VoIP

An Overview of VoIP Technology, How It Works, and How To Use It

Introduction

At Comrex, it's our job to keep ahead of new and intriguing technologies that we can leverage for our customer, the broadcaster. But it's important that as we ride the wave of new tech, we don't forget about the people in our industry who have "stuff to get done", and can't afford to spend hours reading about all the newest developments.

We've found this to be the case in recent years with the introduction of ISDN, POTS codecs, and IP audio codecs. In each case, we decided to put together a "primer" for those who wished to learn the knowledge needed to use these tools effectively, but were short on time. The goal was to put together all the vital information in a booklet that could be consumed in under an hour. The feedback we got proved these efforts have been worthwhile.

A new disruptive technology is taking hold, and it's now time to cut another primer. Due to cost and necessity, broadcasters are finding they need to get educated about Voice over IP (VoIP), and do it fast.

Here are some basics about VoIP in an easily digestible form.

VoIP provides a way for computer networks and other devices to emulate traditional phones and phone lines. Most modern business PBX systems have migrated to VoIP already. In some circumstances, legacy phone lines (PSTN or POTS) are no longer available and VoIP is the only choice.

Like a traditional line, a VoIP link consists of a service provider and an end user who owns a telephone instrument. But in this case, the provider is based in the "cloud". Alternately, the VoIP lines can be delivered from an upstream PBX. The end-user gear is a specialized VoIP telephone, or software running on a PC or mobile device that performs the same functions.

The Comrex STAC VIP is a sample of a device designed to interface with VoIP service. It can handle six or twelve calls simultaneously and provide the typical screening, audio processing, and control functions expected of broadcast call-in systems. For users with less call volume, the VH2 Hybrid is a dual-channel VoIP-to-studio interface. In addition, all Comrex IP codecs like ACCESS and BRIC-Link can communicate over standard VoIP protocols.

IP Concepts you need to know

If you're already an expert on IP networking concepts in general, feel free to skip to the next section about RTP. But here are a few basic concepts you'll need to master to continue learning. This is much less than a complete overview of IP networking--only concepts directly relevant to VoIP are covered.

IP basics

IP is short for Internet Protocol, but it doesn't always pertain to the Internet (as in, the public version). In a nutshell, IP networking involves creating packets of data, attaching certain headers to specify contents and assign addresses, and applying them in sequence to some kind of network capable of transmitting them. Physically, the network is usually Ethernet, although it may be Wi-Fi, 3G, satellite, or lots of other mediums.

Addressing

Devices connected to an IP network are dealt an "IP Address". Under the IPv4 protocol (the most widely implemented), this address consists of a 32-bit numeric value. Putting on your "binary thinking cap", this can also be thought of as four 8-bit bytes. A byte can have a value from 0-255, so IP addresses are usually written as a sequence of four decimal numbers (separated by dots) like 192.168.0.23 with each integer having an upper limit of 255.

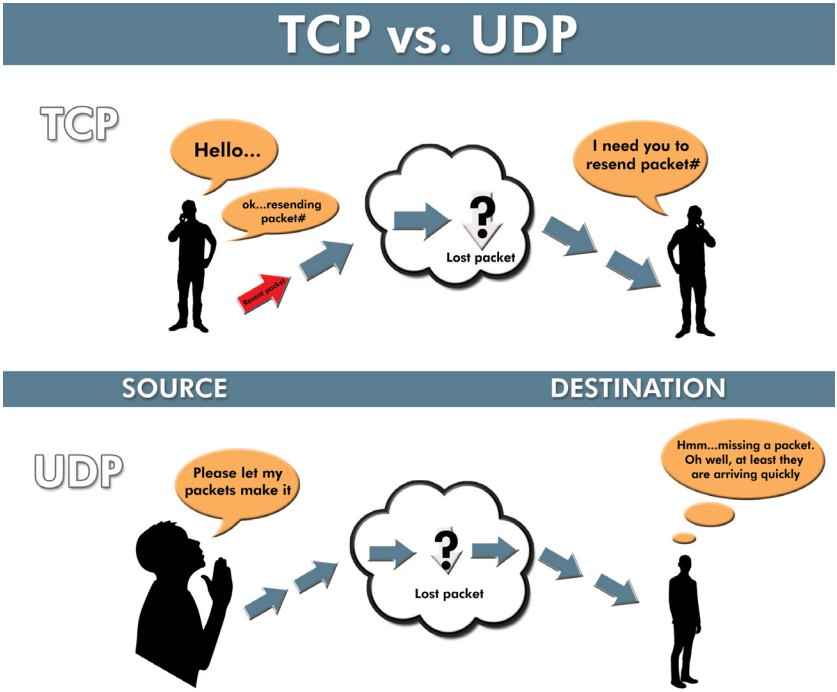
Ports

The IP address is the main identifier used to specify a destination to send packets to within a network. But since IP compatible devices can make simultaneous connections for different reasons (e.g. web surfing and email), a scheme is used to designate a specific "port" on a machine, which is essentially a 16-bit sub-address contained within the header of the packet. These ports are usually written as simple decimal values (e.g. 80, 5060), and traffic sent to a specific port on a machine can only be accessed by a program or service "listening" on that port.

