service (QoS) as well as optional Service Level Agreements (SLAs), placing reliability as high as other, older technologies, and offering the flexibility to mix data types that older tech couldn't do. Private or dedicated IP connections such as these rarely require extraordinary technology from the endpoint equipment - in this case the IP-Audio codecs. Over an essentially perfect and known IP connection, even the most basic and static IP codecs can work reliably.

This paper concerns the use of IP codecs over imperfect links. Such links usually include the Public Internet, but also encompass congested and variable wireless links like 3G and 4G, WiFi, and WiMax services. We must also include the data impairments that occur within a Local Area Network (LAN) as competing packets are routed to and from the local Internet Service Provider (ISP) and on to the Public Internet.

IP-Audio is reliable and robust in a controlled network environment. Latency can be very low with easy routing of channels and superlative operational conveniences. Our challenge - and opportunity - comes with getting excellent audio performance across highly-trafficked, imperfect links that we don't control. And that is where clever technology steps in.

Regular (non-real-time) IP traffic that must be one hundred percent reliable - bit for bit reconstructed at the receiving end - is commonly transported using TCP/IP. This protocol can assure that, eventually, there's a perfect transfer of data across any usable network, as long as it doesn't matter how much time the process takes. Whether a file downloads in five seconds or five minutes is of less consequence than making sure the file is one hundred percent complete and bit-for-bit identical to the source. TCP/IP will slave away, requesting and re-requesting a complete and error-free transmission, packet-by-packet, from the far-end source. TCP/IP doesn't give up until every bit, byte, and packet is transferred, no matter how long it takes, notwithstanding wholesale connection timeouts. If you've ever downloaded a huge file over a slow connection, you've experienced the relentless robustness that hallmarks TCP/IP.

"No matter how long it takes", then, is precisely why TCP/IP is not appropriate for real-time audio or video. Indeed, transferring real-time media and its metadata over a lossy, jittery, packet-switched network appears counter-intuitive on its face.

One-way media distribution, such as music streaming or video entertainment over IP can use TCP/IP, however. Thanks to a large receive buffer, playout applications can request and buffer upwards of 30 to 60 seconds worth of streaming data, then back off further requests until needed to keep the buffer full. The application meters out the buffered data in real-time, but only locally to the user.

For the balance of this paper, we'll refer only to audio over IP and it metadata, but similar concerns and solutions apply to real-time video streaming as well.

Meaningful audio performance for most radio broadcast uses implies two-way audio and low delay while maintaining the highest possible audio quality. When a two-way, low-delay audio connection is desired, there isn't time for TCP/IP's handshaking and retransmission. Neither is there the security of large data buffers to even out the flow of data at our applications' user interfaces.

The protocols of choice for most real-time audio streaming scenarios are User Datagram Protocol (UDP) and Real Time Protocol (RTP). UDP is free of time-consuming handshakes. It's a one-way stream of data, sent at the request of the receiving end. RTP offers synchronizing, timing, and prioritizing information, useful to keep the receiver's playout in sync with the encoder.

A typical IP-codec transport path will present two major delay components and several minor ones. Cumulatively, these delay components comprise the total one-way audio delay. Those major contributors to delay are the summed encoding/decoding, or "codec" delay, the packet's network transit time, and the appropriate receive jitter buffer delay.

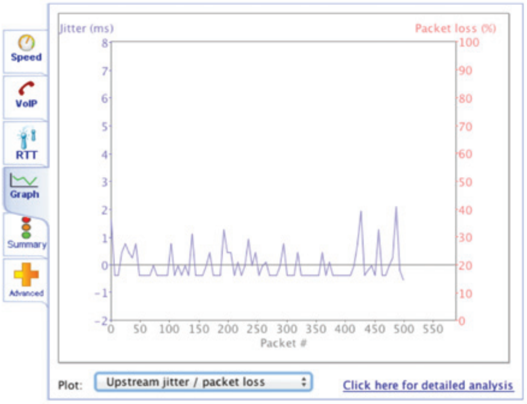
Minor delay contributors include the codecs' A/D/D/A conversions or Sample Rate Converters (SRC's), along with the audio handling delay of the underlying operating system (OS), and packetizing/depacketizing delay.

