

ADVANCED TECH FOR IP REMOTES

STEVE CHURCH

Telos Systems
Cleveland, OH USA

ABSTRACT

ISDN has served broadcasters well. Indeed, it was a small-scale revolution when it first appeared in the early nineties. For the first time, the dial-up network could be used for high-fidelity remotes. Compared to the equalized analog “broadcast loops” that were the only high-fidelity telephone service before, ISDN was a miracle. While ISDN is still a perfectly good technology, it does have some drawbacks. The main one is that usage is billed by the minute. Another is that installation of the line at the remote side usually has a multi-week lead time and has a significant set-up charge. In some parts of the USA and in other countries, ISDN service is being discontinued or has become difficult to get. IP networks are becoming the new way to get broadcast audio to here from there. A broadcast codec taking advantage of new technology and optimized for the real-world conditions on IP networks makes this a practical reality.

IP TO THE RESCUE – BUT...

Years ago, Telcos envisioned ISDN as the way Internet would be delivered to the masses. But DSL now fills that role. Broadcasters remain one of the few users of ISDN basic rate service, but they don’t provide enough business for Telcos to justify the expense of maintaining the infrastructure.

So we turn to the ubiquitous Internet and other IP networks for an alternative. At many remote sites, there is an existing IP connection that can be hijacked for an ad-hoc broadcast. High-speed IP links via mobile phone networks offer the chance to connect from almost anywhere within large cities. International connections are no problem. And the per-minute charge is gone.

But IP has its problems. On the Internet, there are no quality-of-service guarantees. That means that the packets may jitter, may be dropped, and may not provide a consistent bandwidth. To avoid audible problems, equipment for audio transmission over IP networks must be designed to cope with these conditions. While older equipment might work over networks that have good QoS, the latest generation IP codecs have been designed from the ground-up to deliver high-quality audio, even when the networks are not.

The first problem to be solved is the delay variation in packet arrival, known as jitter. On the Internet, this ranges from tens to hundreds of milliseconds. In contrast, ISDN has no significant jitter. A long buffer in the

receiver, one set to the maximum expected jitter time, can be used to even the flow and deliver consistent audio. But this comes with a cost – audio is delayed. Since codecs at remotes are often used in two-way fashion, this delay can cause trouble for the talent, who find it difficult to speak naturally. And there is always the possibility that a packet will arrive later than the maximum buffer time. Thus, a fixed buffer has only limited utility. Much better would be an adaptive buffer that automatically contracts when the jitter is low, and expands when jitter increases. But this means that audio needs to be squeezed and stretched, not something so easily accomplished in real time.

Dropped packets are a normal condition on the Internet. It was designed so that any router node that becomes overloaded can deliberately drop some percentage of the packets. The TCP (Transmission Control Protocol) part of TCP/IP is intended to deal with this. When it senses that a packet has not reached its destination, it requests a re-transmission. It also lowers its rate of flow to adapt dynamically to network conditions. Retransmission recovers lost packets, but again, at a cost. The receiver must have a buffer long enough to cover the time of the detection and re-transmission procedure. Better would be a system that could deal with lost packets by inaudibly concealing them, rather than requiring re-transmission.

Finally, we must cope with the fact that there is no guarantee for bandwidth. If we set the codec bitrate to some high value to get good fidelity, we might find that over time the network can’t provide enough bandwidth to support that rate, causing audio interruptions. So we might then decide to be conservative and set the codec bitrate to some low value to be more confident that there will be no drop-outs. But we now sacrifice audio quality. Wouldn’t it be better to have a codec that automatically detects the bitrate that the network can support and then adjust to that rate? Even better would be if this adjustment could be ongoing and dynamic, adapting to network conditions. Until recently, no codec was able to “gearshift” inaudibly, but a new one has been invented which can.

An ideal IP codec system would have the following characteristics:

- Effective, inaudible packet loss concealment

- Adaptive receive buffer, with the necessary time squeeze/stretch capability

An efficient codec, to achieve maximum audio fidelity from the least bitrate

Adaptive codec bitrate, accommodating dynamically to network conditions

Fortunately, with the latest advances in codec technology, all of these are now possible.

AN IP-OPTIMIZED CODEC SYSTEM

A broadcast codec intended for IP application needs to be optimized for the purpose. An integrated system that pulls together a suite of appropriate pieces will be much more effective than codecs that have not been tuned and optimized specifically for the IP world.