TCP vs. UDP

The most common types of IP traffic fall in two sub-categories, TCP/IP and UDP/IP. The difference is important. Most web-related traffic travels via TCP, which has built in mechanisms for integrity checking and error-correction. This means that if the TCP "stack" within a machine has delivered a packet from the network, the packet is guaranteed to be correct, and if lost will be resent. It might surprise you to know that it's not TCP that's used for most real-time media on the web. This is because TCP has quite a bit of overhead in terms of data, and can easily add time delays if packets get corrupted.

VoIP and other real-time communication protocols use UDP, which is a much simpler delivery method. There is no error correction or resending available at the native UDP layer. UDP is sometimes referred to as the "send and pray" method, since the network provides no guarantees of delivery of any kind. In it's simplicity, UDP is a better choice for real-time communications because higher-level applications can be designed to make smart choices about error protection vs. delay.



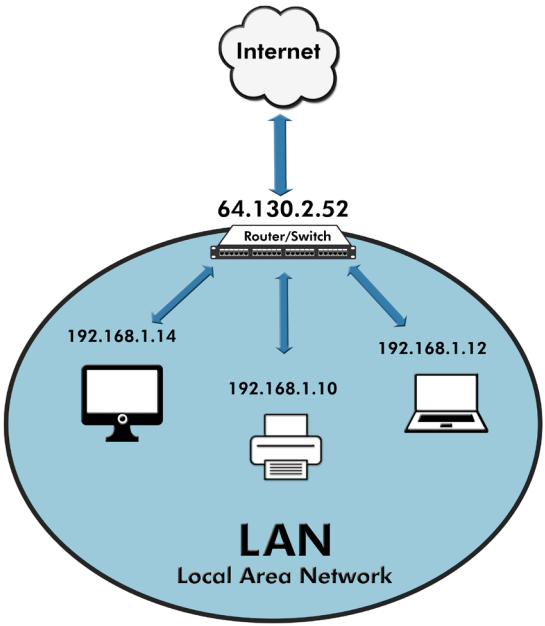
Packets sent on IP networks will include a destination IP address/port combination, and a source IP address/port combination. These act like the destination and return address on an envelope, and allow the packets to be responded to over the network.

The destination port is the most important to IT people, as it's the one that they need to be sure is open to receiving communications. When IT folks refer to a service as "running on port x" they are referring to the destination port.

We designate an IP connection via its protocol, destination IP address, and port combination in this form: **<protocol> <destination address:port>** e.g. **UDP 192.168.0.7:5060**

LAN vs. Internet

Most of the networking you'll be dealing with will exist within your LAN (Local Area Network) and connections between devices within the LAN follow ordinary rules to send packets between each other. But in the situation where you wish to connect to a device outside the LAN (which is most common) special rules need to be followed.



LANs have IP addressing conventions that allow a range of addresses to be reused within the network, and prohibit those addresses not to be used again on the public internet. This allows for many devices to site behind a router, which has a single internet (publicly addressable) IP address, and each LAN device to have a private, reusable IP address. By convention the address ranges start with the digits 192.168.x.x, 172.16.x.x, or 10.0.x.x. So, for example, if a machine tries to connect to another at an address of 10.0.0.75, it is necessarily trying to send packets only within its LAN. The range of addressable LAN addresses is called a subnet, and must be programmed into each machine using a subnet mask entry.

If a machine on a LAN wishes to send packets outside the subnet, it must communicate with a gateway (usually a router) at a fixed IP address.