We'll see how different approaches toward our goal may be used individually, or perhaps together, to mitigate the data impairments presented in typical IP networks.

One only need glance at a histogram of dropped or delayed packets in an IP-Audio stream to identify the challenges that a given IP connection will present.

Following are two histograms showing test results from different connections to the Public Internet. The test shows upstream packet jitter between the testing computer and a server.

Figure 1 shows a test via a hard-wired Internet connection in a low-congestion environment. Packet jitter is 2 ms or less. This is quite good and likely to present a very usable path for high-quality IP-audio carriage, assuming similar performance over a longer time horizon.

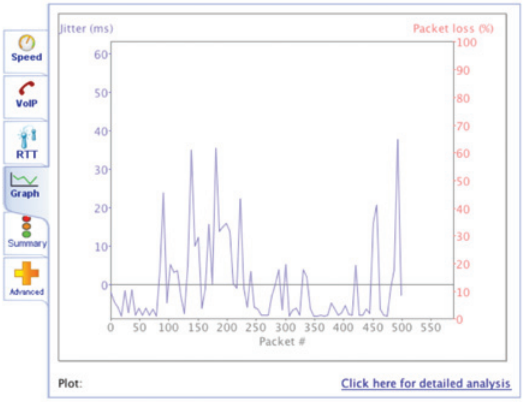


Plot: Upstream jitter / packet loss

Figure 1

Figure 2 was conducted with the same PC and the same software, but the Internet connection is via a wireless WiMax service. The packet jitter reaches nearly 40 ms. This level of jitter is still usable, but will require additional buffering in the receiving IP-audio codec. 40 ms typically represents two packets of compressed audio data. Using a connection with 40 ms of jitter, we would be wise to set at least 80 to 100 ms of additional buffer time.

If a given IP path is exhibiting large amounts of or large variations in jitter, we should endeavor to find out the cause. High jitter figures often indicate an overloaded link or a misconfigured router. Jitter may also indicated simply a poor physical connection somewhere and the TCP/IP protocol is tirelessly working to deal with it, resending packets until they successfully reach their destination.



Plot:

Figure 2

The tests shown above are parts of a comprehensive suite of free tests intended to troubleshoot Voice over IP (VoIP) difficulties, or to pre-qualify an IP connection for VoIP service. The web URL is: http://www.whichvoip.com/voip/speed_test/ppspeed.html. VoIP service and high-quality IP-audio are rather similar, so these tests are useful to broadcast engineers, at least for testing from a local Internet connection to the test servers.

Next we'll examine different approaches to making IP-Audio more reliable. Some work over less-than-perfect IP connections, while others may use two separate connections to be assured of a good connection by dint of aggregation.

## APPROACHES TO RELIABLE IP AUDIO

### Use Perfect IP Networks

Employing a more-or-less perfect or wholly transparent IP Network is certainly simplest from a codec design perspective. Over a perfect IP network, codecs don't have to be smart in any way; they simply need to be compatible. A perfect IP network will transport IP-audio packets from the encoder to the decoder with no packet loss, no packet mis-ordering, very low latency, and sub-packet-interval jitter performance.

Well-managed private Wide Area Networks (WANs) can deliver this network experience, especially with the assistance of end-to-end Quality of Service (QoS) implementation. QoS is often achieved though Multi Packet Label Switching (MPLS) and assures that the desired packets are handled first at each network device. Perfect or near-perfect IP networks may be an option for program contribution/distribution networks within the same corporate infrastructure, as well as critical point-to-point audio transmissions.

### Generate Fully Redundant IP Streams

If one IP-audio stream is mostly reliable, then adding another in parallel for redundancy should prove exemplary. It's possible to design a codec to send identical IP-audio streams through two different physical Ethernet interfaces. Ideally, these distinct interfaces are networked through disparate equipment connected to separate WAN or Internet providers. At the far end, the same disparate and redundant connection scheme completes the provisioning of two thoroughly redundant IP transport paths. Due care is taken to ensure that two different local Internet providers are not themselves connecting to the same Internet backbone. After all, partial redundancy is not true redundancy.