The Codec Core

The MPEG AAC family codecs have always had quite good concealment techniques, but these had been optimized for the bit errors found on non-packet transmission paths. Recent work has expanded the concealment technology so that it can work effectively with packet loss as well. It's a clever technique. The codec keeps an ongoing measure of the spectral shape of the audio. This is easy because the codec already must have a time-to-frequency domain transform as part of its perceptual coding functions. When a packet loss is detected, a synthetic replacement is created by using the spectral values to filter white noise. To the ear, this sounds very much like the original. The amplitude is tuned at each end of the packet to match the preceding and subsequent packets so there is no audible "pop" from the splice. It turns out that this can be very effective, indeed. As much as 20% random packet loss can be inaudibly concealed.

MPEG AAC is an efficient codec, and the Spectral Band Replication (SBR) addition makes it the most efficient within the MPEG family. AAC with SBR is called officially AAC-HE (High Efficiency), but is also known as AAC+. The downside is it has quite long delay – around 150ms. That would mean 300ms for a round-trip, plus yet more for the IP packetization and buffering processes. Too much for interactive two-way conversation. AAC-LD comes to the rescue. It has around 50ms delay, so is much better on that count. But it has 30% less bit-efficiency than vanilla AAC. Since SBR adds approximately the same amount in efficiency, if we could combine that with AAC-LD, we would have a low delay codec with the coding power of plain AAC. And that is just what the new AAC-ELD (Enhanced Low Delay) codec does. It has reasonably good fidelity down to 24kbps and excellent fidelity when used at 64kbps and above. At 128kbps, it is regarded as indistinguishable from the original.

AAC-ELD's wide bitrate range is a good match to the needs of IP networks, since they vary so widely. A mobile phone connection might be limited to perhaps 40kbps, while dedicated links could be sized as desired, supporting codec rates of 256kbps, 384kbps, or more.

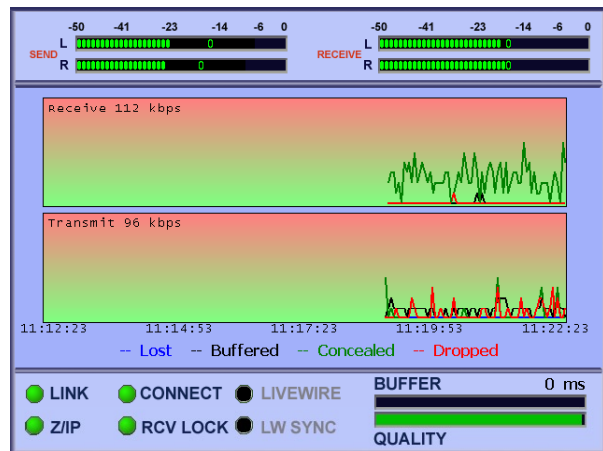
An international Internet connection might support a 64-96kbps rate.

Adaptive Bitrate

An important feature of the AAC-ELD codec is that it can be made to "gearshift" its bitrate without making audible glitches. Coupled with a tuning algorithm, a broadcast codec can automatically adapt to the available network bandwidth. The algorithm constantly probes the network for the maximum rate that can be carried and sets the codec to this rate. When the network has high capacity, audio is as high-fidelity as possible. When the network has limited bandwidth, the codec adjusts to a low bitrate to ensure that audio gets through.

Adaptive Receive Buffer

Unless we have a guaranteed QoS network, it is not possible to predict the jitter. Each packet is subjected to different network load conditions, which affects the transit time. In fact, each packet may take a different route. For uninterrupted audio, a buffer in the receiver must accommodate the longest delay the network presents. If a packet arrives outside of the buffer time, it's as good as lost.



Codec Status Screen Showing Network Performance
Packet Loss, Jitter, and Bandwidth Vary with Time

On the other hand, a long buffer translates to a long delay. So we want to optimize the buffer for the conditions that actually exist. And we want this to vary as needed to adapt to changing network conditions. But how do we detect the network condition? Recall that TCP adjusts its flow rate when packet loss is detected, so this is a long established way for attached equipment to respond to the network. TCP is constantly probing the network for the fastest supported speed by increasing the rate until loss is detected, then backing off. We can borrow exactly this idea for our receive buffer adjustment. We start a new connection with an average-length buffer. If packet loss is detected, we expand it. Unless there is an extreme case, the effect of the lost packets are not heard because the codec conceals them.