Network Address Translation

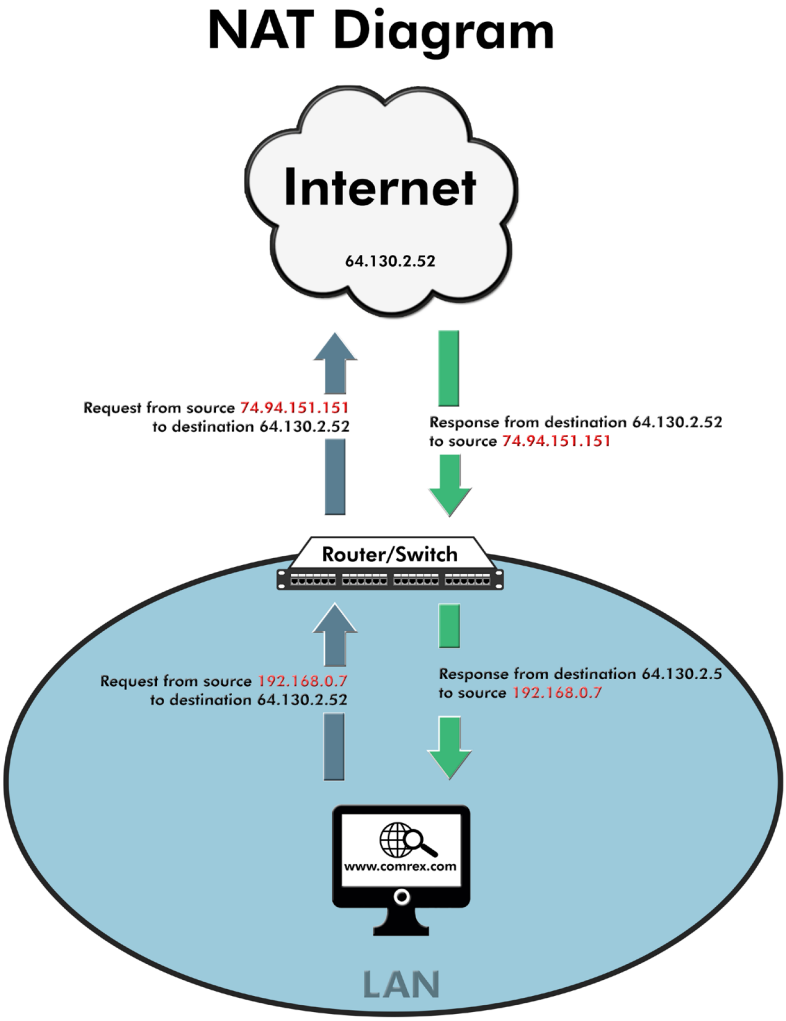
The concept of how a gateway router provides translation services to the Internet is extremely important in the field of VoIP, if only because it causes so many headaches. Known as Network Address Translation (NAT), it's easiest to use a diagram to illustrate a typical gateway scenario describing a user on a LAN accessing a web page at comrex.com. For this illustration, we'll ignore the concepts of DNS and URLs (which aren't particularly useful for VoIP) and live the fantasy that the user is accessing the comrex.com page via its public IP address, which is (as of this writing) 64.130.2.52. In our scenario, the user has a laptop on a LAN using the popular 192.168.0.x subnet addressing scheme, and specifically has the address of 192.168.0.42 assigned to it.

The user will input the web page address into his browser, and the computer will recognize the address as outside the subnet it has been programmed to work on. So it will form a packet, whose payload consists of a request to view the web page, and hand it to the gateway router, which is located at the local address programmed into the laptop (192.168.0.1).

Because the router is acting as a gateway, it actually has two IP addresses. The LAN address (192.168.0.1) is used by devices on the LAN. The WAN address (74.94.151.151) is the address assigned by the Internet Service provider. This address is public, in that it is addressable by every device on earth that is connected to the Internet.

The router will record the source address of the packet (192.168.0.7), change it to the public IP of the router (74.94.151.151), and send it along to the destination IP address. This is so the web site knows the correct address to which to respond.

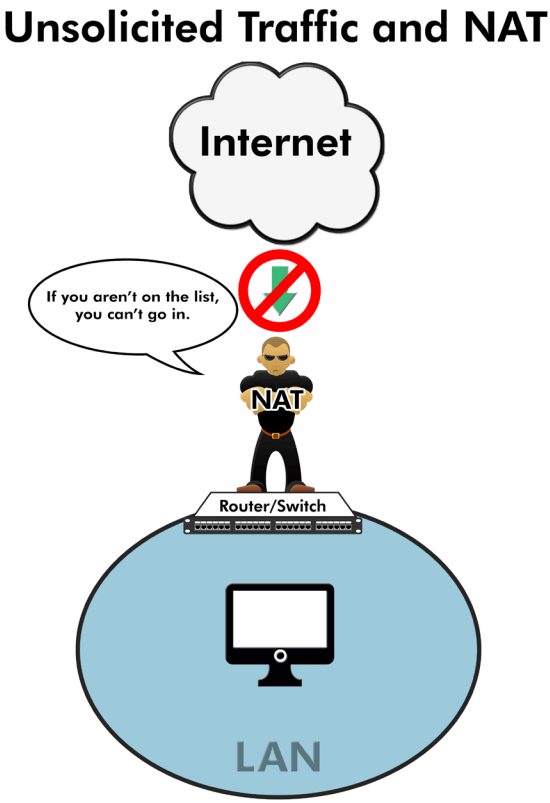
The router will now wait for the response from the web site (it's smart enough to know to expect something from the destination address of the packet it sent). It will then change the destination address of the packet to the private IP address of the laptop before sending it along to the LAN.



In reality, NAT is more complex than this, changing port numbers as well, but we've kept the concept to the bare basics to outline why NAT hurts VoIP.