A benefit of enabling, within a single bi-directional codec, use of redundant IP transport paths is this: The codec's decoder can seamlessly switch between incoming packet sources. Shown in Figure 3, if a packet - or two or ten - are missing from Stream and Path "A", then replacement packets should be timely available from Stream and Path "B". We configure the decoder's receive buffer to accommodate the worst-case latency offset plus jitter from the more latent of the two streams. Then the decoder can be configured to simply replace missing, late, or defective packets with good ones from the redundant stream, affording no interruption in the decoded audio.

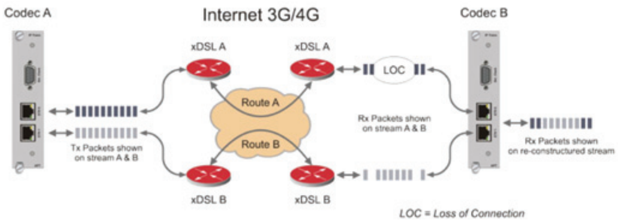

Codec A
Internet 3G/4G
Codec B
xDSL A
xDSL A
LOC
Route A
Route B
Rx Packets shown on stream A & B
Tx Packets shown on stream A & B
Rx Packets shown on re-constructured stream
xDSL B
xDSL B
LOC = Loss of Connection

Figure 3

However, even with redundant data paths this approach isn't truly redundant unless there are two completely disparate codecs at each end. This gives rise to considering how much redundancy is enough.

An alternative to purpose-built redundant-interface codecs is to use one single-interface codec at each end, then using Virtual Private Network (VPN) technology, create IP tunnels over separate IP transport providers. The work of creating and then switching between two data paths is left to the IP routers, which can be inexpensive. Note that this router-based data path redundancy offers switching of the data transport path, but not packet-by-packet replacement.

The redundant IP stream approach to reliability can also offer some degree of benefit even when implemented over a single connection. Assuming adequate end-to-end bandwidth, sending the same data twice over a data path characterized by only occasional, single-packet dropouts can benefit from having a replacement stream readily available, especially if we can also include some temporal diversity.

The advantages of sending and receiving redundant IP-audio streams fully depends upon having the requisite bandwidth to do this. One requires either disparate data paths or a single data path that's twice as big as would be needed with single-stream transmission. When using wireless connections, or a smaller DSL connection, or any connection shared with other users, there's the likelihood of instances when the real-time audio packets will have wait or be dropped by a router. In some situations one must ask, "If the end-to-end system lacks reliability due to occasional bandwidth bottlenecks, then how will sending twice the data , causing more congestion, be helpful?"

### Generate Temporally Diverse Redundant IP Streams

What if the bandwidth is available for redundant streams over a single data path, but we want to obtain more useful redundancy though time diversity? This question gives rise to the approach of sending redundant streams that are separated in time by some appropriate amount, depicted in Figure 4. If one analyzes a given data path and discovers regular or irregular short-term flow interruptions or jitter anomalies, then temporal diversity can provide a solution.

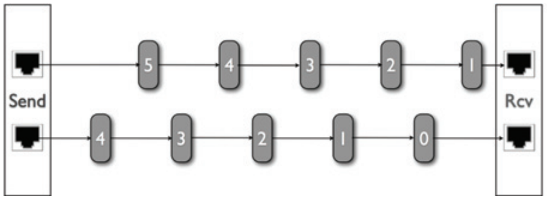


Figure 4

The delay of the 2nd IP-audio stream would necessarily be slightly longer then statistically significant interruptions or periods of increased jitter. The receive buffer would also need to be slightly longer than the stream offset.

Radio engineers familiar with HD Radio transmission know that this data is sent using both frequency and temporal diversity to mitigate multipath and shadowing effects, especially in mobile reception environments.

One codec manufacturer is experimenting with sending packets out of order in the redundant stream, then setting the receive buffer for the maximum random offset of a redundant stream packet.

### Employ IP Forward Error Correction

Forward Error Correction (FEC) as understood by many broadcast engineers, is best applicable to non-packetized serial data. Such has been the case with digital satellite transmission and other serialized data transports. Noise burst or other data loss episodes are expected to be very short-lived in this kind of data path. As such, one can reasonably apply Reed-Solomon Convolutional coding and recover these short-lived data losses appearing at the decoder.