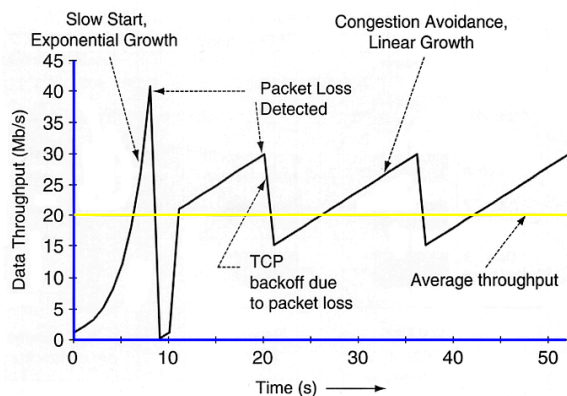
The adaptive algorithm is constantly pushing to reduce the buffer length. To minimize the number of con-

cealments, the algorithm has a fast-attack/slow-release time characteristic – expanding the buffer quickly when a lost packet is detected, but allowing it to contract only slowly. Just as with TCP this feedback loop causes the buffer to automatically adjust to the optimum length. On networks with low-jitter connections, the buffer is automatically made small so as to minimize delay. But when the jitter is high, the buffer is made long to ensure that there are no audio drop-outs.

The buffer adjustment requires that time be squeezed or stretched, and this must be accomplished inaudibly. Fortunately, this is possible – as many audio editors with this feature demonstrate. We broadcasters are well familiar with profanity delay units that also have stretch/squeeze processing. This concept can be used also in the codec system application.

Transport

As we've seen, TCP solves the lost packet problem via retransmission, but this imposes a delay penalty since the buffer has to be long enough to accommodate the time it takes for the replacement packet to arrive. Streaming audio on the Internet usually uses TCP – and there are usually multiple-second-long buffers in the players. That's why it takes so long for audio to start after you click on the link or play button. When we care about delay, this is not going to do. TCP's flow control algorithms are also a potential problem, since they could needlessly restrict bandwidth.



TCP's Flow Control Causes Available Bandwidth to Vary

Forward Error Correction is another way to deal with packet loss. The principle is simple: both the original and some form of copy of the packets are sent on the network. If one is lost, hopefully the copy was not and the receiver can use it as a replacement. The structure of the original-copy sequence is organized to maximize the chances for successful recovery. You don't want to just put the original and a copy adjacent to each other, since that increases the odds that both will be lost. A minimum 2x2 FEC requires the buffering of four packets, while a more reliable 5x5 FEC would require a 25-packet buffer. The latter has more time spread, so is better able to cover losses. But now, unfortunately, we are back to significant

delay – in the 5x5 case, as much as 600ms with the usual packet size. As well, FECs cause streams to take more bandwidth, and a network that is losing packets is one that is probably already near its limit, so adding to the bandwidth requirement is just as likely to create a problem as to solve one. There may be some cases where FECs make sense, but they are generally not useful for audio on the public Internet.

The alternative is User Datagram Protocol (UDP) combined with concealment. Using UDP, we are as close to the underlying network as we can be, so the delay is as low as possible and the bandwidth as high as possible. Responsibility for dealing with packet loss is moved to the "user", which is perfectly OK because we can deal with it in a way specialized for our audio application, rather than accepting the compromise of a general approach that was designed mostly for email, web browsing, and file transfers.

There are clear standards for streaming audio and video over the Internet, written-up in so-called RFCs. (The initials stand for *Request For Comment*, reflecting the open and changing nature of the Internet. But they are, effectively, standards that vendors use to achieve interoperability.) One of these describes the Real Time Protocol (RTP), a scheme for extending UDP to support media streams. This is used for VoIP telephony and is becoming standard for broadcast codecs, as well. RTP adds a sequence number to UDP packets so that the receiver can be sure packets are placed in order at the receiver. A timestamp can be used to synchronize multiple streams, such as audio and video for a television program. Just as TCP/IP are often said in the same breath and considered inseparable, so too RTP/UDP.

To recap, there are only these ways to deal with the inevitable packet loss in most wide area IP networks:

- Retransmission, such by TCP
- Forward Error Correction
- Concealment

Both retransmission and FEC cause an increase in delay that is unacceptable for two-way applications. FEC increases bandwidth and can make the loss problem worse. This leaves concealment as the best solution for interactive applications. Although retransmission may have its place when delay is not an issue, and FEC when retransmission is not possible. RTP/UDP is the Internet standard for media transport and gives us what we need to convey low-delay audio streams over IP networks.