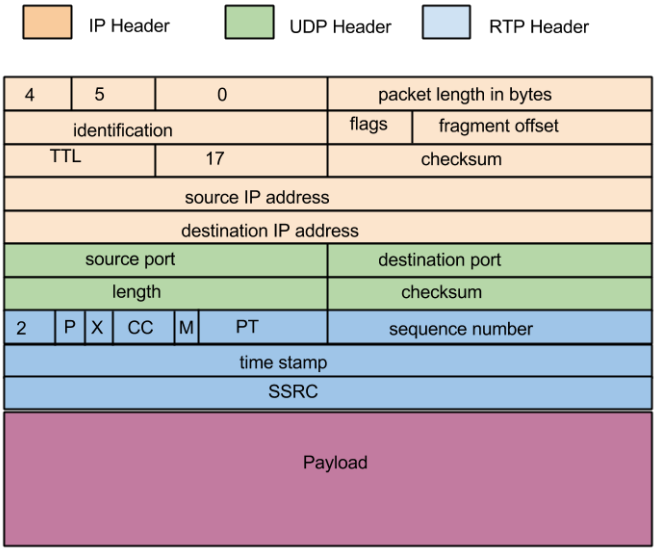
NAT provides for many benefits, including address reuse and basic security. This security exists because packets that arrive from the public Internet without being requested from within the LAN will be discarded. But it's this security element that makes VoIP difficult when using NAT. The concept of placing a VoIP call to a device behind a NAT requires that the NAT deliver unsolicited packets from the Internet to the VoIP device.

This is a complex topic, and as we'll see later on, NAT traversal can cause all sorts of trouble for VoIP.



Real Time Protocol

A fundamental building block of VoIP is the Real-Time Protocol (RTP). This is a protocol layer that exists within a UDP packet specifically designed to transfer audio (and video) media with low delay. RTP consists of a header that is applied directly after the UDP header in the packet, followed by a media "payload" which consists of the actual encoded audio of a VoIP call.

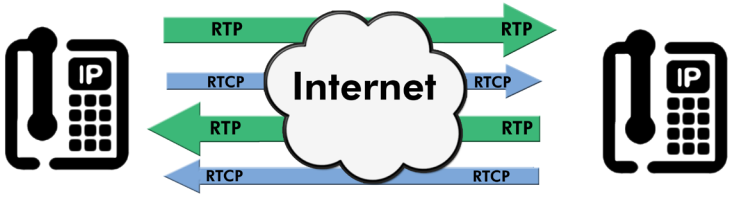


The primary responsibility of the information in the RTP header is to allow the decoder to find the proper playout sequence of the media contained in the packet. RTP doesn't contain any intelligence about what is actually contained in the payload--this has to be handled by other means.

An RTP stream is unidirectional. If a duplex stream is required, an additional independent RTP stream must be initiated in the reverse direction (This function is handled by the Session Initialization Protocol (SIP) layer discussed later).

Finally, an RTP stream (or session, as it's called) has a companion stream that is initiated and travels alongside it for the duration of its life. It's called RTCP and is sent to the same IP address as the RTP stream, but at one port higher. It's used for RTP stream quality statistics but doesn't carry any actual audio, so it uses a small amount of data. But it's important to know about if you're troubleshooting firewall or NAT issues.

RTP Diagram



RTP alone can be the basis of a very primitive VoIP call. If each end of the call knows in advance information about encoders used, no NAT routers are involved, and the call can be manually initiated and answered on each end, RTP streams can be "pushed" between the destinations and will provide the path for VoIP. Of course, real-world VoIP involves much more, so we need to add complexity to the system.

Encoders

Broadcasters who've used POTS, ISDN or IP audio products are familiar with the concept of encoding compression. This is the choice of encoder within the system used to compress digital audio so it uses less network capacity. Encoders like MP3 and AAC are common in that world.

You'll see the VoIP industry use the term "codecs" for this function. But because broadcast transmission devices are also termed "codecs", we'll reserve it to describe hardware, and use "encoders" to describe compression algorithms.

VoIP has its own spectrum of useful encoder choices. VoIP encoders require very low delay and reasonable computational complexity. The RTP protocol has definitions for how to fit all popular encoder payloads into a session.

G.711

The lowest common denominator encoder in VoIP is the same one that has been used by digital telephone networks for decades, defined as G.711. It's a simple way to compress audio, resulting in a network utilization of 64 Kb/s per channel in each direction, a compression of about only 30% from the original uncompressed stream. This is considered the highest amount of allowable data for a single call by modern standards, and it can add up quickly as multiple

calls are handled on the same network. To its benefit, the encoder requires virtually no computer power to compress or decompress.

G.711 is limited in terms of audio fidelity by the choice of its audio sampling rate. Calls using this encoder usually provide only 300 Hz-3 KHz audio response, resulting in the familiar thin sound of phone call, especially when put "on the air".

G.711 actually has two variants, one used mostly in North America (μ-law), and another used elsewhere (a-law). These are defined by the names of the tables used within the encoders to compress. All Comrex codecs and VoIP devices support G.711.

G.729a

Because G.711 is a bit old and primitive, an encoder has been developed to deliver equivalent audio quality while using a fraction of the network bandwidth. G.729a implements a more aggressive compression algorithm, resulting in network usage of around 8 Kb/s per channel, or about 1/8th the data of G.711. This can be very helpful for avoiding excessive network congestion. Of course, equivalent audio means the same limited fidelity as G.711.

This encoder is sometimes simply referred to as G.729 (without the a), but is equivalent to the user. Another variant, G.729ab, is sometimes available that can detect when voice is present and squelch the data stream during periods of silence, further conserving network bandwidth. Comrex STAC VIP supports G.729a.