IP-audio is packet-based. Typically, an entire packet will be lost - or several packets in quick succession. Even partial packet loss implies loss of the entire packet. Either way, much data is lost, and recovery using common FEC techniques requires the FEC data be spread over quite a few packets. As such, one must employ a transmission buffer as well as a receive buffer. Overall delay increases dramatically and could be in the hundreds of milliseconds when aggregated. Clearly, employing FEC becomes counter to our goal of low-latency reliable audio transmission.

Some research is ongoing employing Reed-Solomon Erasure codes in both single and double column RSE algorithms. Packet losses of up to twenty percent are showing promising recovery rates with only a twenty percent increase in overall data rate for the FEC data. Still, the costs of such application of clever FEC algorithms are both additional time delay and computational complexity.

Dynamically-Controlled IP Stream Management Theres's a sharp reality about employing an IP data path for real-time data: the jitter and available bandwidth can be quite variable from moment to moment.

Unless some automatic encoder bit-rate control is implemented, the date path must be over-provisioned and, hence, under-utilized. Additionally, the codecs' receive buffers must be configured for the worst-case packet jitter. In other words, both usable bit-rate and buffer delay cannot be better than the worst-case, in order to maintain a reliable audio stream.

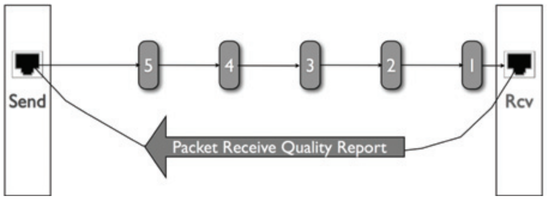


Figure 5

Said another way, a single IP data path can be used optimally only when real-time adjustments to receive buffer size and encode bit rate are managed dynamically by a smart algorithm. This basic feedback path is shown in Figure 5.

Keeping the goal of low-latency audio transport in mind, the encoder/decoder pairs are managed such that buffer size is persistently minimized, consistent with reliable audio decoding. However, if the buffer size reaches some acceptable maximum and some packets are still lost or untimely received, the encoder can reduce the encode data rate in increments until a reliable data rate is found. From time to time, the management algorithm will attempt to increase the encoding rate and decrease the buffer sizes within pre-configured limits.

A key element to dynamic management of the codecs is employing a coding algorithm that conceals occasional data errors or packet losses. The more recent AAC family of audio codecs do exactly this. Indeed, AAC's error concealment allows packet loss of five to ten percent with no obvious degradation to decoded audio quality. We may exploit this sophisticated error concealment as part of a comprehensive algorithm that dynamically seeks to set the receive buffer size just above the minimum required for low packet loss. An algorithm seeking the minimum acceptable receive buffer size can occasionally reduce the buffer to the point where packets are not arriving timely due to packet jitter in the IP data path.

### Selection and Hybrid Approach

Which of the preceding approaches is appropriate for a given operational scenario?

We consider the following factors and requirements:

- Required overall reliability
- Acceptability of short-term or long-term audio dropouts
- Use timeline - permanent or temporary
- Availability of IP connectivity
- Audio quality expectation

Ultimate reliability is best addressed by using two or more IP data paths. An IP codec designed with dual ports and appropriate software can do this, even to the level of packet-by-packet replacement. An external solution, such as a multi-WAN router can perform similarly in terms of reliability, except that switching will be on a path basis and not a packet basis. Such a multi-WAN router may be capable of switching among two or more WAN connections.

Permanent installations are made most reliable by appropriate port-forwarding through routers, allowing direct, peer-to-peer connections between codec pairs. Temporary use cases, such as outside broadcasts are assisted greatly with help from a “rendezvous server”, making connection through unknown routers and firewalls convenient.

The availability of appropriate IP connectivity varies widely. Even where one form of IP connection, such as DSL or cable, is available, obtaining a backup may prove troublesome. Clever engineers are using inexpensive point-to-point IP radios to extend alternate IP connectivity to remote transmitter sites. In some cases, wireless 4G LTE service is serving as a backup path for permanent installations, and as the main (or only) IP connection for outside broadcasts.

Key to selecting the best options in IP codecs and connection schemes is understanding the nature of IP codec operation. Factoring technical requirements with operational goals and connection options will afford high-quality audio and reliable operation is nearly every scenario.

**Kirk Harnack**
Vice President and Executive Director
Telos Systems