Call Setup: SIP and SDP

Broadcast codecs can simply connect to each other by having the transmitter specify the receiver's IP number. The network passes the packet stream on to the destination device, and that is that. For nailed-up connections over dedicated links, this works perfectly well.

But we might well want our IP codecs to work like their ISDN equivalents, with a dialing function to find

and connect to the destination codec. Session Initiation Protocol (SIP) is nearly universally used for VoIP telephone service and is becoming the standard for broadcast codecs. While it is possible to have SIP connect two units with no other component, it is common to use it with a SIP Server that can help get around firewalls, provide “buddy list” features to groups of related users, and supports a relocation service so that a destination can be found regardless of which IP number it is connected to.

SIP serves as a carrier for Session Description Protocol (SDP). This signals between the two ends what codecs are available at each, and allows the system to negotiate the optimum codec among those to be had. Codec users finally have what they have been waiting for – no need to know or set the coding method before attempting a connection. Just “dial” and let the system figure it out.

Delay

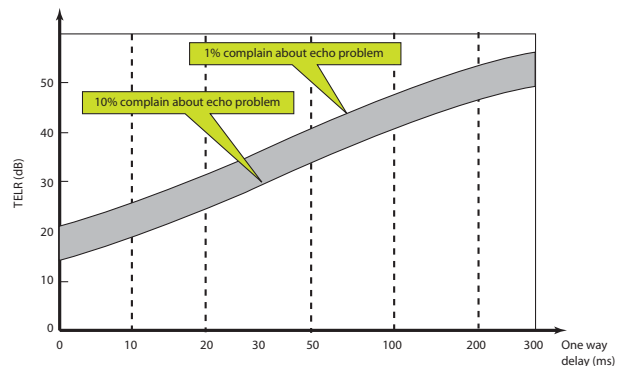
ISDN has almost zero delay, so the delay in ISDN codecs comes from the coding process, not the network. The popular MPEG AAC and Layer 3 (MP3) have around 150ms delay. This is too much for the round-trip, so the usual practice is to use G.722 for the return path, trading-off lower delay for lower fidelity. G.722 has around 20ms delay, so a connection with AAC one way and G.722 the other would have around 170ms delay.

The new AAC-ELD codec has around 60ms delay. Used on both directions, that would result in 120ms delay from the coding process. On a good IP network connection, there could be around 50-100ms delay from packetization and buffering, making the total delay around 170-220ms. That is acceptable for two-way conversation, but is pushing the limit. It’s about the same as the delay in mobile phone conversations, which people have become accustomed to, so we can expect talent to be reasonably satisfied.

Echo

With IP we are going to need to give even more thought to echo than with ISDN codecs. In a perfect system, there will be no delayed return at all to the talent’s earphones. A local mixer at the remote site sends the talent microphone audio directly to headphones and a mix-minus at the studio blocks the codec feed from returning, entirely avoiding the codec delay. But there are unintended causes of the talent audio making its way back. One is headphones worn by talent in the studio when the volume is high and isolation less than perfect. Another is telephone hybrids that don’t have sufficient trans-hybrid loss.

Annoyance from echo is a function of both delay time and amplitude, a phenomenon well-studied in the context of telephony. Thus, anything that can reduce the amplitude is going to help. Don’t use open air headphones, for example.



Annoyance from Echo is a Function of Both Amplitude and Delay (from ITU G.131)

NETWORKS

One of the advantages of IP is that there are so many ways to use it. From local networks to satellites, you have many options from which to choose.

The Public Internet

The Internet has the great twin advantages of ubiquity and low cost. One can find an IP connection almost anywhere and simply jack into it without waiting for installation or paying for it. There are no service guarantees, so you take your chances. But with the adaptive codec technology just described, the Internet becomes a reasonable proposition for many broadcast applications. There have already been successful IP remote broadcasts from airplanes, remarkably. International hook-ups are as easy as local ones.

A broadcast codec intended to be used over the Internet can benefit from having an integrated traceroute and network conditions graphing capability, so that you can see the cause of problems when they occur.

Dedicated Links

For studio-to-transmitter links and other high-reliability point-to-point applications, it is often possible to order Telco IP links that have guaranteed performance. Packet loss, jitter, delay, and bandwidth are specified in a contract called a Service Level Agreement.