G.722

Familiar to ISDN broadcasters, G.722 is an encoder designed to increase the audio fidelity of phone calls. Using the same network bandwidth as G.711 (64 Kb/s each way), G.722 more than doubles the audio spectrum conveyed by the call, making the caller sound much more natural and identifiable. The 7 KHz spectrum carried by G.722 covers the majority of human voice energy, excluding only the most sibilant sounds in speech.

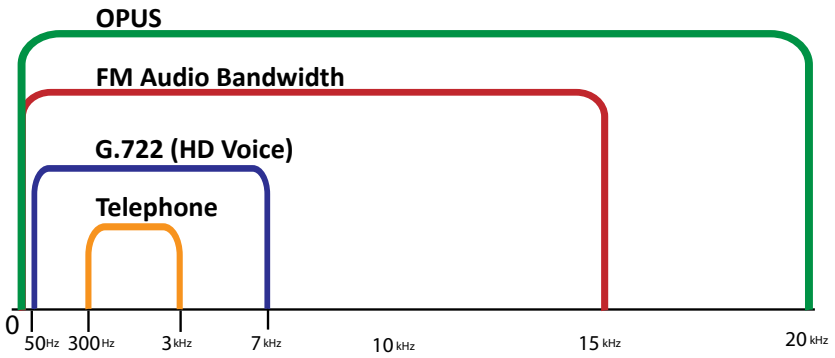
G.722 is the most common encoder for calls that are classified as "HD Voice" in the VoIP world. All Comrex codecs and VoIP devices support G.722.

Opus

Efforts are increasing at combining the worlds of VoIP and web services. Many web audio services have standardized on Opus, an encoder that delivers near-CD quality audio with low delay. As these efforts continue, users can expect to find more support for the Opus codec in VoIP devices and networks. All Comrex codecs and the STAC VIP phone system support Opus.

Other encoders

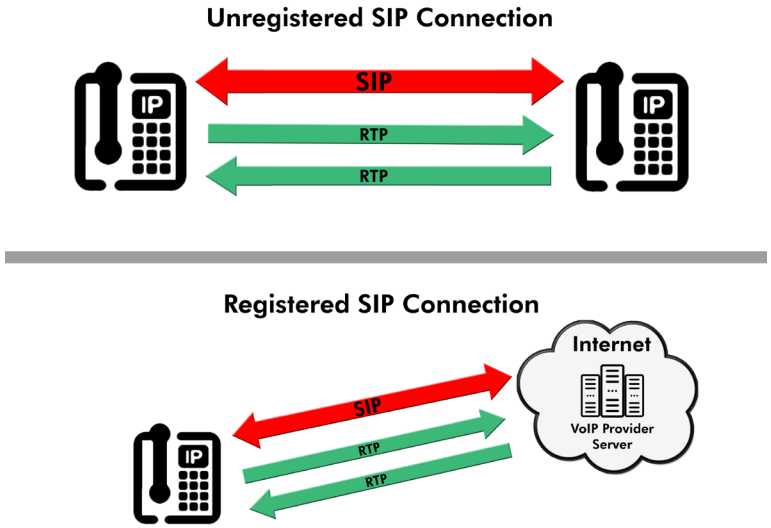
A large spectrum of VoIP-ready encoders have been introduced in the past decades, each having proponents and particular advantages for certain applications. These include iLBC, iSAC, G.722.1, G.722.2, G.726, VMR-WB, SILK and AMR-WB+. For the most part, we expect the industry to support only the four encoders outlined above in most equipment and networks.



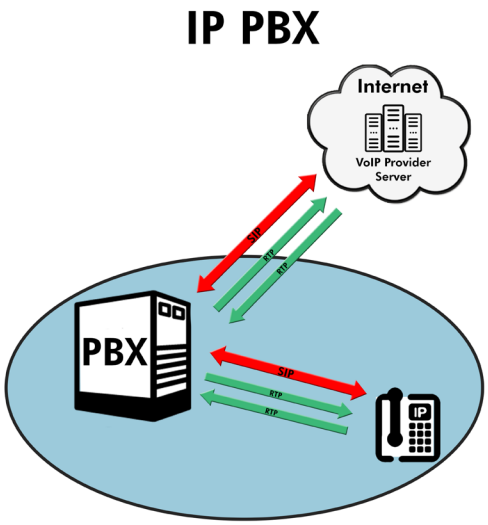
Session Initialization Protocol

The piece that ties RTP sessions and encoders together, and gives VoIP its telephone-like qualities, is another completely separate connection between devices called the SIP. You'll see the term SIP thrown around in place of VoIP in many places (SIP Phones, SIP PBXs). It's a very powerful specification and is being used for an increasing number of applications besides VoIP, like compatibility standards between broadcast IP hardware codecs, studio-style AoIP installations, and real-time web audio and video. It's becoming such a vital element of so much new technology, it's a very valuable thing to be expert in.