MPLS Service

This is a Telco IP service that is growing in popularity. It is intended for high-quality VoIP telephony, video conferencing, and the like. Multi-Protocol Label Switching networks analyze traffic at the entry point and attach a label to each packet that describes the path the packet should take within the network. Because routers can see the packets as a stream, reserving a specified bandwidth is possible and usual. MPLS services are attractive to broadcasters since they offer a good cost/performance compromise – more expensive than non-guaranteed public Internet service, but less costly than dedicated links or ISDN.



An EVDO Mobile IP Radio Card

Mobile IP Services

Mobile IP services with fast enough uplink speeds can be used for remotes. In the CDMA world, EVDO Rev A is fast enough for us to use and is widely deployed in the USA. The uplink has a maximum rate of 1.8 Mbit/s, but under normal conditions users experience a rate of approximately 500-700kbps. EVDO Rev B promises yet faster speeds. The bandwidth is shared and there is a possibility of oversubscription and thus packet loss and insufficient bandwidth, but some trials with IP codecs have been successful. In Europe GSM is not generally fast enough in the uplink direction, though a fast service called High-Speed Uplink Packet Access (HSUPA) is just coming online, offering up to 5.76 Mbit/s bandwidth. Both EVDO and HSUPA have QoS capability in their technology specifications, but it is not clear if or to what extent mobile providers will pass this to their customers.

Access to these services is usually via a PC Card-style radio device. These are designed to be used with laptop PCs, but can just as well be plugged into broadcast codecs that include the appropriate connector and software. Or an external box can be used to interface the PC Card to the codec via a wired Ethernet connection.

LANs

Modern local area networks are usually switched Ethernet, but they can sometimes be routed at the IP level. In either case, they will have no packet loss. Recall that packet loss results from a link being overloaded. Wide area links are expensive Telco circuits, so they often have lower capacity than peak demand requires. On LANs, links are fast and cheap, so no packets need be dropped. Thus, there is no need for codecs and the other features needed for WANs. PCM linear coding can be used and no packet loss correction is necessary. While broadcast codecs could be used on these networks, it doesn't make much sense to do so. They are "overqualified" for the job – and simpler, less expensive interfaces are more appropriate. This is the domain of AoIP (Audio over IP) equipment such as Axia's Livewire.

Ethernet Radios

There are plenty of radio systems on the market that can be used for Ethernet links. Most work on the license-free ISM bands at 2.4, 5.2, and 5.6GHz. With high-gain antennas and a line-of-sight path, these can have range up to many tens of miles/kilometers. Bitrates are in the 10s of megabits, so there is much more bandwidth than needed.

WiMax

These are Ethernet radios that conform to a standard, assuring interoperability. Unlike the usual Ethernet radio which is used for point-to-point operation, these permit multiple sites to share a common channel and infrastructure. Since the channels are shared, there would be the possibility of oversubscription and contention for bandwidth. Perhaps WiMax vendors and providers will introduce some form of priority mechanism to offer guaranteed quality of service.

WiFi

IP codecs usually work over WiFi radio links without trouble. These would normally be only a part of the total IP path, perhaps extending an available DSL connection to the required location at a remote site. Again, the bandwidth is in the 10s of megabits, much more than is needed.

Satellites

Many satellite services are now IP-based and can be used for both point-to-point and point-to-multipoint links. While satellites are certainly exotic compared to other IP connection methods, from the perspective of the terminal equipment, they look about the same as any other link.

SERVICE LEVEL AGREEMENTS

With dedicated links and MPLS service, there will generally be a contract with the provider specifying the terms of their obligations with regard to quality of service. These are called Service Level Agreements (SLAs). Typically an SLA will include the following points:

- QoS guarantees: delay, jitter, and packet loss limits

- Non-QoS guarantees such as network availability. For broadcast, this should usually be at least 99.999%.

- The scope of the service. For example, the specific routes involved.

- The traffic profile of the stream sent into the network. This will be the bandwidth required, including any expected burst.

- Monitoring procedures and reporting.

- Support and troubleshooting procedures including response time.

- Administrative and legal aspects.

SIP SERVERS

A system using SIP requires proxy and registrar (also called User Agent) servers to work as a practical service. Although two SIP endpoints can communicate without any other SIP infrastructure, this approach is impractical for a public service. SIP provides a signaling and call set-up protocol for IP-based communications that can support a superset of the call processing functions and features present in the public switched telephone network. But SIP by itself does not define these features. Rather, it has been designed to enable the building of such features using servers placed on the network, which provide functions that permit familiar telephone-like operations: dialing a number, causing a phone to ring, hearing ring-back tones or a busy signal.