SIP connections can be made in two primary ways--registered and unregistered. In unregistered mode, a SIP channel is opened between devices at the time a call is placed. In registered mode, a SIP channel is constantly maintained between a SIP client (like a studio talkshow system) and a SIP server (like that at an Internet Telephone Provider). Most VoIP users will only use registered mode, so that's what we'll focus on going forward.

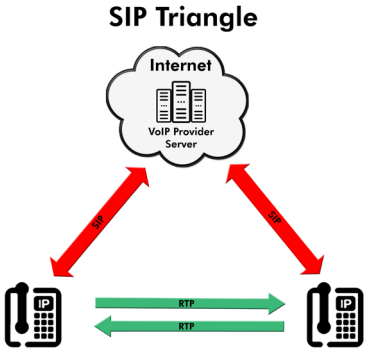


The SIP protocol can be used in more than one link in a VoIP chain. The best example would be a purely IP PBX. In this case, the PBX maintains a SIP channel to an Internet Telephone provider on its WAN port. It also maintains several SIP connections over its LAN to telephone extensions. Because the protocol used in these links is identical, it provides for a lot of flexibility. For example, if need be, the telephone extensions could register directly with the provider, bypassing the PBX entirely.

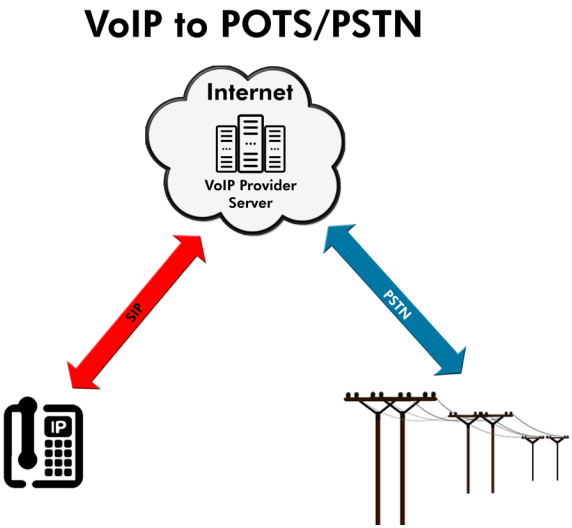


It's important to understand that the SIP protocol does not carry any actual voice between devices-- it simply instructs devices to create separate RTP sessions in each direction. RTP streams are created and destroyed based on commands contained in SIP messages when calls are made or received.

Sometimes the SIP channel is connected to a server that is removed from the RTP sessions entirely. This would likely be the case when two SIP devices are registered to the same (or sometimes even different) providers. The SIP channel would instruct the devices to create RTP sessions between them, rather than to the provider. This is known as the "SIP Triangle".



But more commonly, a SIP device is interested in making and receiving calls to and from the "old fashioned" public switch telephone (PSTN) or "plain old telephone" (POTS) network, whether wired or cellular. In this case both the SIP channel and the RTP sessions are made to a server at the Internet Telephone Provider, and the provider acts as a gateway for the voice call to the "legacy network". The user would be delivered a "real" phone number (DID for Direct Inward Dial) and the provider would handle all the necessary VoIP <-> PSTN conversions. We'll focus on this scenario from here on.



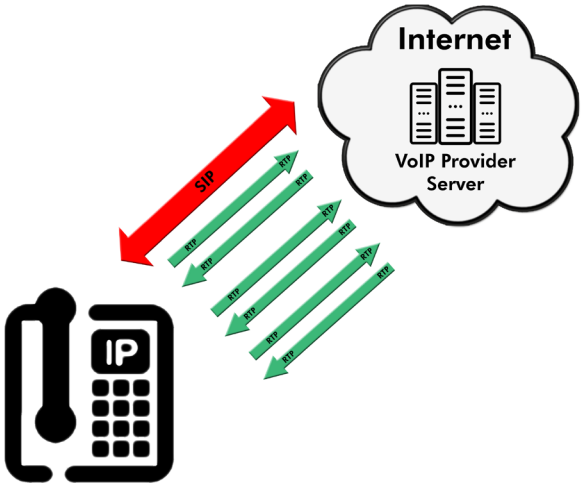
SIP Details

The technical details of SIP are widely available on the web for further research. But essentially, commands and formats are provided to invite users to a call, accept calls, end them, and reject them. SIP also provides a mechanism to register and authenticate with a server.

Another useful function in SIP is encoder negotiation. The SIP protocol can inform users of which encoders are supported on each end of a session and in which priority. In this way, it's easy to make decisions about which encoder to choose that will be in common with both ends, and to reject calls if no common encoder is found.

Like RTP sessions, the SIP channel utilizes the UDP protocol by default. There is a specific port defined, 5060, as the default "well-known" port over which SIP operates, although it can usually be configured to be different.

A single SIP channel can manage multiple RTP sessions simultaneously. In this way, only a single account needs to be registered with the Internet Telephone Provider and a single SIP channel maintained, but multiple VoIP calls can be run simultaneously. Whenever a call is initiated or dropped, a pair of RTP sessions is created or destroyed on the fly for each call.



Challenges with SIP/RTP

To summarize the previous sections, most VoIP connections involve a continuously active SIP channel initiated from the user device to a service provider over port UDP 5060. Using this channel, the two ends negotiate calls and create and destroy RTP sessions (each consisting of one RTP and one RTCP) in each direction. Like the SIP channel, these sessions also run between the end-user and provider, so the provider can bridge them to the legacy phone network. The SIP channel also negotiates which encoders will be used on the RTP channels.

So what can possibly go wrong? Almost every issue can be run down to NAT-based routers or blocking firewalls.

Issues with the SIP channel

The SIP channel generally has the fewest issues, since it's usually originated from the user end of the link. This means NAT routers on the user end will generally allow this outgoing traffic to pass, and allow the response traffic (from the provider) back in. But if a network is heavily firewalled in a way that blocks outgoing access to UDP 5060, this channel will never be created and the user cannot register with the provider.

Also, although we have described the SIP connection as "always active", there are periods of inactivity on the link when no calls are being set up or ended. In order to receive information about new incoming calls from the provider, the user end must keep the SIP connection (or "binding") open through the NAT router to prevent it from terminating the binding and blocking incoming traffic. It does this by sending periodic updates even when no changes are being made to any calls. The interval of these updates is usually adjustable, but must be shorter than the timeout value the router takes to shut down any unused bindings.

Where am I?

According to the SIP standard, the user device will inform the provider of its IP address (over the SIP signaling connection), and the provider will "push" the RTP session containing the incoming voice to that address. But devices on LANs often don't know what their "public" address is, only the private one assigned to them on the LAN. If the provider tries to initiate a stream to that address, it will go nowhere.

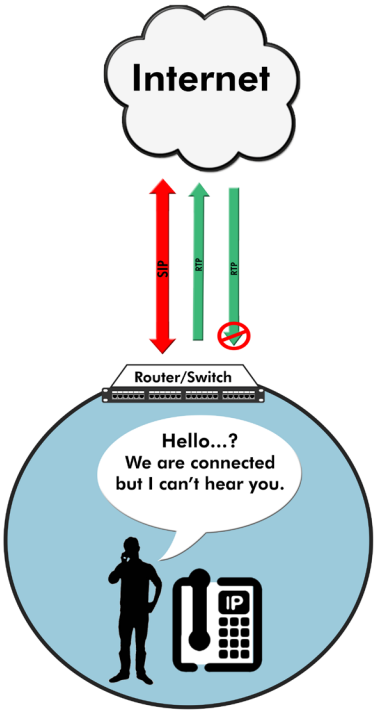
Many VoIP providers install a "cheat" here that will look at the user's IP address and determine if it looks "private". If so, they will ignore it and send the RTP stream to the destination address of the RTP session they receive.

If the cheat isn't implemented, user devices have a way of looking up their public IP address via a protocol called STUN. This protocol can usually be enabled within the user's equipment configuration. If enabled, the device will look to a STUN server out on the public Internet, and query its own address. It will then use that public address to populate the "from" field in the SIP handshake.

Don't block me, bro!

Even if the provider gets the correct IP address of the user, there's plenty that can go wrong. Remember, SIP involves creating extra RTP "channels" in each direction to carry the actual voice. The ports used on each end are negotiated over the SIP signaling channel for each call. There aren't any "standard well-known" ports used for these connections. And there can be many of them active on different ports if lots of simultaneous calls are happening.

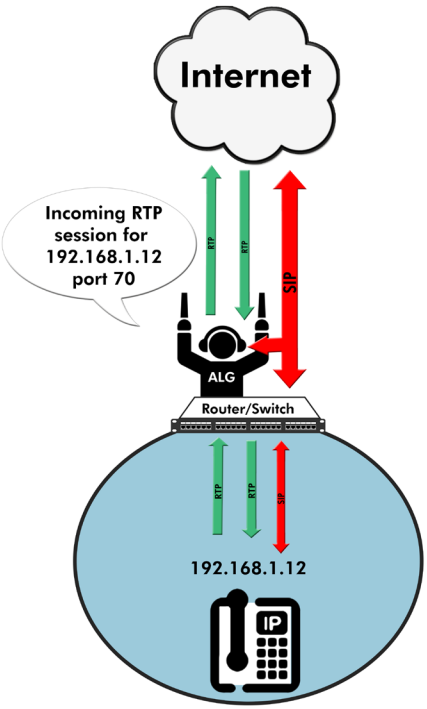
As far as the user's router or firewall is concerned, a new RTP session is trying to make it through its security layer. It's not aware this session has been requested, so it's blocked by default. This usually results in a one-way connection, where no audio can be heard on the SIP user end of the call.



ALG to the rescue

This scenario has become common enough that router and firewall manufacturers have started to address it. The solution is call SIP ALG (for application layer gateway) and has been built into the firmware of most modern devices. It may be on or off by default. And the quality of how it functions may vary--early implementations sometimes did more harm than good.

But a properly functioning ALG will listen to your SIP channel, and gain an understanding of which RTP sessions are being created on which ports. It will then allow the incoming session through.