Servers can include the following components:

Proxy Server: These are the most common type of server in a SIP environment. When a request is generated, the exact address of the recipient is not known in advance. So the client sends the request to a proxy server. The server on behalf of the client (as if giving a proxy for it) forwards the request to another proxy server or the recipient itself.

Redirect Server: A redirect server redirects the request back to the client indicating that the client needs to try a different route to get to the recipient. This generally happens when a recipient has moved from its original position either temporarily or permanently.

Registrar: As you might have guessed already, one of the prime jobs of the servers is to detect the location of a user in a network. How do they know the location? If you are thinking that users have to register their locations to a Registrar server, you are absolutely right. Users from time to time refresh their locations by re-registering.

Location Server: The addresses registered to a Registrar are stored in a Location Server.

Presence Server: Keeps track of the status of users (such as Available or Do Not Disturb)

and makes this available to other users. There is no need to attempt a call to receive the status information.

They also can provide a gateway into the public switched network and other features such as directory and location services.

Calling with SIP

SIP calls can be made using different identification schemes.

Directly by IP number (e.g. sip:user@123.5.64.6)

A form similar to email (e.g. sip:user@host.com)

A text name processed by a directory server (e.g. WABC ZIP Codec 1)

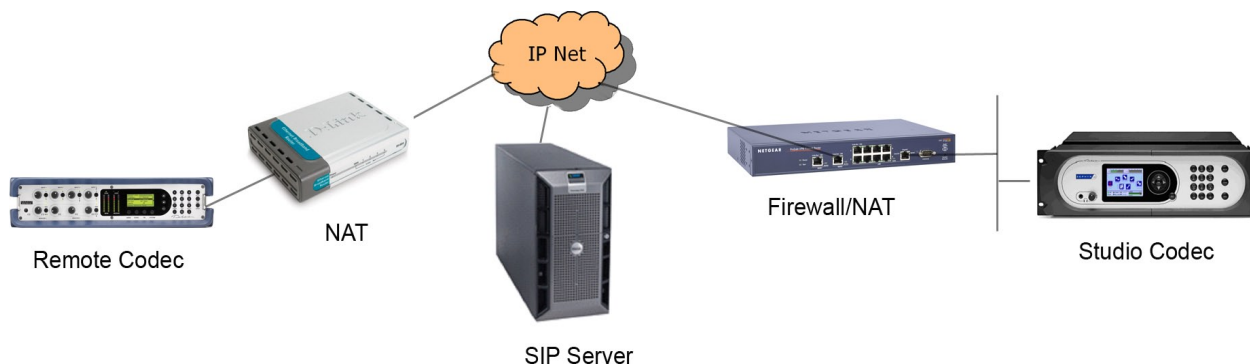
Plain old telephone numbers in the so-called E.154 format (e.g. +1 216 241 7225)

When needed, the SIP server translates telephone numbers to IP addresses using a procedure called ENUM (from tElephone NUmber Mapping). (But as VoIP and SIP catch-on, there is not much need for keeping phone numbers around. Wouldn't it be better to just use someone's email address for both text and voice messages? And if you want to call a company, why not pbx@telos-systems.com rather than an obscure 10-digit number string?)

FIREWALLS AND NATS

Network Address Translators are widely used on DSL Internet connections to allow more than one computer on the inside to share a single IP number toward the outside. All connections must originate from a computer on the inside. Since unsolicited incoming traffic can't get through, NATs provide a basic firewall function. This means that any codec inside a NAT would be both invisible and unreachable by another codec on the other side. Firewalls have the same effect. What can we do about this?

A simple solution is to put the studio codec directly on the public Internet, making it visible to codecs at remote sites even when they are behind NAT/firewalls. But it would not be possible to make a call from the studio



Typical Remote-to-Studio Set-Up
The SIP Server, outside firewalls and NATs, allows codecs inside to connect with each other.

to the remote codec, and having anything on the Internet without firewall protection is inviting problems. So we need to think about alternatives.

Consider that web traffic certainly moves both ways past these NAT/firewall devices. This happens because the NAT or firewall is usually “symmetric”, meaning that when a packet stream is sent from the inside toward the outside, the NAT/firewall opens a return path for some period of time. Placed outside any firewalls, a SIP server can both receive and make calls to codecs located inside firewalls. Codecs register with the server, which then takes advantage of the open return path through the NAT/firewall to send an acknowledgment. The server sends additional messages periodically to keep the path open. When a codec wants to connect with another, it contacts the server, rather than the other codec directly. The server knows where to find the other codec and has an open path to it, so it can signal that a connection is being requested. Each codec can now send messages and audio streams to the other, thus opening a direct return path the other can use.