In reality, an ALG may often take quite a bit of license with your SIP connection. It can rewrite many of the SIP fields in order to comply with its rules, so the IP and port information getting to the service provider may actually be completely different than those sent by the device. As long as it has the intelligence to open the proper ports, this will usually work fine.

It's even possible that your SIP connection is being processed by more than one ALG, as in the instance of a separate router and firewall on the connection. Of course in this scenario, the possibilities for errors compound. Sometimes it's best to disable unnecessary ALGs in the link. Unfortunately, diagnosing these issues require analyzing packet captures. Luckily, SIP is a well-known protocol that can be easily deciphered by packet capture systems.

Summary

The important elements of SIP are as follows:

1. An independent connection stays open on UDP 5060 between the user and the service provider
2. Separate and multiple RTP sessions are established in each direction for calls
3. Routers and firewalls interfere with these RTP sessions by design, but ALGs built into these devices can help.

PBXs

So far we've discussed SIP connections to outside or "cloud" VoIP providers. But many times, the user already has a SIP PBX on premises, which already connects to the public telephone network by VoIP or legacy means, like analog lines or T1s. Since most modern PBXs talk SIP to their extensions, they just need to tie a SIP-compatible device (like a codec or hybrid) to the PBX, and allow the PBX to decide how to route calls to the device.

As mentioned before, the SIP protocol used in this scenario is the same. The device will register and maintain a SIP connection to the PBX, and the PBX will inform the device of incoming calls. RTP channels will be created when required between the SIP device and the PBX. This will usually be successful, since the LAN environment is less reliant on routers, subnets and firewalls to block the RTP channels.

Registering with a SIP Server or PBX

The process of registering a device to a SIP provider, whether it's in the "cloud" or at your location, is usually simple. Much like registering an email client with a mail server, the VoIP client (the VoIP hardware) must know the location of the server, and a username/password combo with which to register. The server location can be in the form of an IP address, or a URL.

Some servers with more complex arrangements may require more information to help choose options. There may be separate settings for your SIP Proxy server, your SIP domain, and your SIP registration server. There may be choices for encoder support, auth username (an additional credential used for authentication), and caller ID options. For the most part, any essential info that needs to be programmed will be delivered from your provider (or in the case of a PBX, your Telco department) and you can set your VoIP device with the parameters that match, and ignore the others.

Making and Receiving calls

Once registered correctly with a SIP server, incoming calls will be routed to your SIP device based on the calling plan set up with your provider or PBX. Whether it's the DID line(s) assigned to you by the provider, or an incoming trunk attached to your PBX, a "ring" on the line will trigger the server to notify your device of a call request using the SIP protocol. Your device can accept or reject the call. If you accept the call, an RTP channel is created to your device each way.

Outgoing calls just reverse the process. The SIP device sends an outgoing call request to the server, which attempts to complete the call. Call progress messages will be sent to your SIP device from the server, which may translate them to familiar tones like ringing and busy. On call completion, the server will create the RTP channels in the same way as for incoming calls.

Hunting

Of particular interest to broadcasters who take lots of calls simultaneously is hunting behavior, or the way the system behaves toward simultaneous incoming calls. Keep in mind, when an incoming call is in the "ringing" state, there are only status messages exchanged over the SIP connection--no actual audio is being transferred. The RTP audio channels are only created after the call is answered.

Only one SIP connection needs be open for multiple voice channels to be created. Your VoIP provider or PBX will be programmed to allow a designated number of simultaneous voice channels, and any further incoming calls will be rejected there. By default, most multi-channel VoIP gear will "hunt" any second, third etc. call to the next "line" on the device. In this way hunting is inherent. If more than the supported number of calls is requested to the VoIP device, it will reject them in the same way as the provider does, and no RTP channel will open for these excess calls.

Alternately, it's possible to set up a separate SIP account for each "line" on the SIP device, and this account should be capable of creating only one "channel" at a time. In this case, it's the responsibility of the provider or PBX to sort the hunting arrangement and notify the proper account about incoming calls.

Choke Lines

Another topic of interest to broadcasters is choke lines, the specially conditioned telephone trunks designed not to fail under loads of thousands of incoming calls (e.g. for contests). In the PBX scenario, choke lines can easily be used as the trunks that feed the PBX, and very little changes.

When using a cloud provider, it's important to notify them about potential peak call volume to avoid overloading their systems. But cloud providers are usually equipped to provide service to high-volume nationwide call centers, so they can usually implement techniques to throttle large amounts of calls without impacting overall service.

You've now gained a general knowledge of VoIP and its underlying technology. Congratulations!

So what now?

The ways you'll use VoIP and SIP will vary, depending on the applications you have in mind.

If you need help figuring out what the best product or set-up would be for what you need to do, reach out to us!

We'd be happy to answer any further questions you have.

Call us at (978) 784-1776
or email us at info@comrex.com

COMREX
BROADCAST RELIABLE

www.comrex.com • email: info@comrex.com
19 Pine Road, Devens, MA 01434 USA
Tel: 978-784-1776 • Fax: 978-784-1717