With unusually restrictive NATs and firewalls, the server can act as a relay for the set-up messages. In extreme cases, it may even have to relay the audio stream.

The Telos Z/IP Server

The Telos Z/IP server was designed to support the Z/IP codec. It is similar to a SIP server, but is specialized for broadcast codec application. It provides the following functions:

Directory Services: Allows for easy discovery of other devices. The user controls the visibility of their device in the directory. A device may be (1) visible to all, (2) visible only to the group it belongs to, or (3) not visible in the directory. A device always belongs to a group (by default it belongs to the “public” group). The user may create a group at any time as long as the group name is not already in use. By giving others the group password you allow them to add their device to the same group. This also allows them to view devices that are visible only to the group.

Presence Services: Allows a user to view the connection state of their “buddies”.

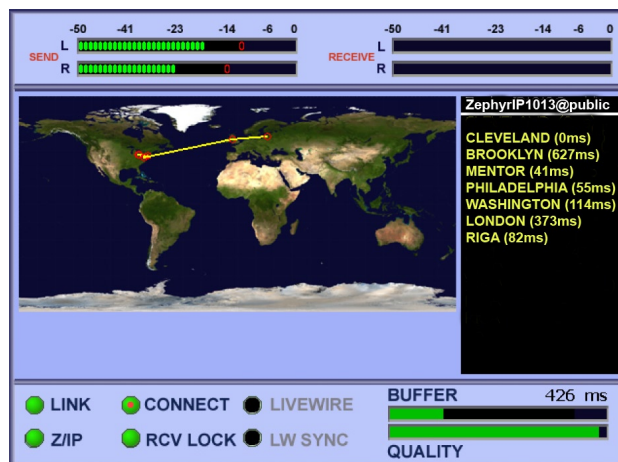
NAT Traversal Services: Allows a device to discover their public address and NAT type, if any. Keeps a connection open to devices that are not usually reachable behind the NAT to allow incoming calls to get through. For more restrictive NAT types, the server relays the signaling information to assist the connection establishment. For the most restrictive NAT types the server can also function as a media relay.

Geolocation Services: Allows a device to find out the geographic location of their device and of the other end as well as the path taken (visual

traceroute).

QoS Data Collection: The server keeps track of call QoS data reported by devices. This information can provide us with some insight about the packet drop patterns, bitrates achieved, etc.

SIP servers can be public or private, and the same is possible with the Z/IP server. Telos operates a public server that Z/IP clients can use without having to support their own, or private servers can be installed by Z/IP owners.



With Server Support for Geolocation Services, a Codec can Display a Visual TraceRoute

THE EBU N/ACIP STANDARD

The European Broadcast Union (EBU) has concluded a process that has resulted in a standard for broadcast codecs. Their goal was to ensure that codecs from various vendors have modes that interwork with each other. To be compliant, each manufacturer must support a core set of functional components. A key is the use of SIP for call set-up and SDP for codec description, since this offers automatic negotiation to the “least common denominator” codec. A manufacturer is free to add its own enhancements as additions to the required core, but these might only work when codecs from that manufacturer are talking to each other.

The N/ACIP (Norm/Audio Codec over IP) standard specifies the following codecs:

Required

- G.711 (the standard telephone codec)
- G.722 at 64kbps
- MPEG-1/2 Layer 2 at 32-384kbps
- PCM linear at 12/16/20/24-bits and 32/48kHz

Recommended

- MPEG-4 AAC
- MPEG-4 AAC-LD
- MPEG-1/2 Layer 3 at 32-320kbps

Optional

MPEG-4 HE-AACv2

Enhanced APT-X

Dolby AC-3

AMR-WB+

Since none of the required codecs have concealment mechanisms and the standard does not require either TCP (or other retransmission) or FEC, it seems the EBU is targeting networks that have guaranteed QoS.

VOIP TELEPHONES FOR BROADCAST

With broadcast codecs migrating to IP and telephony moving to VoIP, seems some kind of convergence is inevitable. Both of these use SIP/SDP, so a particular call can be specified to use either a telephone-grade codec to interwork with the public switched voice network or a high-fidelity one. Broadcast IP codecs that conform to N/ACIP have the G.711 codec that traditional telephony uses, in fact.

Mobile phones might well use VoIP in the future, as some proposals within that industry are suggesting. Already phones like the Nokia E-series and several WiFi enabled mobile phones have SIP clients hardcoded into the firmware. Such clients operate independently of the voice part of the mobile phone network. Some operators actively try to block VoIP traffic from their network and in this case VoIP calls are done over WiFi. Several WiFi only IP hardphones exist, most of them supporting either Skype or SIP.

As a manufacturer of both on-air telephone systems and codecs, we are always thinking about opportunities to make the lives of our clients easier by combining the two systems. As time goes on, they may be entirely merged, though the operator interface would probably

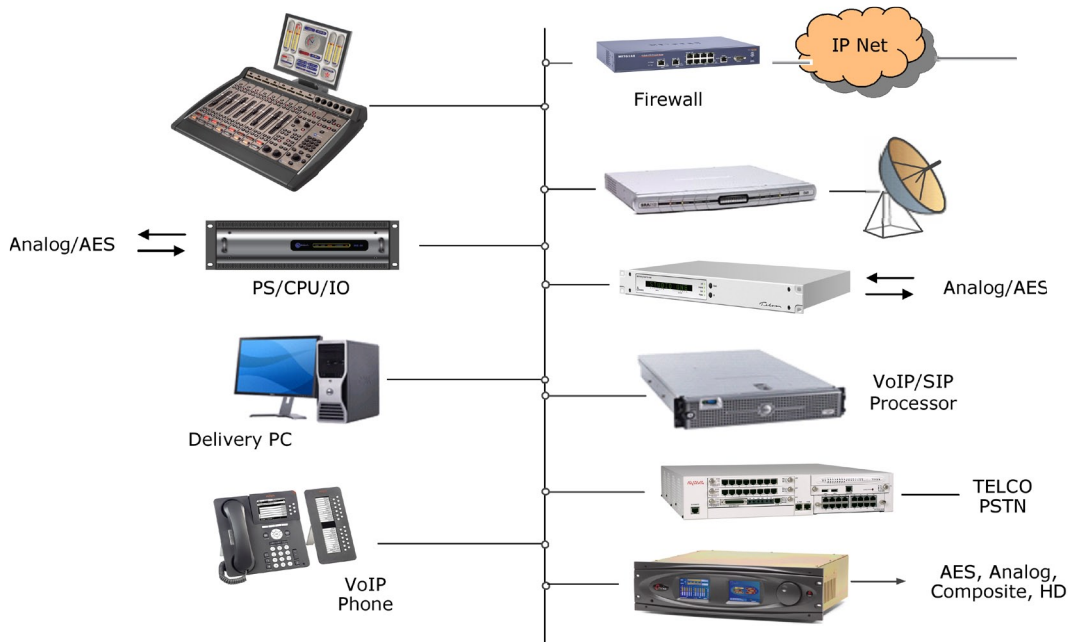
remain much as it is today. It probably would be more confusing than simplifying to have codec feeds appearing on telephone call selector button banks.

IP-BASED STUDIO EQUIPMENT

Studio audio distribution and mixing is migrating to IP, as well. Many stations are now using IP systems to distribute pro-grade audio throughout their facilities. With most audio now either originating or being sent to PCs, there is good logic behind using their native Ethernet/IP connections rather than a sequence of a sound card and an analog or AES3 interface to station audio systems. A 24-bit transparent connection is established simply and directly via the low-cost Ethernet link.

We have another advantage from this approach. It allows yet another convergence step as telephone, codec, and studio-grade audio share the same infrastructure. An Ethernet switch or IP router at the core can serve as a distribution router for everything. Control communication can pass on the same network. The mixing console can be tightly integrated with the telephone, codec, and call-screening application. Multiple studios can share "lines". Etc.

A couple of years ago, the *Wall Street Journal* had an article calling IP the "Pac-Man" of protocols, because it seems to be devouring everything in its path. With Ethernet as its LAN transport partner, its advantages are compelling. More IP-based PBX lines are now being installed than the traditional kind. Radio broadcasting technology has always taken cues from telephone world. It now follows the data technology world as well. (Count the number of PCs in your station.) The two are increasingly coming together outside our stations and are becoming ever more wedded within them.



An All-IP Facility Integrates Studio Audio, Telephone, Codec, and Control on a Single Network Infrastructure